

## INTRODUCCIÓN

¿Tienen alguna utilidad las técnicas de Inteligencia Artificial en el desarrollo de la investigación en Arqueología Teórica? Yo pienso que sí, y por eso un buen día de Noviembre de 1990 empecé a escribir este libro. En un principio pensaba limitarme a escribir una simple introducción a los Sistemas Expertos y cómo debían usarse en Arqueología; creía que era un tema adecuado para un libro, y que encontraría un amplio mercado para él, lo que facilitaría su publicación. Sin embargo, como todo libro que se precie, Arqueología Automática pronto empezó a adquirir vida propia, por lo que su apariencia actual es muy distinta de mis proyectos iniciales.

¿Qué hace un libro como éste en una colección de trabajos sobre Arqueología Mediterránea? La razón más evidente es porque la práctica totalidad de los casos y programas de ordenador que en él se estudian hacen referencia de una manera u otra a la Arqueología de la cuenca mediterránea. No todo en Arqueología se reduce a la descripción de piedras polvorrientas y cacharros rotos más o menos bonitos. Y este libro pretende ser un ejemplo de ello: Arqueología Mediterránea entendida como una reflexión acerca de cómo hacemos arqueología en el mediterráneo, y una propuesta teórico y práctica acerca de qué arqueología debemos hacer para estudiar las sociedades que han vivido, a lo largo de la Historia, en algún lugar de ese mar que puedo ver a través de mi ventana.

En este libro no se propone ninguna renovación radical de lo que hemos convenido en denominar *Arqueología*, sino que se presentan algunas de las técnicas de programación que podemos aplicar al estudio de los métodos de inferencia en Arqueología. La idea general es relativamente sencilla: el libro expone la noción de *problema* que se deriva de los recientes trabajos en Psicología Cognitiva, aplicándola a la caracterización de los *problemas arqueológicos* (en especial aquello que hacen referencia al pasado del Mediterráneo). A continuación se detallan las diferentes técnicas informáticas encargadas de resolver problemas de diagnóstico, diseño y monitorización, intentando siempre explicar las semejanzas entre esas técnicas y los modos de razonamiento clásicos: deducción, inducción, analogía.

He intentado escribir un libro fácil de leer, sin embargo la materia de la cual trata es ciertamente difícil, y requerirá del lector una atención constante, múltiples lecturas complementarias y, sobre todo, desprenderse de muchos de los apriorismos que sobre informática, lógica o filosofía de la ciencia haya aprendido. Los dos primeros capítulos son, probablemente, los más difíciles de todo el libro. ¡Que nadie se asuste! Constituyen tan sólo una introducción general al tema. Todas y cada una de las cuestiones que tratan son desarrolladas, con diversos ejemplos arqueológicamente relevantes, en los capítulos siguientes. Al final del libro descubrirá el lector que aquello que no acababa de entender al principio, se ha hecho mucho más claro.

Pienso que el libro puede ser comprendido por cualquier estudiante con algunos conocimientos básicos de Arqueología y de Lógica de Predicados. Puede interesar también al lector familiarizado con la Inteligencia Artificial, en tanto que introduce un dominio de aplicación hasta hace poco insospechado. Este segundo tipo de lector encontrará a faltar referencias a los métodos débiles (*weak methods*) de resolución de problemas. Esta omisión es voluntaria. En general, se trata de técnicas interesantes para representar el pensamiento cotidiano, el sentido común, pero que me han parecido completamente fuera de lugar en una obra cuyo objetivo es la simulación informática del razonamiento científico.

No está de más insistir en el hecho que este libro es una *introducción*, de ahí que haya abordado tan sólo la superficie del tema. Nadie debe esperar convertirse en un experto en Inteligencia Artificial después de haber leído las páginas que siguen. Habré cumplido con los objetivos que me planteé al escribirlo si alguno de sus lectores potenciales considera que «le ha sabido a poco» y se decide a desarrollar por su cuenta alguno de los temas. Con ese fin aparece al final de cada capítulo una serie de lecturas recomendadas. Además, el Ane-

xo final incluye una lista de programas de ordenador «asequibles», para que el arqueólogo empiece a ejercitarse en el arte de programar *arqueólogos automáticos*.

Tampoco debe confundirse este libro con un recetario de cocina. No ofrezco ni técnicas perfectas, ni fórmulas milagrosas que solucionen todos los problemas que la Arqueología Mediterránea tiene planteados; sino una descripción «desde dentro» del razonamiento arqueológico. Aunque hay algunas referencias *prácticas*, no es éste un libro para aprender a construir un Sistema Experto o que explique cómo se usa una Red Neuronal. Para ello lo mejor es referirse a los manuales que acompañan los diferentes programas recomendados al final del texto. La utilidad del libro radica, encarnamente, en reconocer que existe una manera distinta de estudiar qué es y cómo es la Arqueología. Los resultados de ese estudio es algo que dejo en manos de aquellos lectores que elijan desarrollar los contenidos del libro. Hoy por hoy la *Arqueología Automática* es más una promesa futura que una realidad; es un dominio por explorar que proporcionará resultados extraordinarios e impredecibles en los próximos años. Animo a los lectores a no ser simples espectadores de esos avances, sino a participar activamente en la aplicación y desarrollo de la Inteligencia Artificial en Arqueología.

A lo largo de la elaboración de estas páginas he recibido la ayuda de bastantes investigadores, la mayoría de los cuales han tenido la amabilidad de ponerme al día acerca de sus trabajos. Deseo expresar mi agradecimiento, por tanto a: Christiane Bron-Pury (Université de Lausanne), Costis Dallas (Benaki Museum, Salónica), François Djindjian (CNRS, Paris), Jim Doran (University of Essex), Henri-Paul Francfort (CNRS, Paris), J. Ganascia (Université de Paris), Jean Claude Gardin (CNRS, Paris), Paul Gibson (University of York), Amador González (Universidad Politécnica de Madrid), Marie-Salomé Lagrange (CNRS, Paris), Jean Le Dizes (CETE-Mediterránea), Ruth Maicas (Universidad Autónoma de Madrid), Mike Palmer (University of Essex), Raquel Piqué (Universitat Autònoma de Barcelona), Paul Reilly (IBM Research-Center), Robert Reynolds (Wayne State University, Detroit), Steve Shennan (University of Southampton), Wiktor Stoczkowski (CNRS, Paris), Arthur Stutt (The Open University), Vanda Vitali (Ontario Museum), Bert Voorrips (Universidad de Amsterdam). Ninguno de ellos es responsable de las inexactitudes u omisiones que yo hubiese podido cometer al comentar sus investigaciones.

Empezado en París, concluí este libro en Barcelona. En ambas ciudades recibí el apoyo y la amistad de los miembros de dos soberbias organizaciones científicas: el Equipe de Recherche No. 315 del Centre National de la Recherche Scientifique (bajo la dirección de Henri-Paul Francfort), y el Departament de Història de les Societats Precapitalistes i Antropología Social de la Universitat Autònoma de Barcelona, convertido recientemente en División de Prehistoria.

Con Aureli Alvarez, Jordi Estévez, Luis Lumbreras, Raquel Piqué y Asunción Vila tuve ocasión de discutir muchos de los temas que aquí aparecen, y de llevar a la práctica aplicaciones que antes creía irealizables.

En la Universidad Autónoma de Barcelona, utilicé a mis alumnos del curso de doctorado «Técnicas de Inteligencia Artificial en Arqueología», como conejillos de indias: si eran capaces de entender el contenido del libro, otros lectores podrían. Ana Delgado, Elisenda Curià, María Pallarés y Mercedes Párraga, especialmente, supieron enseñarme que un profesor no siempre acierta cuando cree estar acertado. Gracias a ellos y ellas, el volumen inicial de más de 400 páginas, llenas de fórmulas matemáticas y demostraciones de teoremas de lógica formal, se ha convertido en lo que el lector tiene hoy en sus manos.

Mis compañeras y compañeros del equipo «Guadalhorce» (María Eugenia Aubet, Pedro Báscones, Elisenda Curià, Ana Delgado, Antonio Fernández, Mercedes Párraga, Apen Ruiz y Magda Salas (el nombre del equipo deriva del nombre del yacimiento fenicio que excavaban) son también deudores de mi gratitud, ya que aceptaron una tecnología «excéntrica» por el mero hecho de ser yo el que la defendía, y además acogieron al excéntrico tecnólogo que la proponía. Su amistad y colaboración me han permitido no sólo acabar un libro que parecía no tener fin, sino que todos ellos me enseñaron que la arqueología no acaba ni en el método ni en la técnica.

Mis amigos y mi familia también han contribuido lo suyo. Con unos he discutido de todo lo divino y lo humano, y muchas de esas discusiones aparecen de una forma u otra en este libro. Los otros se han limitado a soportarme, lo cual, conociéndome, es todo un mérito.

Y finalmente, pero no en último lugar, a *alguien* (!) sin cuya ayuda todo el proyecto habría quedado en el reino de los sueños. Aun cuando pueda parecer una herejía, doy las gracias a mi ex-sufrido ordenador (un Macintosh SE/30, adaptado a mi imagen y semejanza), del cual me «divorcié» hace un año, sustituyéndolo por un flamante Power Mac 6100, con el cual vivo aparejado, para estupor de mi familia. Entre virus (el pavoroso WDEF-B!) y otras incomprendiciones hemos podido trabajar juntos, aunque sin entendernos del todo.

Y como no sólo de investigaciones vive el investigador, es preciso agraceder a la fuente de toda ayuda crematística que nos permita sobrevivir: el Ministerio Español de Educación y Ciencia ha sufragado toda la investigación mediante la concesión de una Beca Postdoctoral en el Extranjero y de una Beca de Reincorpora-

ción de Doctores y Tecnólogos. Algunas de las aplicaciones fueron desarrolladas con software adquirido para el «Proyecto Guadalhore», dirigido por la Profesora M.E. Aubet y subvencionado por la DGICIT (PB090-0680).

Creo que debiera dedicar este libro a los *arqueólogos automáticos* que, un día u otro, nos sustituirán en la ingrata tarea de estudiarnos a nosotros mismos. Sin embargo, como aún no existe ninguno capaz de llevar a cabo esa tarea, tendré que dejar este libro sin dedicatoria.

## Comentarios bibliográficos

Como se verá más adelante, todos los capítulos de este libro cuentan con una selección de libros y artículos de consulta recomendada. ¿Es posible sugerir un título acorde con lo dicho en esta introducción? Aunque pueda parecer sorprendente, la respuesta a esa pregunta es un rotundo sí. Y se trata de una novela: *Hermes and the Golden Thinking Machine*, de Alexander Tzonis, que podríamos traducir por *Hermes y la Dorada Máquina Pensante*. Publicada por el prestigioso Massachussets Institute of Technology (que no es, evidentemente, una editorial dedicada a la comercialización de novelas de evasión) cuenta la historia de un arqueólogo (Hermes Steganos), famoso por ser el primero en aplicar técnicas de Inteligencia Artificial en el estudio de la iconografía de la cerámica griega de pinturas negras, que vuelve a utilizarlas... para resolver el asesinato de su tío. No cuento quién es el asesino, porque para eso está la novela. Lo interesante de la misma (y por eso la publicó el MIT) es la sencillez y precisión con que explica las técnicas de Inteligencia Artificial. Si además añadimos que el asunto trata de arqueólogos y demás cacharros, su lectura pasa a ser muy recomendable.

## 1. PROBLEMAS CIENTÍFICOS

### ¿Qué es un Problema científico?

Utilizamos la palabra «problema» en distintas ocasiones de la vida cotidiana:

- como una cuestión o interrogante por resolver
- como un conjunto de hechos o circunstancias que dificultan la consecución de un objetivo
- como una proposición dirigida a averiguar el modo de obtener un resultado cuando ciertos datos son conocidos.

Todos esos usos diferentes del término parecen tener algo en común: «un problema es una dificultad que no puede resolverse automáticamente, sino que requiere una investigación conceptual o empírica». En otras palabras, nos planteamos un problema en cuanto nos hallamos en una situación en la que queremos obtener o hacer algo y no conocemos las acciones que hay que emprender para obtener lo que queremos o hacer lo que deseamos.

Si aplicásemos esta idea general al ámbito científico, tendríamos que concluir que toda interpretación, todo significado, no son más que la solución a un determinado problema. Un ejemplo clásico en nuestra disciplina sería: «¿Cuál es la cronología del objeto con el número de inventario CA-56917/85? En términos formales representaremos este interrogante por medio de la siguiente proposición:

«¿Cuál es el valor de  $x$ , para que la expresión «Cronología de CA-56917/85 = ( $x$ )» sea verdadera?

Sólo hay un modo de resolver este problema: disponiendo de un conjunto de soluciones posibles alternativas (cronologías que puede tener ese objeto arqueológico) y decidiendo cuál de ellas es la más apropiada en el caso en cuestión, según ciertos criterios bien especificados. En otras palabras,

«dado un dominio  $D$ , es necesario encontrar en un conjunto  $X$  de soluciones posibles, los elementos  $x$  que cumplan un conjunto de condiciones  $K(x)$  bien especificadas y definidas por el propio enunciado del problema».

¿Qué es  $D$ ? El tema o el dominio del saber en el cual se ha planteado el problema. Es decir, un conjunto de unidades de conocimiento tanto factuales (hechos, axiomas, hipótesis) como procedurales (reglas para producir nuevas unidades de conocimiento). Este conjunto suele denominarse *generador* o *espacio* del problema. El generador de un problema arqueológico está formado por el conjunto de hechos científicos válidos, interpretaciones posibles e hipótesis de trabajo en un dominio concreto; así, en el ejemplo cronológico anterior, el espacio del problema estaría formado por un conjunto finito de categorías, representando cada una de ellas una fase cronológica u horizonte cultural en el cual el artefacto CA-56917/85 pudiera incluirse.

El *estado inicial* de un problema coincide con lo que los lógicos denominan *explanans*, esto es, aquello que queremos interpretar. El estado inicial de un problema arqueológico es obvio: el registro arqueológico, el artefacto o la disposición de artefactos que queremos interpretar. En términos operativos, dicho estado inicial no es más que un conjunto de rasgos descriptivos. En otros dominios, la definición del estado inicial no es siempre tan evidente.

No debemos confundir el término *solución* con el término *objetivo*. El primero hace referencia al *explicandum* y suele denominarse *estado final* del problema. Al igual que el estado inicial, ese estado final no consiste en un simple predicado del tipo «siglo xx», sino en el conjunto de características de ciertos objetos del si-

glo xx: «blancos, estrechos, 8 cm. de largo, con forma de tubo, malolientes, rellenos de hojas de la planta Nicotina Tabacum finamente picadas». En otras palabras, el Estado Final de un problema está definido por todas aquellas características que diferencian esa solución precisa de las demás posibles. Cuanto más generales sean los términos que describen esa solución, tanto más amplio será su ámbito de aplicación, y responderá a más problemas.

Ahora bien, para que una unidad de conocimiento así caracterizada sea realmente una solución precisamos de un conjunto de criterios de validación o requisitos para aceptarla como válida; ese conjunto de requisitos constituye el *objetivo*, que puede ser definido en términos sencillos como: «aquel que desea obtenerse». En cierto sentido podría decirse que los objetivos son las entidades estrictamente opuestas a los *operadores*, que suelen definirse como «unidad de conocimiento necesaria para alcanzar un fin». Según esto, todo objetivo estaría representado por la detección de una «falta de información» para obtener algo, en tanto que los operadores estarían representados por el «uso» de cierta información para poder llenar ese vacío de conocimiento. Ni que decir tiene que el conocimiento al que se refieren objetivo y operador son distintos, puesto que uno depende del otro.

Usando de nuevo el ejemplo inicial,

- «calcular la cronología del artefacto CA-56917/85» es el objetivo
- «el artefacto CA-56917/85» es el estado inicial, caracterizado por: «decoración a base de motivos geométricos, altura 11,25 cm.»
- «presencia de cerámica pintada, decoración con motivos geométricos, borde exvasado, altura entre 10 y 15 cm., ...» es el operador
- «siglo IX antes de nuestra era» es la solución,

A muchos lectores puede sorprenderles esta caracterización del mecanismo de solución de un problema: ¿no habíamos definido un problema como «aquel que deseamos obtener y no sabemos cómo? Ahora resulta que ese «no sabemos cómo» está constituido, en realidad, por un conjunto de soluciones posibles, una de las cuales se convertirá en la solución idónea al final de un procedimiento específico de búsqueda y selección. Lo cierto es que no hay problema que pueda responderse si no se conocen previamente, de una manera u otra, un cierto número de soluciones; y este principio es válido en cualquier ámbito del saber, incluso en los más formalizados, como las matemáticas.

La expresión «conjunto de soluciones posibles» puede resultar confusa. Es evidente que en muchos casos -sobre todo en Arqueología- las soluciones posibles están totalmente definidas, y que el mecanismo de solución se limita a una búsqueda entre todas ellas y la selección de la mejor. Es, precisamente, esa metáfora de la «búsqueda» de solución lo que da razón de ser al símil geométrico en el término «espacio» del problema. El procedimiento o argumento utilizado para elegir una de las soluciones constituye el *operador* del problema. En otros ámbitos del saber, especialmente en las disciplinas más formalizadas, las soluciones posibles son tantas, que un operador basado en la búsqueda y selección de la mejor entre ellas no resultaría efectivo. En Matemáticas, por ejemplo, el conjunto de las soluciones posibles a una ecuación -el Generador de ese problema- está constituido por el conjunto de los Números Reales. No hay manera de construir un operador que busque en ese conjunto infinito, porque esa búsqueda no acabaría nunca. En Inteligencia Artificial denominamos a esa dificultad *explosión combinatoria*. Para evitarlo, podemos hacer que el generador coincida con el operador. Es decir, que el conjunto de soluciones no esté definido en términos de las características individuales de cada solución, sino en términos de las operaciones necesarias para «producirla» o «generarla», usando como material de construcción el estado inicial. Aunque pueda parecerlo, este procedimiento no está reñido con la necesidad de búsqueda y selección, sólo que ahora la búsqueda no tendrá lugar entre soluciones, sino entre operadores posibles.

De todo lo dicho hasta aquí se deduce que los problemas científicos deben estar enunciados, necesariamente, en un dominio concreto, puesto que sin conocimiento previo nunca habrá solución. La solución a un problema -ya sea arqueológico o propio de la vida cotidiana- surge de la manipulación de un conjunto de unidades de conocimiento, las cuales pueden llegar a ser muy numerosas. Cuanto más estructurado esté ese conjunto de unidades de conocimiento, más se parezca a una Teoría Científica y más integrado esté el operador en esa Teoría, más fácil será resolver el problema y más adecuadas las soluciones obtenidas.

Por consiguiente, puede considerarse la resolución de un problema como el conjunto de acciones sucesivas que conducen de la descripción empírica de un fenómeno arqueológico a su *interpretación o significado*. Esas «acciones» son, en realidad, la aplicación de distintos operadores cuya función es poner en contacto el

*estado inicial* (descripción del fenómeno) con el *estado final* (o interpretación considerada válida). En definitiva, para resolver un problema necesitamos:

- la definición de un espacio del problema o generador que contenga todas las interpretaciones posibles a un fenómeno dado. Ese espacio es, en realidad, una Teoría Científica acerca de ciertos fenómenos.
- la especificación de uno o más estados que sean interpretaciones aceptables (válidas) de ese fenómeno
- la especificación de un conjunto de reglas (u operadores) derivadas de la Teoría en la cual se ha definido el problema, que describan las acciones permitidas para pasar de la descripción empírica del fenómeno a su interpretación.

No obstante, no tenemos suficiente con estos requisitos; para que un problema sea soluble es preciso que esté *bien estructurado*, y para ello:

1. el conjunto de todas las soluciones posibles ( $X$ ) -o *espacio del problema*- y las condiciones de cada solución real [ $K(x)$ ] han de ser totalmente conocidas por el agente encargado de resolver el problema;
2. debe existir un criterio definido para verificar cualquier solución propuesta, y un procedimiento «mechanizable» para aplicar ese criterio;
3. en el espacio del problema ha de estar representado el estado inicial y su estado final, así como todos los operadores necesarios para ponerlos en relación;
4. los cambios o transiciones permitidos en los estados de un problema (generados por un «operador» o regla específica) pueden representarse en el espacio de estados del problema -conjunto de todas las soluciones posibles-, como transiciones entre estados sucesivos, es decir, como soluciones parciales entre el estado inicial y el estado final del problema;
5. cualquier información que el agente encargado de resolver el problema pueda usar ha de estar representada en uno o más estados del mismo;
6. todas esas condiciones exigen una cantidad de cálculos y un tiempo de computación practicables.

## ¿Cómo se resuelven los Problemas científicos?

La resolución de un problema empieza siempre con la identificación de la dificultad que debe resolverse. Una vez identificada recurrimos a la información disponible (conocimiento previo) y decidimos el punto de partida del procedimiento «mecánico» de resolución, es decir, construimos un estado inicial sobre el cual aplicaremos el operador o los operadores necesarios. A continuación estimamos la diferencia existente entre nuestro punto de partida y al punto al que queremos llegar (el objetivo), eligiendo para ello un procedimiento u operador que reduzca esa diferencia y permita que nos traslademos del punto de partida hasta el punto de llegada o solución (Fig. 1.1.).

Tanto el estado inicial como el estado final de ese proceso constituyen una *representación* del conocimiento previo: en el primer caso, el conocimiento que define la situación en la que se ha detectado una dificultad, en el segundo caso, el conocimiento que resolverá esa dificultad. En el caso de la arqueología, la situación en la que se detecta una dificultad coincide con aquello que se quiere interpretar; por ejemplo, la cronología de un registro arqueológico específico. El estado inicial está definido, obviamente, por una descripción de ese registro arqueológico, porque es allí donde identificamos la «dificultad» que deseamos resolver. El estado inicial está de un modo u otro implícito en el enunciado del problema; es decir, aparece determinado por el objetivo a resolver y por el conocimiento previo de que se dispone. «Representar» la situación inicial, por tanto, equivale a *describirla*. Representar el resultado, por su parte, exige construir un conjunto de soluciones posibles a partir de un conjunto de conocimiento previo. Como veremos a continuación, gran parte de la eficiencia del procedimiento de resolución se basa en la «construcción» de ese conjunto, es decir, en el isomorfismo entre las soluciones posibles y el conocimiento del cual proceden.

La solución a un problema es una *función* de su estado inicial; es decir, dada una dificultad concreta (objetivo), la solución *dependerá* de la situación en la que se haya identificado esa dificultad. Esta afirmación es una consecuencia evidente de la caracterización de los mecanismos de resolución como búsqueda y selección de la mejor solución posible. Por tanto, la solución concreta que obtengamos habrá de estar determinada por el estado inicial que hayamos empleado como punto de arranque del procedimiento de solución. Un ejemplo

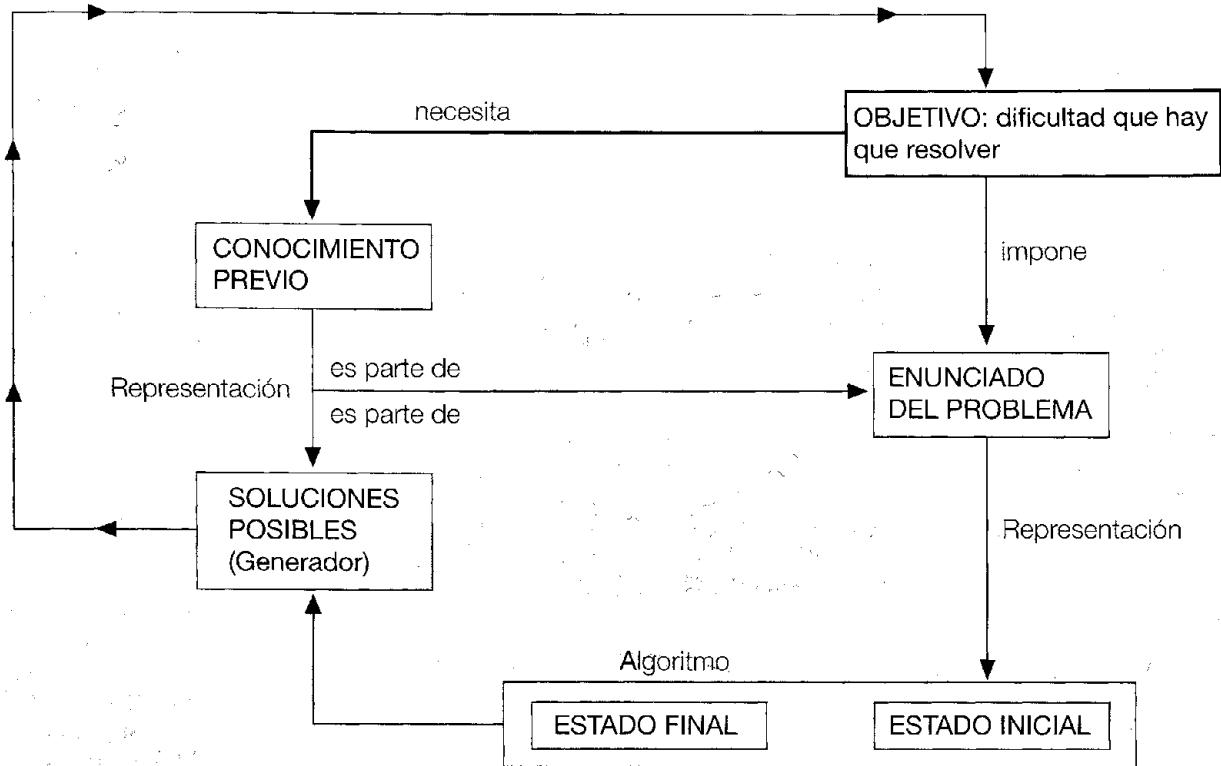


Fig. 1.1. Esquema General del Proceso de Resolución de un Problema.

trivial: supongamos una situación cualquiera en la que hemos identificado una dificultad, puede ser la cronología de una fase de ocupación concreta en un yacimiento prehistórico, o la clase social a la que debiera asignarse determinada tumba. Para resolver cualquiera de esos problemas debemos encontrar una unidad de conocimiento particular que *dependa* de la dificultad identificada, es decir, las categorías «Siglo IX» ó «Caudillo Tribal», pongamos por caso, serían soluciones apropiadas de los estados iniciales «Fase de Ocupación B» ó «Tumba 75», si y sólo si existiese algún tipo de dependencia entre ambas.

El diccionario define la relación de dependencia como «subordinación» o «consecuencia» :

**B** depende de **A** si **B** es consecuencia de **A**.

A su vez, el término «consecuencia» suele definirse como «proposición que deriva de otra». Dicho de otro modo, la solución depende del estado inicial porque ha sido producida (o elegida) *a partir de* un estado inicial. Al caracterizar el procedimiento de resolver un problema como búsqueda y selección de una entre varias soluciones posibles, estamos dando a entender que la solución a un problema es una entidad *virtual* -no existente- hasta la formulación de un estado inicial capaz de seleccionarla: el estado final existe porque existe un estado inicial *capaz de seleccionarla*. Ahora bien, esta capacidad de selección no sólo depende del estado inicial mismo, sino del operador que va a ponerlo en relación con la solución. Es precisamente el hecho de aplicar un operador lo que estable la «función» o dependencia en cuestión.

Concluiremos, entonces, que la relación existente entre estado inicial y solución coincide con la aplicación de un operador. En realidad, ese operador se limita a *representar* la función entre ambos, traduciéndola a un formato ejecutable, esto es, como una secuencia ordenada de reglas, acciones e instrucciones. Esta relación (y el operador que la representa) es un elemento de conocimiento científico, que forma parte del conjunto de conocimiento en el que se ha enunciado el problema, el estado inicial y el conjunto de soluciones posibles.

Esta relación entre estado inicial y estado final, y que hemos definido, a falta de un término mejor, como de «dependencia», no tiene por qué ser una relación formal o matemática. Al menos no siempre. El objetivo de este libro es precisamente analizar la gran diversidad de funciones o relaciones posibles existentes entre un estado inicial y una solución posible, la mayoría de ellas incluidas en el mal llamado y peor comprendido «sentido común». Las cinco modalidades de operador antes enunciadas (Correspondencia, Simulación, Ordenación, Transformación, Formal) ya nos indican la extrema diversidad de los mismos. Esas relaciones tampoco son «evidentes» o fáciles de establecer. No olvidemos que los problemas surgen de la identificación de cierta dificultad; por lo tanto, la relación entre un estado inicial y un estado final no es nunca obvia, sino no se llamaría dificultad. Como hemos descrito el procedimiento de resolver problemas en tanto que una búsqueda y selección del mejor resultado de entre un conjunto de soluciones posibles, el punto clave en la resolución radica en determinar si existe o no alguna relación entre los datos iniciales y la solución propuesta.

En la vida cotidiana, pocas veces nos molestamos en definir esa relación, pues actuamos por ensayo y error o bien recurrimos a nuestra experiencia: si la situación es semejante a alguna antes experimentada, actuaremos de igual forma que lo hicimos en aquel momento. La resolución de problemas científicos suele estar basada muchas veces en el ensayo y error, si bien lo más aconsejable sería establecer la relación entre datos iniciales y resultado mediante:

- experimentación
- deducción
- aprendizaje

A nivel de la resolución del problema, es decir, una vez que se ha obtenido por experimentación, deducción o aprendizaje una lista de posibles operadores, se procede de nuevo por búsqueda y selección del más idóneo entre ellos, esto es, de aquel que mejor reproduzca la relación existente entre el estado inicial y el estado final que resolverá el problema. En definitiva, para poder buscar y seleccionar una solución, es preciso que antes busquemos y seleccionemos un operador, operación para la que necesitaremos, también, algún tipo de conocimiento previo que nos permita llevar a cabo esa «suboperación» de búsqueda de la relación u operador.

Si lo que se desea es alcanzar cierto objetivo (resolver un problema), debe formularse la tarea de alguna manera, esto es, las diversas acciones (operadores) han de disponerse de algún modo racionalmente ligado al objetivo. Y esa ordenación sólo puede llevarse a cabo recurriendo al conocimiento disponible. Incluso el hecho de buscar el conocimiento necesario para disponer el conjunto de acciones necesarios forma parte de las acciones que satisfarán el objetivo. Cuanto más problemática sea la solución, menos conocimiento estará disponible, y más difícil resultará su acceso. En este sentido «problemático» significa «pobre en conocimiento».

## Descomposición de un Problema

Aceptemos, pues, que «establecer» una relación cualquiera entre el estado inicial y uno de los estados finales posibles equivale a resolver definitivamente un problema. La imaginación y la creatividad, tantas veces exigida por los filósofos críticos de la Inteligencia Artificial, radica precisamente en ese carácter laxo y muy libre de la relación. Lo que importa es que haya algún tipo de asociación, no que esa asociación siga un criterio pre establecido o muy riguroso formalmente. Obviamente, cuanto más «formalizable» sea la relación, mayor *validez* otorgaremos a la solución.

Consideremos el siguiente ejemplo: «¿Cuál es la cronología de la espada de bronce encontrada en el yacimiento (A)?». Dado que la situación inicial ya está definida (una descripción de la espada), pues es ella la que nos ha sugerido la necesidad de resolver una dificultad, lo primero que tendremos que hacer es deducir del conocimiento a nuestra disposición (fruto de nuestra experiencia o de lecturas sobre la materia) el conjunto de soluciones posibles. Definimos la relación entre el estado inicial y una de entre todas esas soluciones alternativas como una asociación simple: la solución elegida ha de contener entre sus características definitorias algunos de los rasgos descriptivos del objeto. Por ejemplo, el estado inicial «espada de bronce caracterizada por la forma peculiar de su hoja y pomo, la primera estrangulada en su tercio superior, y el segundo acabado en cola de pez» se asociará con aquella de las soluciones que contengan entre sus características la expresión «Hoja estrangulada en su tercio superior y pomo en forma de cola de pez».

Este problema no podrá ser resuelto si:

- no hemos encontrado ningún término de comparación entre estado inicial y estado final, esto es, ninguna de las soluciones posibles mantiene relación de dependencia alguna con el estado inicial.

Por ejemplo, si ninguna solución contuviese esa mención a las espadas con un tipo de hoja y pomo determinado.

- hemos encontrado más de una solución posible que comparte características con ese estado inicial, y no sabemos cual de ellas elegir como «mejor» solución. Las soluciones «Siglo ix», «Siglo x» y «Siglo xiv» contienen la característica «presencia de espadas con hoja estrangulada en su tercio superior y pomo en forma de cola de pez». ¿Cuál de las tres soluciones, siglos ix, x ó xiv es la correcta?

En el primer caso, existe incompatibilidad entre el estado inicial y el estado final, y por tanto no puede encontrarse el operador necesario; en el segundo, el único operador disponible proporciona demasiadas soluciones opuestas entre sí (una espada no puede darse simultáneamente en tres épocas distintas). En ambos casos aparece la necesidad de *descomponer* el problema.

En otras palabras, los «fallos» en la resolución dan lugar a que nos planteemos *subproblemas* cuyo objetivo es, precisamente, resolver la nueva dificultad aparecida. En el caso que aquí nos ocupa ese «subobjetivo» consiste en reducir la distancia entre el estado inicial y alguna de las soluciones, es decir, buscando un estado intermedio, relacionado a su vez con la descripción de la espada y con la solución idónea y que haga «de puente» entre uno y otro (Fig. 1.2.). La detección de una unidad de conocimiento entre los datos iniciales y la solución permitiría descomponer la aplicación del operador, y con ello el establecimiento de la relación entre ambas, en dos acciones sucesivas: un primer operador entre el estado inicial y el estado intermedio y un segundo -distinto probablemente del primero- entre ese estado intermedio y el estado final.

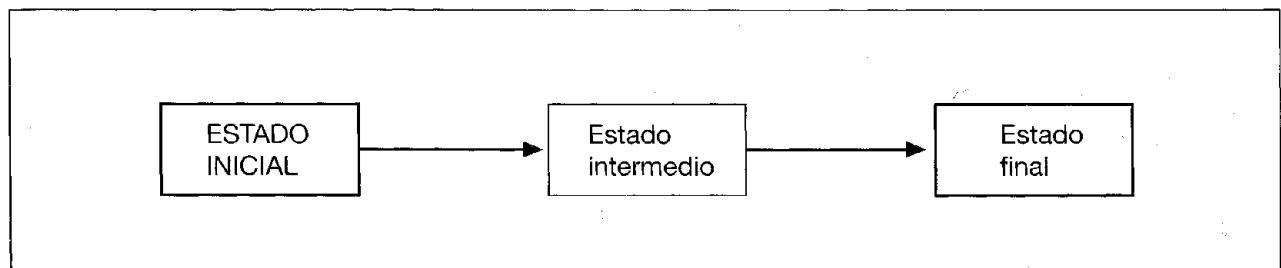


Fig. 1.2. Elementos en los que se descompone un problema.

La descomposición del problema constituye, a su vez, un problema, cuyo propósito será buscar y seleccionar un estado intermedio que reduzca la distancia entre el Estado Inicial y el Estado Final. Esa búsqueda tendrá lugar en un espacio de estados intermedios posibles, obviamente distinto del espacio de soluciones alternativas, por consiguiente, el espacio del problema estará constituido, no sólo por el conjunto de las soluciones, sino también por el conjunto de todas las descomposiciones posibles. El procedimiento de descomposición puede esquematizarse, entonces, del siguiente modo:

#### *Descomponer el problema*

Si se desea alcanzar el objetivo G  
y no se cumple la condición previa C  
Entonces, alcance previamente el subobjetivo C'.

Los subobjetivos son unidades de conocimiento preespecificadas cuyo objetivo es, precisamente, resolver los errores de ejecución. Se trata de un conocimiento acerca del procedimiento para resolver el problema antes que acerca de la solución correcta del problema; nos referiremos a él como *meta-conocimiento*. Es importante destacar que, sin ese meta-conocimiento, no puede haber solución en el caso de problemas medianamente complejos.

La aparición de errores de ejecución durante la resolución de un problema depende de los fallos en el mecanismo de decisión del operador más conveniente o de la solución más adecuada. Podemos encontrarnos, entre otros, con los siguientes:

- la aplicación de un operador impide decidir cual es la mejor solución, ya que el espacio del proble-

ma (la diversidad de soluciones alternativas) es demasiado grande: varias soluciones posibles son adecuadas para ese estado inicial,

- la aplicación de un operador no da lugar a solución , pues ninguna de las soluciones existentes en el espacio del problema parece mantener relación alguna con su estado inicial,
- el único operador aplicable indica que el estado inicial no es el adecuado en ese espacio del problema (la descripción del ajuar de una tumba no nos permite responder a preguntas metafísicas acerca de la vida y la muerte),
- puede registrarse un conflicto entre dos o más operadores aplicables que proporcionan soluciones opuestas.

Todos esos errores de ejecución se resuelven por medio de búsqueda y selección de conocimiento en un subespacio específico. Es decir, el agente que resuelva el problema debe tener acceso a una fuente de información alternativa en la cual encontrará el conocimiento que permitirá resolver la dificultad y alcanzar el objetivo primario. Cada uno de los cuatro tipos de error da lugar a un subproblema peculiar, con un subespacio propio, que puede contener, según sea el caso, nuevos operadores, nuevos estados iniciales, etc.

En resumen, para poder resolverlo, todo problema precisa de un sistema de subespacios asociados, cada uno de ellos particular a un subobjetivo específico. La apariencia externa de ese sistema es la de una compleja red en la que están integrados todos los subproblemas posibles, unidos entre sí por medio de relaciones de «cooperación», pues cada subproblema, con ayuda de los demás contribuye a encontrar la, solución idónea al estado inicial . Veamos un ejemplo. La Figura 1.3. representa un problema que requiere tres descomposiciones (o tres subproblemas, como queramos decirlo) para poder evaluar la relación de dependencia entre el estado inicial y el estado final.

#### Problema General

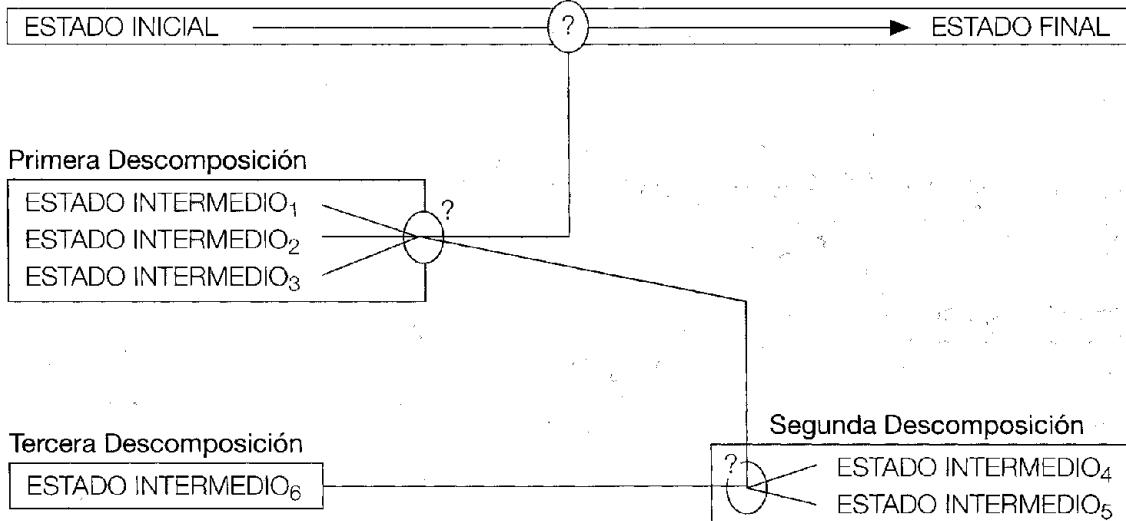


Fig. 1.3. Ejemplo de Árbol de descomposición asociado a un problema arqueológico.

El problema general es el siguiente:

«Cuál es la cronología de la espada de bronce hallada en el yacimiento (A)?»

El estado inicial está constituido por la descripción morfométrica de dicha espada, y las soluciones posibles (cronologías) están expresadas en términos de la cerámica característica en cada una de las sucesivas fases cronológicas:

- cerámica con decoración pintada (SIGLO ix)
- cerámica con decoración incisa (SIGLO viii)
- cerámica con decoración excisa (SIGLO vii)

En este caso existe, obviamente, incompatibilidad entre el estado inicial y los tres estados finales posibles. Para resolverla habremos de descomponer el problema definiendo un primer subobjetivo:

*¿Qué cerámica está asociada a la espada de bronce encontrada en el yacimiento (A)?*

Ahora bien, el operador «lugar de hallazgo de la espada» no es aplicable porque desgraciadamente desconocemos dónde se encontró la espada. Pertenece a la colección privada de un aficionado local y nadie se acuerda de cuándo y dónde se desenterró. Los estados intermedios posibles son tres, uno por cada tipo de cerámica que pudiera haberse encontrado junto a la espada. Necesitamos de una nueva descomposición para poder elegir una de las tres soluciones. Recurrirímos ahora a un operador analógico:

*¿Qué espada morfométricamente semejante a la del yacimiento (A) ha sido encontrada asociada a uno de los tres tipos anteriores?*

Supongamos que disponemos de dos ejemplos, una espada asociada a cerámica pintada en el yacimiento (B) y otra asociada a cerámica incisa en el yacimiento (C), siendo las espadas halladas en A, B y C semejantes entre sí. También se repite el problema de la decisión, por lo que plantearemos un tercer subproblema, que nos proporcionará la información necesaria para resolver el segundo problema:

*¿Cuál de las espadas encontradas en (B) y en (C) es más semejante -en términos de similaridad matemática- a la encontrada en (A)?*

La respuesta a este subproblema es única: sólo la espada encontrada en (A) es más semejante a la encontrada en (B) que a la encontrada en (C). La solución a este subproblema contribuirá a solucionar la segunda («Espada encontrada en (B) asociada a cerámica pintada») y primera descomposición («Espada encontrada en (A) asociada a cerámica pintada») hasta llegar al problema general, cuya solución será:

*La espada de bronce encontrada en el yacimiento A puede datarse  
en el siglo IX.*

La selección de los subproblemas y de los subespacios en los cuales deberemos buscar los estados intermedios es abordada por las *estructuras de control del problema*, o conjunto de operadores encargados de gestionar el meta-conocimiento. Esos operadores, que regulan las transiciones entre subproblema y subproblema y controlan la «colaboración» entre todos ellos son en realidad «meta-operadores», ya que utilizan unidades de «meta-conocimiento». Operadores de ese tipo serán, por ejemplo:

- Seleccionar el Subespacio más conveniente para solucionar la dificultad (x).
- Crear un nuevo estado inicial a partir de las características concretas de la dificultad encontrada.
- Buscar de entre todos los posibles el estado intermedio que más se ajuste a la incompatibilidad detectada

La Descomposición de los problemas es un tipo de inferencia capital para poder encontrar soluciones en problemas mínimamente complejos. Es más, me atrevería a decir, incluso, que un problema que no requiere descomposición en subproblemas proporciona soluciones «triviales». Hay que tener siempre en cuenta, por tanto, que la relación [estado inicial/estado final] no tiene por qué establecerse *directamente* entre el estado inicial y el estado final; una simple aplicación del operador sobre el estado inicial no tiene por qué conducirnos a la solución. No debiéramos extrañarnos si los problemas arqueológicos requieren un centenar o más de descomposiciones.

Los arqueólogos hemos experimentado esta situación en demasiadas ocasiones, cuando los objetivos a alcanzar están muy alejados de la naturaleza fragmentaria e incompleta de los restos arqueológicos; los esfuerzos por construir Teorías de Alcance Medio (Binford 1983, Raab y Goodyear 1984, Torrence 1986, entre otros) se enmarcan en este contexto de «aproximar» el estado inicial al estado final mediante la adición controlada de conocimiento. El espacio de resolución resultante es tan complejo que no hay investigador que pueda establecer procedimientos de búsqueda y selección en él. Esa es la justificación principal de este libro: los ordenadores y las técnicas de Inteligencia Artificial son los instrumentos por excelencia para resolver problemas complejos en la disciplina que sea.

## Comentarios bibliográficos

El primer gran clásico acerca del tema de la resolución de problemas es el libro de Polya (1945). Con posterioridad el tema fue completamente reformulado por Newell y Simon (1972) y por Sacerdoti (1977), trabajos en los que se fundamenta la Teoría moderna de la resolución de problemas. Obras de referencia actualizadas son los libros de Pearl (1985), Laurière (1986), Torasso y Console (1988), Brown y Chandrasekaran (1989), Gilhooly (1990).

## 2. ORDENADORES Y CEREBROS: UNA INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL

### Solución a un problema en la mente humana y en el ordenador

El capítulo anterior puede haber dado la impresión al lector que la resolución de problemas es muy simple y esencialmente mecánica. Nada más lejos de la realidad. Supongo que todos tenemos la misma experiencia en darnos de cabeza contra problemas insolubles, tanto en la vida cotidiana como en la práctica profesional. Mi propósito en este capítulo es mostrar algo de la complejidad asociada a la resolución de cualquier problema, aunque se trate del más sencillo. El aspecto «mecánico» en el que he insistido en el capítulo anterior y en el que insistiré a lo largo de este libro se debe tan sólo a la necesidad de programar un ordenador para que sea capaz de simular, aunque sea sólo en parte, la manera en que los científicos dan soluciones a sus problemas.

El estudio de cómo los seres humanos resuelven los problemas que se les plantean en la vida cotidiana suele confundirse con el estudio de los mecanismos de razonamiento. Usualmente, se diferencian ambos dominios por la especificidad de los mecanismos psicológicos de la solución de problemas, comparada con los mecanismos, mucho más generales, del razonamiento humano. En capítulos posteriores analizaremos esas diferencias, así como su relevancia en la metodología de las Ciencias Sociales; por ahora nos limitaremos a estudiar la «psicología» o contenido «cognitivo» de la resolución de problemas.

Según algunos autores las características de la memoria humana y de sus mecanismos de funcionamiento determinan que el espacio del problema y la búsqueda entre los estados que constituyen ese espacio, sea un rasgo invariante en todo mecanismo de solución de problemas; esto es, que cualquier resolución de un problema tiene lugar en el espacio del mismo» (Newell y Simon 1972: 789). Aceptemos, momentáneamente, este punto de partida, dado que se considera que es un resultado empírico: la casi totalidad de los estudios psicológicos revelarían que los individuos resuelven los problemas de ese modo (más adelante veremos que eso sólo es cierto en parte).

¿Qué son, psicológicamente hablando, los «estados» de un problema y por qué configuran un «espacio», esto es, un conjunto más o menos estructurado? Cualquiera de los estados de un problema representa lo que un sujeto sabe acerca del problema en un momento dado; por «saber» se entiende aquí la información que existe en la memoria y a la que se puede acceder. Los distintos estados de un problema son, pues, partículas de conocimiento específicas, llamadas *conceptos*, los cuales están compuestos, a su vez, por unidades de información elementales, a las que denominaremos *símbolos*. Consideraremos, además, que todo el conocimiento existente en la memoria tiene una apariencia *relacional*, ya que está expresado en términos de los símbolos que lo definen y de las operaciones mentales necesarias para «activar» esos símbolos, esto es, para acceder a la información almacenada en la memoria.

Vimos en el capítulo anterior que la resolución de problemas se representaba en términos de estados sucesivos y de los operadores necesarios para «buscarlos» o «definirlos» en un espacio previamente especificado. Usaremos idéntico modelo para referirnos a la resolución de los problemas en la mente humana, sólo que ahora el espacio del problema está configurado por el conjunto estructurado de conceptos existentes en la memoria del sujeto, y la manera en que se accede a ellos equivaldrá a los operadores para pasar de estado a estado. Dado que los conceptos están definidos en función de los símbolos empleados para su activación, la resolución de un problema podrá representarse mediante una secuencia finita de operaciones elementales con símbolos. Esas operaciones elementales se refieren a la «activación» o acceso a la información, almacenada en la memoria en forma relacional.

Las operaciones con las unidades mínimas de conocimiento pueden ser inconscientes; en otras palabras, no sabemos cómo hemos producido la solución, si bien esa solución parece *obvia*. Este hecho se debe, esencialmente, a la extrema simplicidad de esas operaciones (normalmente asociaciones por similaridad o analogía), y simultáneamente a la gran cantidad de ellas que se necesita para resolver el problema. La cantidad de símbolos necesarios para activar un concepto (un estado del problema) puede ser bastante grande, aún cuando cada uno de los símbolos que lo componen sea relativamente específico. La especificidad de esas subunidades de conocimiento implica que exista una cantidad inmensa de ellas en la memoria, por lo tanto, si el mecanismo de resolución fuese una mera búsqueda y decisión por «ensayo y error» entre todas las informaciones disponibles, ni el sujeto más inteligente podría dar una respuesta en un tiempo prudencial.

Para resolver esa dificultad, los psicólogos han diseñado un modelo de la memoria humana dividido en dos componentes, de manera que la resolución de un problema pueda considerarse como un intercambio entre una *Memoria a Largo Plazo*, en la que se encuentra almacenada *toda* la información que un individuo ha ido recogiendo a lo largo de su experiencia, y una *Memoria a Corto Plazo*. En realidad, ésta no es más que una parte de la Memoria a Largo Plazo, la *memoria activa* en un momento dado (Fig. 2.1.)

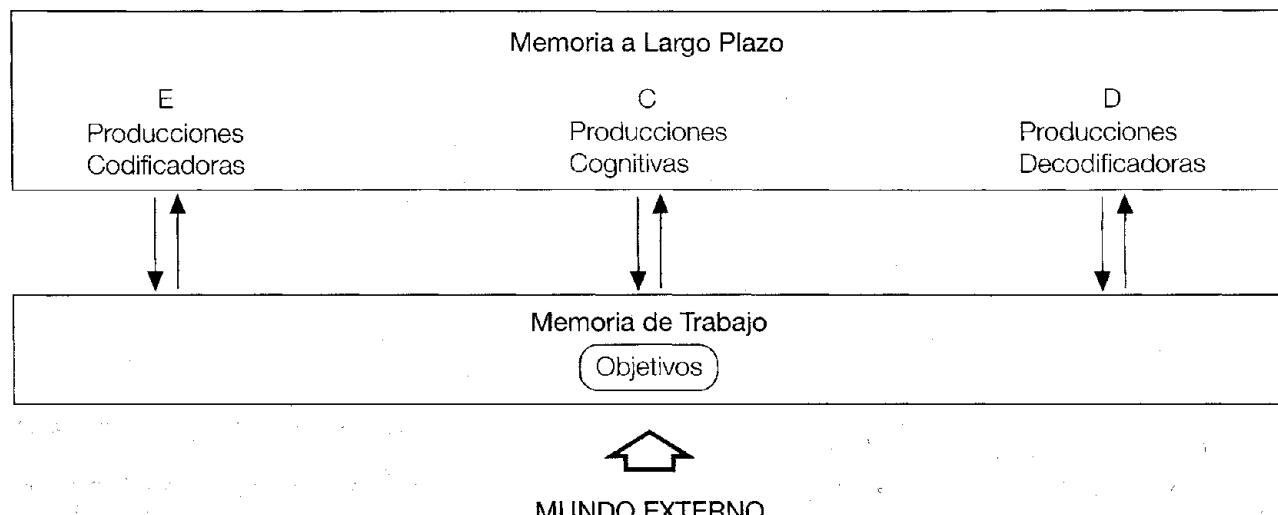


Fig. 2.1. Operaciones de transferencia de conocimiento entre una Memoria de trabajo y una Memoria a largo plazo no especializada.

Debido a su tamaño, la Memoria a Largo Plazo no es fácilmente manejable; por su parte, el pequeño tamaño de la Memoria a Corto Plazo permite utilizarla como memoria de trabajo, esto es, el lugar en el que se van a llevar a cabo las operaciones de resolución. Si esa resolución precisa la utilización de ciertas unidades de información contenidas en la Memoria a Largo Plazo, entonces se empleará algún procedimiento que permita trasladarla temporalmente a la memoria de trabajo o *memoria activa*; si ese procedimiento no consigue localizar la información, entonces ninguna solución tendrá lugar. Al decir de los psicólogos la transferencia entre una y otra memoria se efectúa por *asociación*, lo cual se debe, esencialmente, a la representación «relacional» del conocimiento en la memoria.

Así pues, para obtener una solución sin incurrir en la innumerabilidad del conjunto de soluciones posibles (*explosión combinatoria*), es preciso extraer de la Memoria a Largo Plazo un conjunto delimitado de unidades de conocimiento y situarlo en la Memoria a Corto Plazo, en la cual esas unidades serán manipuladas de acuerdo con la representación subjetiva del problema que se haya hecho el agente encargado de resolverlo. Es decir, el sujeto humano suele emplear partes muy pequeñas de las unidades de conocimiento que conserva en su Memoria a Largo Plazo para responder a los estímulos del mundo externo. Nunca se aborda, al menos no al mismo tiempo, la complejidad del entorno con ayuda de la totalidad de conocimientos que se han adquirido a lo largo de toda una vida: los espacios del problema han de almacenarse en la misma memoria activa encar-

gada de procesar los estímulos (esto es, resolver el problema), lo cual supone siempre una selección de la información contenida en la Memoria a Largo Plazo.

Simon (1973) ha propuesto aplicar este mecanismo general para explicar cómo la mente humana es capaz de resolver tanto los Problemas Bien Estructurados, como los Problemas Mal Estructurados. Los primeros son aquellos problemas en cuyo espacio pueden especificarse claramente los pasos (o la secuencia de operadores) para llegar a la solución. La dificultad para resolver estos problemas radica exclusivamente en la coordinación de la secuencia de operadores, pues no se precisa de ningún subespacio en el cual buscar el operador necesario.

Se suele decir que los Problemas Mal Estructurados exigen una o dos intuiciones acerca de la entrada del problema (estado inicial) a fin de obtener la solución; esa «intuición» puede representarse por medio de un procedimiento de búsqueda de los operadores en un subespacio estructurado por medio de la imposición de una serie de condiciones o *heurísticas* que progresivamente limitarán la cantidad de operadores posibles y que, finalmente, permitirán elegir el más adecuado. El procedimiento es aparentemente simple: en cada una de las etapas de la resolución se compara el estado actual del problema con el estado final del mismo (si es conocido) o con las condiciones previamente enunciadas que debe cumplir ese estado para poder ser considerado la solución; a continuación se lleva a cabo una operación que permita reducir la diferencia entre el estado actual y las características del estado final. El uso de este procedimiento heurístico permite al agente encargado de resolver el problema avanzar reduciendo progresivamente la distancia a la solución y evitando el uso de estados del problema que aumenten esa diferencia.

La idea de *heurística* aplicada al razonamiento científico no tiene nada de insólito. Tal y como la ha propuesto H. Simon, son ciertas HIPÓTESIS VEROSEMILES las que organizan el proceso de resolución de un problema científico, reduciendo el espacio de dicho problema y la búsqueda en él de un estado final, si éste es desconocido. Si el estado inicial fuese, por ejemplo, la descripción de un yacimiento arqueológico y de los objetos que en él han aparecido, los estados posibles del problema arqueológico son infinitos, con lo que nunca se llegaría a una solución. Mas si proponemos una HIPÓTESIS, esto es, una Función de la Cultura Material en ese yacimiento y su Contexto de uso habríamos limitado drásticamente el conjunto de interpretaciones posibles. La tarea sería entonces *construir* el mecanismo de conexión entre el estado inicial y el estado final que proponemos. La diferencia con la visión positivista de la demostración de hipótesis radica, precisamente, en la «plausibilidad», antes que en la «probabilidad» de esa construcción: no descubrimos una conexión real, sino que proponemos una demostración plausible. En el momento en que cambiemos las condiciones que hemos impuesto en el espacio del problema, el resultado será otro, impredecible.

En resumen, según Newell y Simon, para resolver problemas los seres humanos utilizan la información que son capaces de extraer de la estructura particular del espacio de un problema, evaluando gracias a ella las informaciones que hay que buscar en la Memoria a Largo Plazo y trasladar a la memoria activa, así como los operadores necesarios para realizar esas tareas. Si la información que hemos introducido en la memoria activa es demasiado compleja, o bien, no concuerda con la representación que el sujeto se ha hecho del objetivo del problema, habrá un exceso de información inútil, que ralentizará la resolución y, aún, podrá entorpecerla. Si la información fuese menor, sin embargo, nunca se encontrará la solución por falta de información.

## **Heurística**

Una búsqueda heurística es un procedimiento que aplica pasos plausibles para navegar en un espacio de alternativas inmenso, con el fin de conocer la mejor (o una colección de las mejores) alternativas para cierto propósito. Lo que convierte a esta búsqueda en *heurística* es que el procedimiento no garantiza que el resultado sea la mejor de todas las alternativas, o una colección que incluya la mejor alternativa, si bien el procedimiento se aproximará bastante a ella, de acuerdo con cierto criterio. Ese criterio puede que no sea muy riguroso y que no siempre proporcione una solución óptima, si bien ésta será la mejor que se puede obtener, muy superior a la que se habría logrado sin una búsqueda heurística. Se insiste en aquello que es alcanzable y suficientemente bueno, prescindiendo de lo que es óptimo, pero inalcanzable.

La racionalidad científica consiste, precisamente, en usar los mejores medios que permiten reducir la búsqueda de una solución *aceptable* a unas proporciones manejables. Esos medios heurísticos sugieren qué operadores hay que aplicar en cada momento, cuál es el estado del problema que más nos acerca a la solución y cómo *generar* ese estado -esto es, esa unidad de información particular- si no la conocemos previamente, cómo

han de evaluarse los estados sucesivos de un problema,... Es decir, la forma de utilizar la información necesaria para resolver un problema está controlada, de un modo u otro, por lo que sabemos acerca de ese problema. Esa es la única forma de poder llegar a una solución en un plazo de tiempo razonable: el conocimiento del problema nos permite estipular las condiciones que han de cumplir los estados sucesivos del problema para llegar a ser una solución aceptable al mismo. Si la información disponible es muy escasa, entonces se usarán ciertos conocimientos generales acerca de la solución más aconsejable para proponer nuevas hipótesis (forzosamente muy generales en un primer momento) y contrastarlas (parcialmente).

J.L. Laurière (1986) esquematiza del siguiente modo el procedimiento heurístico de solución:

#### *Primer paso*

Elegir una acción de entre todas las posibles. Es decir, elegir los operadores que nos han de llevar de un estado a otro. Ello se puede hacer:

- al azar
- por concordancia al objetivo propuesto al plantear el problema
- por experiencia, repitiendo acciones previamente realizadas, o reconociendo acciones clave.
- por condición necesaria:
  - obligados por nuestro análisis de la situación
  - obligados porque alguna solución no es contrastable

#### *Segundo paso*

Se lleva a cabo la acción elegida y se modifica la situación inicial.

#### *Tercer paso*

Evaluación de la nueva situación (nuevo estado del problema):

- por analogía a un problema anterior o con otro estado de ese mismo problema antes evaluado
- por criterio matemático, usando una función de evaluación que relacione los estados del problema con un conjunto de cifras, las cuales se usarán para determinar el operador siguiente.
- por beneficio: esperanza de salir ganando con el nuevo estado del problema alcanzado, ya sea porque se trata de una situación más simple o porque permite una reducción significativa en la cantidad de investigación necesaria.

#### *Cuarto paso*

Se rechazan las situaciones o estados del problema que sean inútiles para la consecución del objetivo

#### *Quinto paso*

Fin. Si la evaluación de la situación permite afirmar que ése es el estado final del problema, se detiene el procedimiento. Sino, se siguen aplicando informaciones heurísticas para alcanzar ese estado final.

Es evidente que la complejidad del procedimiento heurístico estará relacionada con la complejidad del problema a resolver. Si disponemos de mucha información acerca del mismo, esto es, el espacio del problema es muy grande debido al número de estados conocidos y accesibles, los métodos heurísticos serán muy concretos (relativos exclusivamente al dominio en el cual se ha planteado el problema) y se referirán a la elección de los estados más prometedores para alcanzar la «mejor» solución, de acuerdo con un criterio dado. Si, por el contrario, el problema científico es bastante nuevo y se tienen muy pocas informaciones que permitan alcanzar la solución final, los procedimientos heurísticos serán muy generales (es decir, se basarán en reglas sintácticas más o menos universales para obtener nuevos estados del problema a partir de los ya conocidos), lo cual irá en detrimento de su idoneidad al problema que nos interesa.

En esos casos, mientras que un operador genera las diversas acciones en el espacio del problema (los movimientos o *inferencias* de un estado a otro), otro operador comprueba si se ha alcanzado una situación final. Según la estructura del generador, la búsqueda de la solución será puramente aleatoria o heurística. En el caso de Problemas Mal Estructurados, esos generadores tenderán más hacia la búsqueda aleatoria (así se podrá ampliar al máximo el espacio del problema); los operadores de comprobación, por su parte, aplicarán criterios heurísticos para evaluar las hipótesis libremente producidas. En realidad, es así como suele actuar el científico humano cuando el dominio que investiga es prácticamente desconocido: proponiendo un gran número de hi-

pótesis muy generales y contrastándolas con ayuda de unos criterios muy concretos, específicos a un único objetivo a la vez. Obviamente, cuanto mejor especificados estén los propósitos finales (los criterios que permiten saber si una solución es apropiada para algo en particular), mejores serán las soluciones, por muy libre o poco estructurado heurísticamente que sea el generador de hipótesis.

En Arqueología el término heurística ha sido utilizado de una forma algo distinta (Kintigh y Ammerman 1982, Simek, Ammerman y Kintigh 1985): supongamos que hemos realizado la excavación de un determinado sector en un yacimiento prehistórico, y asumimos que todos los artefactos (por ejemplo, útiles líticos) encontrados en ese nivel arqueológico son contemporáneos, y forman parte del mismo nivel de ocupación. De lo que se trata es de averiguar la asociación espacial de esos objetos, es decir, si en ausencia de vestigios arquitectónicos u otros es posible distinguir áreas diferenciadas usando como información (estado inicial del problema) las coordenadas **x** e **y** de cada uno de los artefactos. La cantidad de agrupaciones estadísticamente significativas es extraordinariamente grande (una por objeto triangulado), por lo que se requiere de un procedimiento heurístico para poder elegir de entre todas las soluciones posibles, la más verosímil.

Para ello se utiliza un procedimiento estadístico de agrupación, el análisis de conglomerados por *k*-medias. Este procedimiento, sin embargo, exige que el usuario defina el número de agrupaciones (*k*), calculándose entonces la *mejor* de todas las divisiones posibles con esa cantidad de clases. El procedimiento estadístico no usa ningún criterio heurístico, se limita a calcular todas las divisiones posibles en los *k*-grupos definidos por el usuario y elegir aquella en la que la varianza intra-grupo aparece minimizada. El problema radica no en la parte matemática, sino en la obligación del analista de definir *a priori* el número de agrupaciones y en el significado «arqueológico» de la división obtenida. Kintigh, Ammerman y Simek proponen repetir el análisis numerosas veces, solicitando en cada caso un número distinto de clases. Así, se obtendrán, por ejemplo, 15 divisiones, una con una clase, otra con dos, otra con tres, y así sucesivamente hasta llegar a la solución con 15 clases. A continuación proponen una completa batería de tests para elegir la mejor de esas soluciones. Aunque estadísticos, esos tests son claramente *heurísticos*: ni el error de la suma de cuadrados, ni la diversidad interna de cada grupo son criterios absolutos (como sí lo era la minimización de la varianza en el algoritmo de la división), sino índices comparativos que debiera permitir al arqueólogo considerar «mejor» la división en 9 clases que la división en 11, siendo posible, incluso, que ambas soluciones sean correctas, cada una en su propia escala.

Ahora bien, reducir la complejidad de un problema no es lo mismo que buscar «atajos» verosímiles en el espacio de un problema; en ese segundo caso, lo que simplificamos es el procedimiento de solución, no la construcción del espacio, que sigue siendo muy complejo. Si redujésemos el tamaño o simplificásemos la estructura relacional del espacio del problema estaríamos sustituyendo un problema complejo, por otro más sencillo que *no es equivalente al primero, aunque intuitivamente esté relacionado con él*. Una respuesta adecuada al problema simplificado, no tiene por qué ser una respuesta adecuada al problema original del cual hemos derivado la simplificación.

El ejemplo más conocido de simplificación del problema, y el que más discusiones ha generado entre arqueólogos (los post-procesualistas en primera línea!) es el de la descontextualización: ningún fenómeno social es independiente o se explica a sí mismo. Es precisamente esta «dependencia» mutua de todos los fenómenos sociales la que explica la enorme complejidad de cualquier problema social, complejidad aún mayor si le añadimos la polémica acerca del «contexto» de la observación, es decir, la alteración consciente o inconsciente que el analista introduce en el fenómeno que pretende estudiar. Un procedimiento heurístico no debiera descontextualizar los problemas, independizar fenómenos que no se explican a sí mismos, ni evitar la influencia del observador con la excusa de simplificar el problema y posibilitar *la solución*. En este estado de cosas, heurística quiere decir, tan sólo, que el analista es consciente de la complejidad, y, sin reducirla, simplifica la solución, prefiriendo una respuesta meramente *válida*, a la respuesta *definitiva*, a la cual quizás se llegue en un momento posterior de la historia de la disciplina.

A lo largo de este libro se irán proponiendo numerosos ejemplos no sólo de algunas de las heurísticas más apropiadas en la investigación arqueológica, sino también acerca de la manera de usar esas heurísticas. Por ahora, lo más importante es retener el principio general: es imprescindible el uso de una información adicional acerca de la estructura particular y el alcance del problema, para poder generar y acceder a la información que nos permitirá resolverlo.

## Reglas de Producción

De todo lo dicho hasta aquí se desprende que la solución a un problema implica la manipulación del conocimiento. Sin embargo, aún no nos hemos referido a la manera en que eso puede hacerse en la práctica: se ha mostrado la forma de llegar hasta una solución (empleando condiciones heurísticas), pero todavía no la manera de programar un ordenador para que haga lo mismo (o *simule* que hace lo mismo) que el cerebro humano.

Muchos psicólogos (cf. entre otros Newell y Simon 1972, Anderson 1983, Holland et al. 1986, Johnson-Laird 1988, Newell 1990) han propuesto representar la asociación entre unidades de conocimiento (esto es, entre estados del problema sucesivos) bajo la forma de pares Estímulo-Respuesta; es decir, la búsqueda de un estado u operador en un espacio se representaría por medio de acciones condicionales del tipo:

SI el estado actual es A<sub>1</sub>  
ENTONCES, pasar al estado A<sub>2</sub>

Siguiendo la terminología propuesta por Post (1943) llamaremos a esas acciones condicionales *producciones* o reglas de producción. La ventaja fundamental que facilita este modelo es la de «indexar» el conocimiento, asociando un estado mental a una acción mental (operador). Todo lo que se necesita para resolver un problema ha de estar debidamente «indexado»; es decir, para que la búsqueda heurística tenga éxito, *todos* los datos (tanto los sustantivos al problema, como el procedimiento heurístico propiamente dicho) habrán de estar asociados entre sí para que puedan ser aplicables en un momento dado. La manera más sencilla de indexar esa información es por medio de las características y rasgos descriptivos de las situaciones en las que, heurísticamente, suponemos que una unidad de información específica debe aplicarse. Así, cuando el agente que resuelve un problema (Hombre o Máquina) «identifique» la situación (el estado del problema) gracias a la lista de rasgos descriptivos que posee, aplicará la información y dará un paso hacia la solución.

En otras palabras, lo que se propone es que para resolver heurísticamente un problema usemos un conjunto finito de reglas sencillas, como

si A entonces haz B (1)  
si A' entonces x es B' (2)

Por ejemplo,

SI  
(x) es un asentamiento  
Y (x) tiene (y) en cantidad (h)  
Y (y) es un objeto de cerámica  
Ó (y) es un objeto de vidrio  
Y (y) está datado en el siglo X

ENTONCES  
AVERIGUA LA PROCEDENCIA DE (y)

SI  
OBJETIVO es AVERIGUAR LA PROCEDENCIA DE (y)  
Y (y) está hecho de una materia prima foránea

ENTONCES  
(y) es un Objeto Importado

SI  
(y) es un Objeto Importado  
Y (y) es semejante a la cerámica musulmana del Castillo de Silves  
Y (h) > 50

ENTONCES  
(x) tiene evidencias de Comercio foráneo.

En todas ellas, la parte izquierda (SI A...) es la condición de aplicación, y la parte derecha (ENTONCES...B) la acción o *generador* de la solución o de un estado sucesivo del problema.

¿Cómo funcionan estas reglas? Asumamos que su número es finito y que la situación de partida es el estado inicial del problema (una comunidad humana y un conjunto de objetos que utilizó o utiliza actualmente esa comunidad). Las condiciones de aplicabilidad de las reglas (SI A...) hacen referencia a la descripción de los estados *conocidos* del problema. Si la situación presente coincide con la descripción de una de las situaciones conocidas (esto es, almacenadas en la memoria de trabajo), entonces se activan las unidades de información a las que éstas últimas están asociadas. El resultado es, evidentemente, una nueva situación o estado del problema, una *transformación* del estado inicial.

La función de las reglas es, por tanto, la de añadir nuevos elementos al conocimiento activo (o memoria de trabajo), los cuales se irán utilizando, sucesivamente, para dar forma al espacio del problema. Las reglas en sí no son unidades de conocimiento propiamente dichas, sino un tipo de datos secundario que complementa el conocimiento declarativo. Constituyen una forma de almacenar conocimientos procedurales relativos a situaciones concretas.

Esta descripción de las reglas y de su funcionamiento permite definir de forma más conveniente los *operadores* necesarios en el mecanismo de solución de un problema. La solución a la que queremos llegar puede estar formulada explícitamente en tanto que *hecho declarativo*: la función de una Espada, por ejemplo, es la de servir de signo de identidad social; la función de una vasija es contener un líquido. Usaremos como representación la regla (2). Pero también podemos caracterizar la solución en términos de las acciones necesarias para *construir* nuevos estados (previamente desconocidos) dentro del espacio del problema: la función de un buril es si «estudia las huellas de uso y comprueba si coinciden con las obtenidas al perforar piel en las simulaciones hechas en el laboratorio». Emplearemos para ello las reglas del tipo (1), cuya conclusión es un *procedimiento* para alcanzar la solución (por ejemplo, una serie de condiciones heurísticas para elegir la mejor solución final de entre un conjunto de soluciones posibles), o bien, simplemente, para crearla.

En Inteligencia Artificial suele trabajarse con los siguientes tipos de reglas de producción (cf. Holland et al. 1986, pp.42ss):

#### *Reglas empíricas*

Son las que contienen y asocian las unidades de conocimiento (estados del problema). Se distinguen dos tipos:

**SINCRÓNICAS:** asociaciones entre individuos y categorías en un momento dado.

**Categóricas.**- proporcionan información acerca de las relaciones jerárquicas entre categorías, proporcionando la base para establecer relaciones de pertenencia y asignar propiedades. Por ejemplo:

«Si un objeto es un ánfora, entonces es una cerámica»;

«si un objeto es una vasija bicónica, de forma lenticular, sin cuello ni asas, entonces es una forma 7»;

«si una vasija es una forma 7, entonces su cronología es siglo XVII a.C.».

**Asociativas.**- relacionan conceptos sin establecer relaciones jerárquicas o de pertenencia entre ellos. Estas reglas se limitan a asociar abductivamente categorías o propiedades. Por ejemplo:

«si aparece una cerámica forma 7, entonces aparecerá al lado una alabarda»;

«si aparece una espada en la tumba, entonces esa tumba es masculina».

En esos casos la regla asocia la condición de la regla no a la categoría a la que pertenece, sino a no importa qué otro concepto.

**DIACRÓNICAS:** especifican la manera en la que cambiará el conocimiento a medida que pase el tiempo. Podemos diferenciar las reglas de predicción, que explican al sistema qué es lo que debe esperar del futuro, y las reglas causales, que conducen al sistema a provocar cambios en el entorno. Un ejemplo arqueológico sencillo sería:

«si la Fase Cronológica A se caracteriza por la presencia exclusiva de la Forma Cerámica 7, entonces en la Fase Cronológica sucesiva B esa forma cerámica será más escasa».

La regla anterior es predictiva si se limita a asociar hallazgos en contextos estratigráficos sucesivos, o causal si existe algún fenómeno causal que explique esa asociación temporal.

*Reglas inferenciales*

Pocedimientos abstractos (que no dependen del dominio en el que se ha enunciado el problema) y conciernen a las posibles modificaciones de las distintas reglas empíricas. En otras palabras, son reglas cuya condición y conclusión contienen, a su vez, reglas. Su función principal es, precisamente, la de producir reglas empíricas, por lo que serán, necesariamente, más abstractas que éstas. En general nos referiremos a ellas como *Meta-reglas*. Ejemplo:

**SI** el objetivo es reducir la diferencia entre el estado actual y otro estado,

**ENTONCES**, se tiene que encontrar una operación que lo realice.

**SI** el objetivo es encontrar una operación *O* y no se cumple la condición

**ENTONCES** establecer como objetivo la reducción de la diferencia entre el estado actual y otro que sí cumpla la condición *C*.

**SI** el objetivo es aplicar una operación *O* sobre el estado actual

**ENTONCES** hacerlo

**SI** existe disparidad entre el estado actual y el objetivo al que se pretende llegar,

**ENTONCES** usar el objetivo que establece la necesidad de reducir esa disparidad

**SI** no hay disparidad entre el estado actual y el objetivo al que se pretende llegar,

**ENTONCES** fin del procedimiento.

*Principios operativos*

El mecanismo de control que regula el sistema encargado de resolver el problema. Podríamos referirnos a ellos como reglas cuya condición y conclusión contienen meta-reglas, aunque suelen expresarse en términos distintos a los de las unidades de conocimiento, más cercano al mecanismo de funcionamiento del sistema que al conocimiento que éste contiene. En el caso humano estos principios operativos están expresados por medio de *conexiones neuronales*, y en el caso del ordenador en lenguajes de bajo nivel (expresión en lenguaje de programación y no en forma de regla de producción). Esos principios hacen referencia a la *arquitectura* general del mecanismo general de resolución del problema. Incluyen los procedimientos de activación de reglas, asociación de estados del problema,... En otras palabras, aquellos procedimientos que convierten una lista estática de producciones en un sistema *dinámico*, es decir, en un «programa».

**Introducción a la Programación declarativa**

Un conjunto de reglas **SI...ENTONCES**, debidamente ordenado constituye un programa declarativo. Para que ese programa pueda ser ejecutable -ya sea por la mente humana, ya por un ordenador- es preciso que los símbolos, conceptos, reglas empíricas e inferenciales y los principios operativos estén debidamente ordenados. Dado un conjunto de condiciones de partida y objetivos [*reglas inferenciales*], el programa activa aquellas reglas empíricas que conducirán de la situación inicial a la unidad de conocimiento que satisfaga el objetivo. La solución a un problema, por tanto, es una secuencia de activaciones de reglas.

Estas reglas no deben confundirse con las estructuras de control de los estados del problema, sino que representan a los operadores de *activación* de conceptos. Su función es la de asociar dos tipos de conocimiento: el declarativo propiamente dicho, que describe el estado actual de la memoria activa y el procedural, constituido por los operadores necesarios para alcanzar un nuevo estado (a partir del anterior). Un programa declarativo, por tanto, es diferente a los programas algoritmos usuales, básicamente porque no contiene conjuntos ordenados secuencialmente de mandatos imperativos de ejecución, sino series muy largas y complejas de asociaciones. Así pues, los programas declarativos se distinguen por:

- estar constituidos por enunciados y no por mandatos
- esos enunciados están expresados de manera independiente a su modo de empleo
- los enunciados sucesivos no tienen por qué estar ordenados secuencialmente
- los enunciados no son ni verdaderos ni falsos, sino interpretados y/o verificados por el orden del discurso (esto es, por la forma peculiar en que son utilizados para una tarea concreta, orden que cambia con las tareas a realizar o los problemas a resolver) y en relación con los conocimientos «ciertos» a los que se tiene acceso.

Los elementos constitutivos de un *programa declarativo* o *sistema de producción* son: una Base de Datos, una base de Reglas y un intérprete.

Las **Reglas** -con la estructura típica SI...ENTONCES...- constituyen la representación interna que hace el sistema de unas unidades de conocimiento o información, ya sean éstas factuales o heurísticas. Dentro de la misma Base de Reglas se sitúan ciertas Meta-Reglas, o instrucciones que definen la estrategia de la resolución, arbitran entre reglas conflictivas, definen las reglas que permitirán adquirir nuevas reglas y mantienen los coeficientes de verosimilitud asociados a cada una de las reglas factuales. Mientras que las informaciones «generales», esto es, aquellas que permiten «interpretar» están representadas en forma de Reglas u Operadores, la descripción del problema a resolver -sus estados inicial y final- están representados en la **Base de Datos o de Acontecimientos**, la cual puede tener diferentes estructuras o formatos. Obviamente, la satisfacción de la condición de una regla se establece por referencia a esa Base de Datos, a la vez que la acción estipulada por esas reglas afecta al contenido y a la estructura interna de la misma. Por su parte, el **intérprete** es el responsable de seleccionar y aplicar las reglas.

El sistema opera por ciclos, cada uno de los cuales se descomponen en tres pasos:

- COMPARACIÓN.- El primer paso obliga a encontrar todas las reglas producción aplicables en una situación inicial, comprobando las partes condicionales de dichas reglas.
- RESOLUCIÓN DE CONFLICTOS.- Si en el paso anterior se ha encontrado más de una regla aplicable a una situación en particular, entonces se elige una de ellas. Algunas de las estrategias empleadas habitualmente para resolver esta clase de conflictos de aplicación de reglas son:
  - selección arbitraria
  - cambiar de regla a cada nuevo ciclo
  - elegir la primera regla aplicable que se haya identificado
  - definir a priori un orden de prioridad entre las reglas
  - usar la regla más específica
  - explorar todas las reglas en paralelo.
- ACCIÓN.- el último paso ejecuta la segunda parte de la regla elegida, modificando la base de datos y dando paso a un nuevo ciclo, que operará ya sobre una base de datos modificada; por lo tanto, el nuevo ciclo empezará recalculando las condiciones de las reglas. Lo habitual es que el sistema sea *monótono* y *parcialmente conmutativo*, es decir, que la aplicación de una regla y la modificación que ésta provoca en la base de datos no debe impedir la aplicación posterior de otra regla: dadas dos reglas  $x$  e  $y$ , cuyas condiciones de aplicación son idénticas, la modificación que cualquiera de ellas imponga a la Base de Datos no debiera eliminar ninguno de los hechos en ella existentes que coincidan con los de la parte condicional de la otra regla. Del mismo modo, si un conjunto de reglas permite transformar un estado del problema  $P_1$  en un estado  $P_2$ , cualquier cambio en el orden de prioridad de esas reglas debe seguir transformando  $P_1$  en  $P_2$ . En capítulos sucesivos veremos las ventajas que supone operar con sistemas no-monótonos y/o no conmutativos.

En los «sistemas de producción clásicos», el enfoque está *dirigido por los datos*: el intérprete busca en la base de datos las condiciones estipuladas por la regla, si las encuentra, entonces activa la acción o resuelve el conflicto de hallar más de una regla con esas condiciones. En la mayoría de los casos la acción de la regla modifica la base de datos, de modo que se da opción a otra regla, con unas condiciones distintas, a ser elegida y aplicada. El intérprete continúa hasta que: a) el problema llega a su solución (heurísticamente estipulada por alguna regla, a la que llamaremos regla de detención); b) se alcanza un estado a partir del cual ya no se pueden aplicar nuevas reglas.

Un enfoque alternativo está *dirigido por los objetivos*. El sistema sólo considera las reglas que son relevantes a una solución previamente elegida por el usuario del sistema. El objetivo, por tanto, es establecer la «valididad» o demostración de esa solución a priori. Para ello el intérprete busca aquellas reglas que desde la solución vayan a un estado inicial, fijándose en los consecuentes (acciones) de las reglas y no en sus condiciones: primero la acción que ha activado esa solución, luego la acción anterior y así sucesivamente. Cuando se detecta una regla cuya acción coincide con la solución propuesta al problema, se comparan sus condiciones a las de la base de datos (que define, obviamente, el estado inicial del problema):

- si las condiciones de la regla coinciden con los hechos en la base de datos, la solución que queríamos demostrar será «correcta», con lo que habremos obtenido lo que pretendíamos averiguar
- si las condiciones de la regla son falsas, es decir, la base de datos contiene sus opuestos, la solución

propuesta será incorrecta: habremos demostrado su no idoneidad para resolver el problema cuyo estado inicial está representado en la base de datos

- si las condiciones de la regla no existen en la base de datos (no son ni falsas ni verdaderas), entonces esas condiciones se convierten en «subproblema», intentando demostrarlas buscando otra regla que las contenga en su consecuente.

Resulta evidente que, por sí solas, las reglas no pueden hacer nada. Necesitan de un programa para funcionar (el intérprete), cuyo funcionamiento aparece esquematizado en la figura 2.2.

Las partes procedurales de todo programa declarativo son los *principios operativos* del sistema de resolución, que no tienen por qué estar expresados en el mismo formato que las partes declarativas o los operadores de activación. Esos principios operativos se limitan a aplicar los enunciados declarativos. En principio, es posible extraer del mecanismo de resolución todo lo que hace referencia a su *control*, implementándolo independientemente. De ese modo diferenciaremos explícitamente lo que es *conocimiento* factual, propiamente dicho, de lo que constituye el *mecanismo de razonamiento*. Esta diferenciación nos va a permitir estudiar las relaciones entre uno y otro, pues ninguna de las dos partes puede funcionar sin la otra: Conocimiento (unidades declarativas) y Razonamiento (unidades procedurales o de procedimiento) son dos aspectos de una misma entidad.

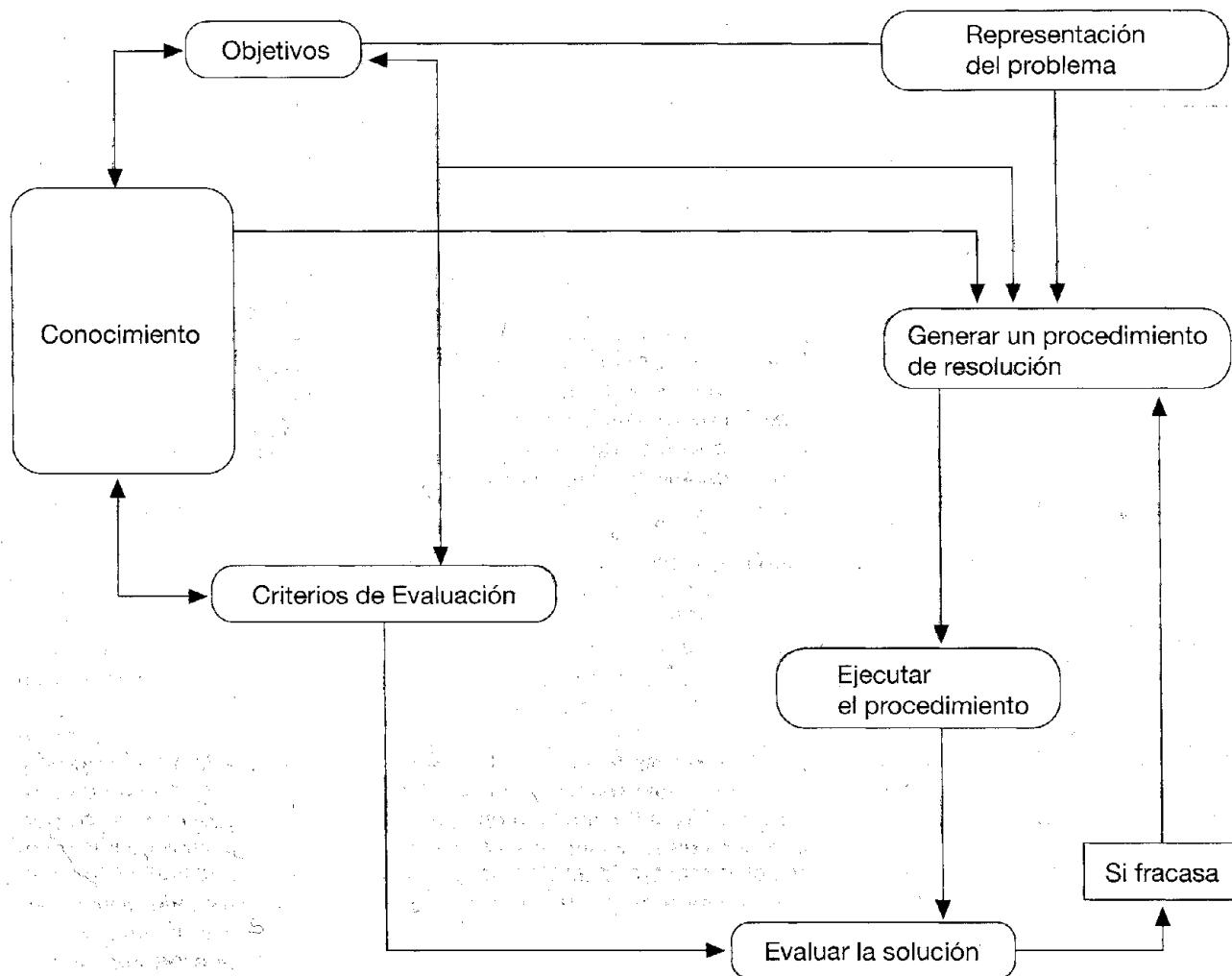


Fig. 2.2. Esquema General del funcionamiento de un Programa capaz de resolver problemas científicos.

Las unidades de procedimiento debieran permitir al arqueólogo usar las unidades declarativas de información que ha introducido en el ordenador. En este sentido, podemos definirlas como aquel tipo de conocimiento que es particular y específico a la tarea concreta que hay que realizar; por tarea debemos entender: «cumplir una serie de objetivos manteniendo ciertas exigencias» (Waern 1989). Distinguiremos entre tres tipos de conocimiento procedural:

- conocimiento acerca de la tarea que hay que realizar,
- conocimiento acerca del sistema como tal,
- conocimiento acerca del sistema como herramienta para realizar la tarea.

El conocimiento procedural es fundamental para poder sacar provecho del conocimiento declarativo. Utilizar el Conocimiento Declarativo sin recurrir al metaconocimiento supondría una mera acumulación no inteligente de hechos sin utilidad alguna. Es por eso por lo que se necesita una auténtica integración de ambos, en la que las unidades de información procedural funcionen como intérpretes. Todo conocimiento declarativo presupone la existencia de algún tipo de conocimiento procedural; por el contrario, el conocimiento procedural puede ser, en principio, totalmente independiente del declarativo -por ejemplo, en los lenguajes clásicos de programación-. En la práctica, sin embargo, debiera establecerse una relación de dependencia entre uno y otro: el conocimiento declarativo sirve también de estructura de control de la acción de los procedimientos de resolución. Es decir, el contenido factual de cada una de las unidades de conocimiento serviría de regla heurística para especificar los movimientos entre estados del problema alternativos.

Esto quiere decir que, en cierto modo, los «usos» del Conocimiento Declarativo derivan del mismo contenido de ese conocimiento. En teoría, por tanto, si añadimos a la expresión de un dato empírico (esto es, a su descripción), una representación de la manera de utilizarlo, obtendríamos «conocimiento». Consideremos que ese «uso» es resolver un problema de índole científica. Para ello, construiremos con ayuda del «dato» un estado del problema. Una manera sencilla de hacerlo sería creando una variable global cuyo contenido fuese la descripción inicial del objeto; a continuación, necesitaríamos el Conocimiento Declarativo de ese objeto, esto es, su significado, y lo «usaríamos» para transformar el estado inicial de esa variable global, avanzando en la resolución del problema.

### **Un enfoque cognitivo alternativo: Redes neuronales**

Las distintas simulaciones de una arquitectura cognitiva esbozadas en la sección anterior no son unánimemente aceptadas por todos los investigadores. En los últimos años se ha insistido especialmente en su poca verosimilitud «biológica», esto es, que el cerebro humano actúa de otra manera. El punto focal de la discusión se sitúa en el tema de la *ejecución en paralelo* de las instrucciones programadas. En las arquitecturas cognitivas anteriores (llamadas «simbólicas») el flujo de información entre la Memoria a Largo Plazo y la Memoria de Trabajo es *secuencial*, esto es, los diferentes operadores contenidos en el segundo término de las reglas de producción se ejecutan unos después de otros. Por lo poco que sabemos acerca de la manera en que el cerebro humano funciona, se supone con bastante fundamento que los diversos operadores son ejecutados simultáneamente en paralelo. Otra crítica bastante difundida es la que afirma que el cerebro no contiene «símbolos declarativos», y que los problemas no se resuelven en términos de operaciones con símbolos. En otras palabras, aunque fuera posible distinguir una Memoria a Largo Plazo y otra Memoria de Trabajo en el cerebro humano, éstas no estarían constituidas por símbolos declarativos, ni el flujo de información de una a otra se basaría en operadores lógicos tales como la Unión y la Intersección.

La investigación reciente en Inteligencia Artificial ha intentado desarrollar nuevas vías. El concepto fundamental desarrollado por estos investigadores recibe el nombre de *memoria asociativa* (Pao 1989, Caudill y Butler 1990). Se trata básicamente de un programa de ordenador que guarda la información por medio de asociaciones o correlaciones con el resto de la información contenida en la memoria. Este sistema informático permite que diferentes *partes* de una entidad estén asociadas entre sí, de modo tal que la activación de cualquiera de esas partes activa las restantes.

El uso de *memorias asociativas* en informática es un resultado de los intentos de simular por medios artificiales algunas de las características del razonamiento humano: cuando un individuo «recuerda» algo, tiende a reconstruir el hecho a partir no sólo del estímulo momentáneo, sino de las relaciones que ese estímulo tenga con todo lo que el individuo ha aprendido previamente (Mayor y Vega 1992, Ruiz-Vargas 1991). Necesitamos, pues, de un método que «recuerde» cual era la forma o la descripción de un objeto, a partir de un estí-

mulo incompleto o ambiguo. El estímulo está asociado al recuerdo, con lo que «recordaremos» el objeto según las asociaciones que pudieran existir entre éste y el estímulo. Si no podemos recordar la forma del objeto, entonces no existe una asociación adecuada. Este proceso de construir asociaciones entre estímulos y conceptos memorizados se denomina *aprendizaje*.

Consideremos el siguiente ejemplo trivial. Soy un arqueólogo que a lo largo de mi experiencia de campo he «visto» y descrito una gran cantidad de cerámicas fenicias (o de cualquier otra cronología). Como resultado de esa experiencia básicamente visual, mi cerebro ha creado una definición peculiar de las cerámicas fenicias. Esos conceptos no han sido aprendidos en tanto que entidades cerradas: en este sector del cerebro no se guardan, por tanto, las imágenes tridimensionales de las ánforas, en este otra, los platos, etc., sino que cada uno de los conceptos existe en tanto que un conjunto de neuronas está «activado». Aquí, *activación* significa, tan sólo, que una célula cerebral envía información (por medio de reacciones físico-químicas) a otras células. Estarán *asociadas*, por tanto, todas aquellas neuronas activadas en un momento dado.

El concepto -ese tipo de cerámica fenicia- no existe en ningún lugar del cerebro, sino que la activación conjunta de un determinado número de neuronas *representa* dicho concepto. Ahora bien, esas neuronas no están especializadas en la representación de un único tipo, sino que muchas neuronas participan en el esquema de *asociatividad* empleado en la representación de diferentes tipos de cerámicas. Ante determinados estímulos visuales -desentierro una nueva cerámica fenicia- mi cerebro «recuerda» de qué cerámica se trata al *activar* las neuronas que estaban asociadas con este estímulo y no con otro. *Todo estímulo idéntico al almacenado produce idéntica respuesta*; incluso estímulos parcialmente semejantes, podrán generar respuestas semejantes, dependiendo la calidad de la respuesta del grado de semejanza entre estímulos. Esta propiedad es importantísima para nuestro propósito, ya que nos permitirá «recordar» un concepto aunque su estimulación sea reducida; en otras palabras, para «recordar» la forma de cierto tipo de cerámica fenicia no es necesario que el estímulo sea idéntico al concepto. Bastará con la activación de un escaso número de neuronas para que se activen simultáneamente todas las neuronas asociadas con aquellas activadas.

Así pues, el rasgo distintivo de estos sistemas es la *asociatividad* existente, no sólo entre estímulo y concepto, es decir, entre observación empírica e interpretación, sino entre todas y cada una de las unidades utilizadas en la representación del concepto. La manera de utilizar una Memoria Asociativa, por tanto, es muy distinta a la manera de consultar un diccionario: el usuario de un diccionario, o una base de datos clásica, selecciona la información que le interesa por medio de mandatos de búsqueda muy concretos, que hacen referencia a la manera en que la información está *indexada*: alfabéticamente, numéricamente, etc. En una Memoria Asociativa, por el contrario, el usuario alcanza la información deseada buscando estímulos que sean «similares» a aquellos que han permitido construir una representación de cierto concepto *en tanto que esquema de asociatividad entre neuronas*. Los conceptos almacenados no tienen una ordenación precisa, sino que están situados en relación a la similaridad que tienen con los demás.

Lo importante es, pues, de qué manera se «guardan» las entidades en la memoria (Representación) y de qué manera puede recuperarse esa información usando información inicial incompleta. (Fig. 2.3).

La memoria asociativa es una amplia estructura de la información en la que el conocimiento está organizado por medio de las relaciones que unen a unos conceptos con otros. Por consiguiente, las memorias asociativas podrán representarse por medio de una red, a base de nudos y de lazos entre nudos (los nudos corresponden a los conceptos y los lazos a las relaciones entre conceptos).

A pesar de sus ventajas, las memorias asociativas tienen un defecto evidente: son muy difíciles de programar. En este libro se expondrá un modo de programar este tipo de sistemas de información, basándonos en la teoría matemática de las *redes neuronales*. Una *red neuronal* o sistema conectivo es un conjunto de unidades de computación sin contenido declarativo alguno, generalmente inspiradas en algún modelo teórico de neurona cerebral. Por sí mismas, estas unidades no pueden resolver ningún tipo de problema, pero si unimos una gran cantidad de unidades en una red por medio de enlaces y uniones, podremos usar ese mismo esquema de conexiones para almacenar información. Así, por ejemplo, la información declarativa

«esta vasija de forma troncocónica exvasada y decoración pintada se puede datar en el siglo XV» adoptaría la siguiente apariencia en un sistema basado en arquitecturas simbólicas (reglas de producción):

Si

[NATURALEZA: VASJA] [FORMA: TRONCOCÓNICA EXVASADA] [DECORACIÓN: PINTADA]

Entonces

DATACIÓN: SIGLO XV

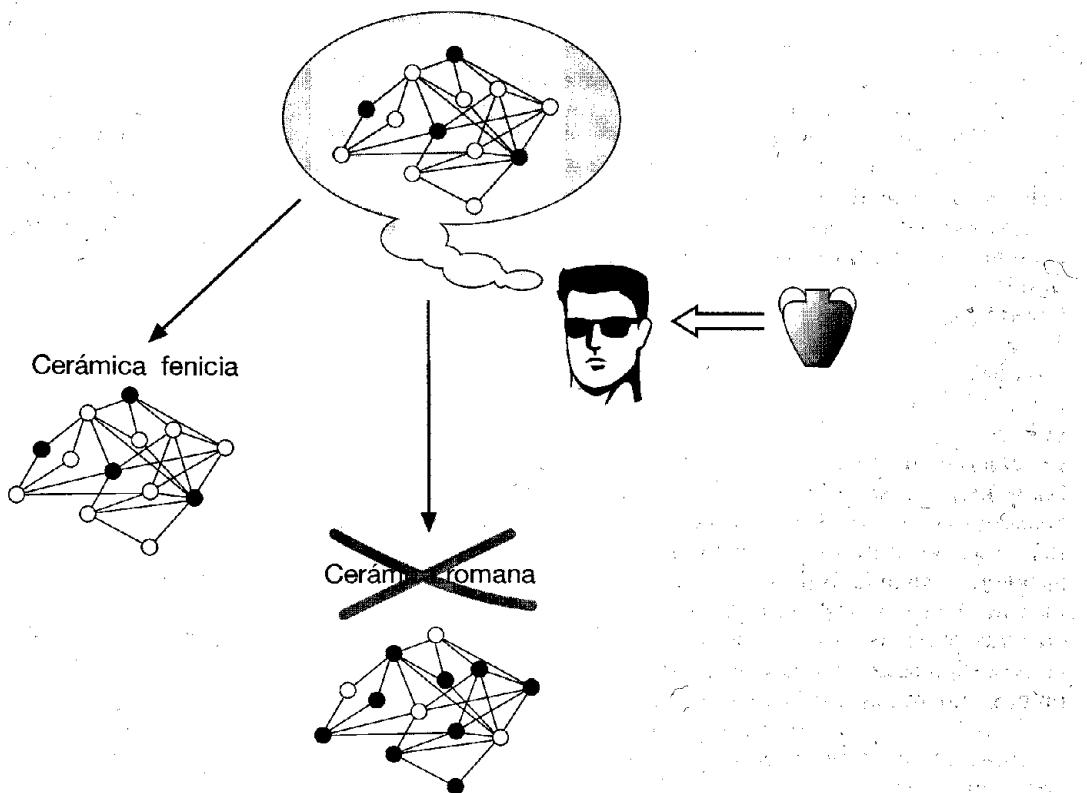


FIG. 2.3. Esquema de funcionamiento de una Memoria Asociativa en el cerebro humano.

En una red neuronal, por el contrario, esa misma información no estaría representada en términos pseudolingüísticos, sino en términos de un subconjunto de conexiones entre unidades de computación, ninguna de las cuales significa nada por sí misma, pero si las unimos de cierto modo (matemáticamente) configuraremos un *vector* que sí que tendrá «significado»: la cerámica del siglo XV. Los rasgos descriptivos de la representación declarativa se han convertido aquí en coeficientes matemáticos:

(0,5678 0,6978 6587,12 5687 0 6587 56978,24)

que miden la *intensidad* de las conexiones entre unidades (Fig. 2.4).

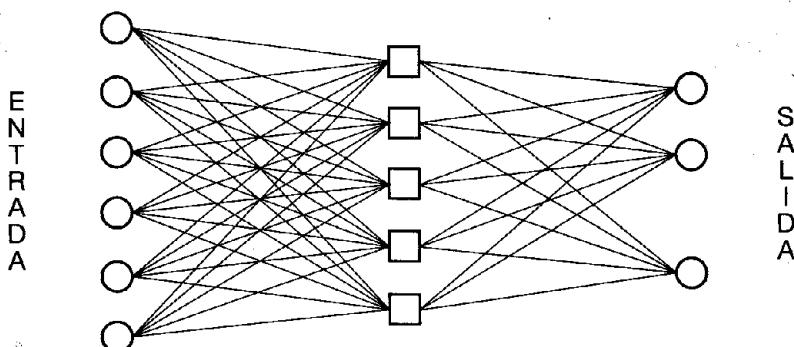


Fig. 2.4. Esquema de una Red Neuronal.

Por lo general se usan dos tipos de redes neuronales: las redes *localizadas*, en las cuales las unidades de computación (*neuronas*) están especializadas en la representación de una información particular, y las redes *distribuidas*, en las que ninguna neurona está especializada, sino que distintas informaciones están representadas por medio de los valores de una función matemática que evalúa la importancia de cada neurona y de cada conexión en la representación de esa unidad de información. En lugar de un programa (en forma de instrucciones o de reglas de producción) que actúe sobre diferentes estructuras de datos, una red neuronal consiste en un gran número de elementos simples de procesamiento (nodos o neuronas) conectados entre sí. Cada conexión está asociada a un valor numérico: el peso o la intensidad que determinan su importancia e influencia en la red neuronal. Ese valor numérico se define como medida de la frecuencia con que la neurona de destino y la neurona de origen han estado activadas simultáneamente. Las neuronas se comunican con el resto de la red transmitiendo una respuesta particular (un número) a todas las unidades con las que está conectada. Ese valor de salida está determinado, a su vez, por el *potencial* o grado de activación de la neurona: una función monótona de los estímulos que recibe de las neuronas vecinas.

Cada neurona es, en realidad, una unidad primaria de cálculo, que opera sobre los estímulos que le llegan a través de las conexiones que la unen al resto de neuronas de la red (Fig. 2.5). No trata esos estímulos uniformemente, sino que los recibe a través de enlaces (conexiones) de importancia desigual. Cada conexión contribuye con un estímulo cuya magnitud iguala al valor de la respuesta que emana de esa neurona; la neurona realiza una suma ponderada de todos esos estímulos para definir su nivel o grado de activación, es decir, para saber si resulta activada o no por el conjunto específico de estímulos recibido: si la *cantidad de estímulo* alcanza un nivel determinado, la neurona es capaz de generar una respuesta a ese estímulo. Se denomina *función de transferencia* esa correspondencia o función que asocia un estímulo dado a un nivel particular de activación de la neurona. Una forma de calcular esa función sería multiplicando cada estímulo por la medida de la importancia de la conexión a través de la cual llega, sumando algebraicamente, a continuación, todos los estímulos que llegan a una neurona.

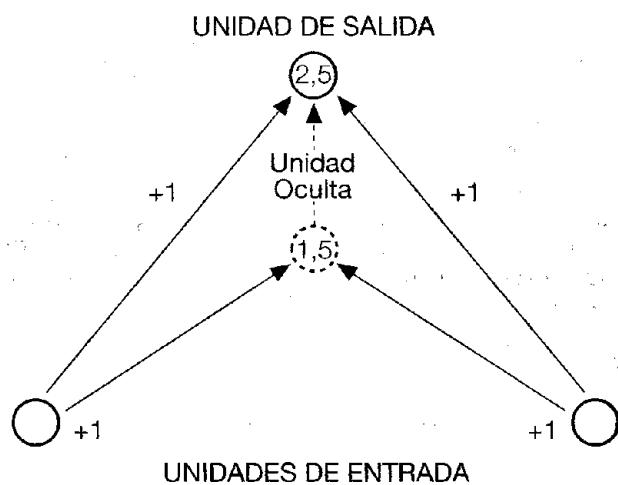


Fig. 2.5. Simulación informática del comportamiento de una Neurona.

En la mayoría de aplicaciones usuales, unos cuantos de esos nodos o neuronas son los elementos de entrada a la red, cuya activación está fijada por el usuario (estímulo externo). Otro conjunto de elementos está formado por las neuronas de salida o resultados del problema. Entre ambos se sitúa un tercer conjunto, que suelen recibir el nombre de *neuronas ocultas*. Dado un estímulo específico, los elementos de entrada han de ser capaces de obtener una respuesta, representada por una configuración determinada de las neuronas de salida. La tarea de la red es calcular los valores correctos de activación de las neuronas ocultas que pongan en correspondencia los estímulos con las respuestas. En primer lugar, de las neuronas de entrada a las neuronas ocultas, el estímulo externo es traducido en términos de la representación interna que es específica y particu-

lar a esa red. Al pasar de las neuronas ocultas a las neuronas de salida, esa representación es traducida de nuevo en términos del resultado esperado. Por lo tanto, el esquema peculiar de conexiones y funciones de transferencias de las neuronas ocultas constituye un código interno que la red va a usar para guardar las instrucciones que permitirán asociar en el futuro ese estímulo con esa respuesta.

Volvamos al tema que más nos interesa, la descripción de la arquitectura del sistema encargado de resolver automáticamente un problema. Es preciso aceptar que en ocasiones resulta más conveniente representar el espacio del problema por medio de una red neuronal que por medio de un programa declarativo: cada neurona en la red constituiría la representación de un estado del problema; en un momento dado, el grado de activación de esa neurona podría considerarse como el grado en el que la red «cree» que ese estado es necesario para trazar el camino entre el estado inicial y el final. Del mismo modo, el estado inicial del problema coincide con los elementos de entrada (estímulo), el estado final con el elemento de salida (o respuesta proporcionada por la red), y los estados transitorios o enlaces entre uno y otro se representarían por medio de las «neuronas ocultas». Los operadores necesarios para pasar de un estado a otro ya no serían operadores lógicos basados en la manipulación de símbolos, sino funciones matemáticas que representarían los pesos diferenciados de las interconexiones entre las neuronas implicadas. El conjunto de valores numéricos que representan ese esquema de activación no ha sido programado, sino que la red lo aprende generalizando diversos ejemplos (estímulos). Ese aprendizaje se lleva a cabo variando las conexiones entre las neuronas de acuerdo con una ley de aprendizaje específica, hasta que el estímulo presentado provoque en la red la respuesta pretendida. En capítulos sucesivos se desarrollarán y explicarán más detenidamente las técnicas de aprendizaje neuronal.

¿Qué diferencias hay entre un programa declarativo y una red neuronal? En el primer caso se considera que el nivel de representación simbólica (las reglas) constituye el nivel de descripción apropiado. No se plantea de dónde viene la información, sino tan sólo cómo se manipula: un sistema de producciones puede describirse en tanto que tal sin considerar ni su implementación (su estructura operativa) ni su origen (procedencia de los estados del problema), porque el conocimiento está explícitamente diferenciado de sus estructuras de control; la arquitectura del sistema -las reglas que permiten manipular las reglas- aparece tan sólo como un medio y no como conocimiento en sí. Por el contrario, en una red neuronal la arquitectura del sistema, esto es, la implementación del conocimiento afecta y define la acción misma del sistema, pues son indisolubles. No se necesitan estructuras de control, porque los datos y su tratamiento no están diferenciados. Las unidades de información no se representan mediante expresiones declarativas, sino por la estructura y estado de activación de la red. El contexto del problema no es otro que la red entera.

Lo comprenderemos mejor si recordamos que una red neuronal no es otra cosa que la *simulación* del mecanismo de funcionamiento del cerebro humano: la función de transferencia propia de cada neurona y la propagación de sus valores de neurona a neurona a través de conexiones de importancia desigual es el motivo por el que la respuesta de una neurona no sólo depende del estímulo externo, sino de la respuesta de las otras neuronas a las que está conectada. Los conjuntos de neuronas que constituyen una red son, por tanto, conglomerados interconectados en los que la actividad persiste una vez pasada la estimulación inicial. Debido, precisamente a esa persistencia de la actividad otras neuronas pueden activarse independientemente del estímulo, emitiendo respuestas que a su vez afectarán a la neurona activada en primer lugar, inhibiéndola en algunos casos, con lo que el estado de activación final de la red será una representación semi-autónoma de la realidad, no dependiente exclusivamente de la información empírica externa.

### Comentarios bibliográficos

Los obras de referencia idóneas a las que puede acudir el lector para ampliar las ideas aquí expresadas son el *Handbook of Artificial Intelligence* (Barr y Feigenbaum 1981-1992, 4 vols.). La *Encyclopedia of Artificial Intelligence* (Shapiro 1992, 2 vols.) y los manuales de introducción a la Inteligencia Artificial escritos por Winston (1984), Haugeland (1989), Sharples et al. (1989), Boy (1990), Partridge (1990), Nagao (1990), Ginsberg (1993), Russell y Norvig (1994), Luger (1994).

Respecto al tema de la Resolución de Problemas en la Psicología Cognitiva: Bourne et al. (1979), Anderson (1983), Holland et al. (1986), Stillings et al. (1987), Johnson-Laird (1988), Sternberg y Smith (1988), Kaplan et al. (1989), Waern (1989), Newell (1990).

Sobre la modelización de los mecanismos de resolución de un problema en términos de un sistema de producciones son fundamentales los trabajos de Newell y Simon (Newell y Simon 1972, 1976; Newell 1973, 1980,

Simon 1973, 1979, 1983), con abundantes referencias acerca de los Sistemas de Producciones, si bien para ese tema específico es mejor referirse a los libros generales citados anteriormente o a la bibliografía sobre Sistemas Expertos (cf. Capítulo 3). Con respecto a las Redes Neuronales y Arquitecturas Conectivas, incluso los libros de introducción suelen ser difíciles de seguir. Así y todo se pueden recomendar los libros de Aleksander (1990), Aleksander y Morton (1990), Caudill y Butler (1990), Dayhoff (1990), Eberhart y Dobbins (1990), Khanna (1990).

### 3. INTERPRETACIÓN AUTOMÁTICA

Imagino que, tras la lectura del capítulo anterior, el lector tiene más o menos clara la tarea de introducir («representar») conocimiento en el ordenador, y de las dificultades que ha de solventar para llevar a buen término esa acción. De lo que se trata ahora es de usar el conocimiento que hemos representado para resolver problemas arqueológicos. En este capítulo abordaremos el tipo más simple de resolución: el razonamiento deductivo, que consiste tan sólo en la asociación simple de conceptos interpretativos (o entidades teóricas) con datos empíricos. Como veremos a continuación, el interés de un sistema informático aparentemente tan sencillo estriba en sus aplicaciones prácticas:

- se puede usar el programa cuando no disponemos de un experto en ese tema (por ejemplo, datación de un tipo de cerámica característico)
- sirve de programa de gestión de la información, cuando la cantidad de conocimientos existente en ese dominio es demasiado grande.

#### Un ejemplo sencillo

Supongamos que nos interesa averiguar la cronología de un conjunto de Estelas Decoradas con motivos de «guerrero», halladas en el Sudeste de la Península Ibérica (sobre todo en Extremadura y Andalucía Occidental, cf. Almagro Basch 1966, Barceló 1989). Una vez puesto en marcha el ordenador, cargado el disco que contiene el programa (ESTELAS v. 1.0), y abierto éste último, la pantalla muestra el texto de la siguiente pregunta:

¿APARECE LA REPRESENTACIÓN DE UN ANTROPOMORFO?

Supongamos que, en efecto, en la Estela que estamos estudiando aparece la representación de un Antropomorfo. Responderemos por tanto «SI» a la pregunta anterior. El programa responde acto seguido con una nueva pregunta:

¿APARECE LA REPRESENTACIÓN DE UNA ESPADA?

Consideraremos que esa Estela hipotética también cuenta con la representación de una Espada. A continuación el programa nos interroga acerca de la morfología de esa espada: responderemos «NO» a la morfología «ESTRANGULADA» y «SI» a la morfología de «BORDES PARALELOS». El programa proporciona una primera evaluación cronológica:

ESPADA POSIBLEMENTE DE LENGUA DE CARPA

y pide más información para evaluar la cronología de la Estela. Al cabo de unos segundos (tiempo necesario para que el ordenador abra un nuevo subprograma), se nos pregunta por la existencia de la representación de una Fíbula de Codo. Tras nuestra respuesta afirmativa, el Sistema pregunta si aparece un Casco de Cresta. Una respuesta afirmativa a esta pregunta nos lleva automáticamente a la solución:

ESTELA SEMEJANTE A LA DE CABEZA DE BUEY I (inicios siglo IX a.C)

Estudiemos ahora una Estela conocida, la de Baraçal, por ejemplo (Curado 1984). Volvemos a abrir el programa principal ESTELAS y reiniciamos la sesión. Respondemos «NO» a la primera pregunta («¿APARECE LA REPRESENTACIÓN DE UN ANTROPOMORFO?») y «SÍ» a la segunda («PRESENCIA DE ESPADA»). Seguidamente, estimamos que la hoja de la espada representada tiene forma «ESTRANGULADA». La respuesta provisional del programa es considerar esa espada como «PISTILIFORME», solicitando más datos. Al cabo de unos segundos, en la pantalla del ordenador aparece una nueva pregunta que inquierte por la localización geográfica de la Estela («¿HA APARECIDO LA ESTELA EN EL VALLE DEL TAJO O EN LA ALTA EXTREMADURA?»). Tras una contestación afirmativa, el programa pregunta por la existencia de una Fíbula de Codo, de un casco de Cresta y de un Casco de Cuernos. Si respondemos «NO» a todas ellas obtendremos la siguiente contestación:

#### ESTELA SEMEJANTE A LAS DE BARAÇAL Y FOIOS (ca. 1000 -950 a.C.)

Consideremos ahora el caso de Estelas de interpretación difícil, como pudiera ser el caso de las de Belalcázar o Capilla I. Un resumen del desarrollo de la sesión sería el siguiente:

ANTROPOMORFO?	SÍ
ESPADA?	NO
ESCUDO EN V, LANZA, Y/O CASCO DE CRESTA?	NO
CASCO DE CUERNOS Y/O ARCO-FLECHA?	NO
ARMAS?	NO
JOYAS, ADORNOS Y/O DIADEMA?	SÍ

La respuesta sería la siguiente:

#### ESTELA SEMEJANTE A LA DE TORREJON DEL RUBIO II Y DEMÁS ESTELAS FEMENINAS (probablemente tardía, siglos VIII - VII a.C.)

Sea ahora el caso de una Estela sin elementos cronológicos conocidos:

ANTROPOMORFO?	SÍ
ESPADA?	SÍ
ESPADA DE HOJA ESTRANGULADA?	NO
ESPADA DE BORDES PARALELOS?	NO
ESPADA DE HOJA HUSIFORME?	NO
ESPADA EN FORMA DE SIMPLE TRAZO LINEAL?	SÍ

La primera respuesta que proporciona el programa es : «ESTELA PROBABLEMENTE TARDÍA». Al cabo de unos segundos continúa la sesión:

FÍBULA DE CODO?	NO
CASCO DE CRESTA?	NO
Casco de cuernos?	NO

A lo que responde el programa «FALTAN ATRIBUTOS CRONOLÓGICAMENTE FIABLES». A continuación solicita más datos:

REPRESENTACIÓN DETALLES ANATÓMICOS?	NO
APARECEN EN TOTAL MENOS DE CINCO ELEMENTOS REPRESENTADOS?	SÍ

La solución es la siguiente:

#### ESTELA SEMEJANTE A LA DE SETEFILLA (siglos VIII-VII a.C.)

El programa es lo suficientemente sofisticado como para poder analizar una Estela para la cual no existe prototipo, es decir, cuya iconografía no coincida con el estado actual de los conocimientos sobre el tema. Así, por ejemplo, si la sesión de consulta se desarrollase de la siguiente forma:

ANTROPOMORFO?	NO
ESPADA?	SÍ
HOJA ESTRANGULADA	NO
ESPADA DE BORDES PARALELOS?	SÍ

Respuesta Provisional: ESPADA PROBABLEMENTE DE LENGUA DE CARPA

FÍBULA DE CODO?	NO
CASCO DE CUERNOS?	SÍ

Como respuesta aparecería el siguiente texto:

**ESTELA ÚNICA:** La presencia de una representación de casco de cuernos sorprende en este contexto, pues las Estelas con Espada de lengua de carpa y sin antropomorfo son consideradas como «antiguas» frente a las que cuentan con antropomorfo, casco de cuernos y espada de lengua de carpa. Ahora bien, no es imposible una tal asociación debido a la incertidumbre cronológica suscitada por el casco de cuernos, cuyo origen cultural no está claro). Cronología: posiblemente algo más reciente que las Estelas con casco de cresta, en un momento avanzado del siglo IX a.C. o incluso en el siglo VIII a.C.

## Introducción a los Sistemas Expertos

El anterior es un ejemplo típico de Sistema Experto. Podemos considerar que un Sistema Experto es un programa informático que utiliza «automáticamente» los conocimientos propios a un experto humano en un ámbito específico, de forma que el sistema resultante ofrezca CONSEJOS INTELIGENTES o bien sea capaz de adoptar una DECISIÓN INTELIGENTE en ese dominio. Otra característica adicional deseable, que para algunos es fundamental, es la capacidad del sistema para JUSTIFICAR SU PROPIA LÍNEA DE RAZONAMIENTO de manera comprensible al usuario. El enfoque adoptado para conseguirlo es la PROGRAMACIÓN BASADA EN REGLAS (o programación *declarativa*, cf. Capítulo 2).

El Sistema Experto presentado al principio de este capítulo (ESTELAS v. 1.0) es un ejemplo de programa capaz de tomar ciertas decisiones (atribución cronológica) en un ámbito muy específico de la Arqueología: las Estelas Decoradas del Bronce Final en el Sudoeste de la Península Ibérica. Partiendo de unos rasgos descriptivos (identificación de las representaciones grabadas en las Estelas) el programa produce unos resultados aparentemente «inteligentes», esto es, diagnostica la cronología de cada Estela. El programa está constituido por un conjunto de cuarenta factores de conocimiento, de los cuales derivan las interpretaciones (la asignación cronológica de las Estelas). Así por ejemplo, el resultado «ESTELA SEMEJANTE A LA DE BARAÇAL Y A LA DE FOIOS» deriva del factor:

VALLE DEL TAJO O ALTA ANDALUCÍA?	SÍ
FÍBULA DE CODO?	NO
CASCO DE CRESTA?	NO
CASCO DE CUERNOS?	NO

que está expresado, de forma no ambigua, mediante una regla del tipo SI... ENTONCES... :

ESTELA SEMEJANTE A LA DE BARAÇAL Y A LA DE FOIOS, SI...

VALLE DEL TAJO O ALTA ANDALUCÍA	= SÍ
FÍBULA DE CODO	= NO
CASCO DE CRESTA	= NO
CASCO DE CUERNOS	= NO

¿Cómo puede un programa de ordenador reproducir el razonamiento de un experto humano? Ya hemos abordado esta cuestión en capítulos anteriores: representando el conocimiento científico en un formato «computable». Veíamos que eso era posible diferenciando el Conocimiento Declarativo del Procedural. Ahora bien, por separado, ninguno de esos componentes puede funcionar ni resolver un problema: sin instrucciones acerca de su uso el Conocimiento Declarativo no es, propiamente hablando «Conocimiento»; el Conocimiento Procedural, por su parte, será incapaz de proponer soluciones a un problema sin información relevante acerca de dicho problema. En definitiva, para que un Sistema Experto reproduzca efectivamente el razonamiento de un experto humano (un arqueólogo, en nuestro caso), es necesario que combine el Conocimiento Declarativo y el Procedural. El primero se incluye en una Base de Hechos, o Base de Datos que contiene observaciones empíricas y resultados experimentales (datos procedentes de una excavación arqueológica, por ejemplo). El Conocimiento Procedural suele estar contenido en una base de Reglas, y representa la Teoría Científica que queremos usar para «interpretar» los datos.

Precisamos además de un programa que se encargue de traducir las instrucciones contenidas en el componente procedural en instrucciones computacionales; en otras palabras, un programa (o bien, en un futuro cercano, un circuito impreso) que ponga en marcha los mecanismos generales de combinación de los conocimientos declarativos y procedurales, buscando las relaciones lógicas entre la base de datos y la base de Reglas y ejecutando los operadores existentes en la parte conclusión de las reglas. Ese programa recibe el nombre de *Motor de Inferencias* (Fig. 3.1.).

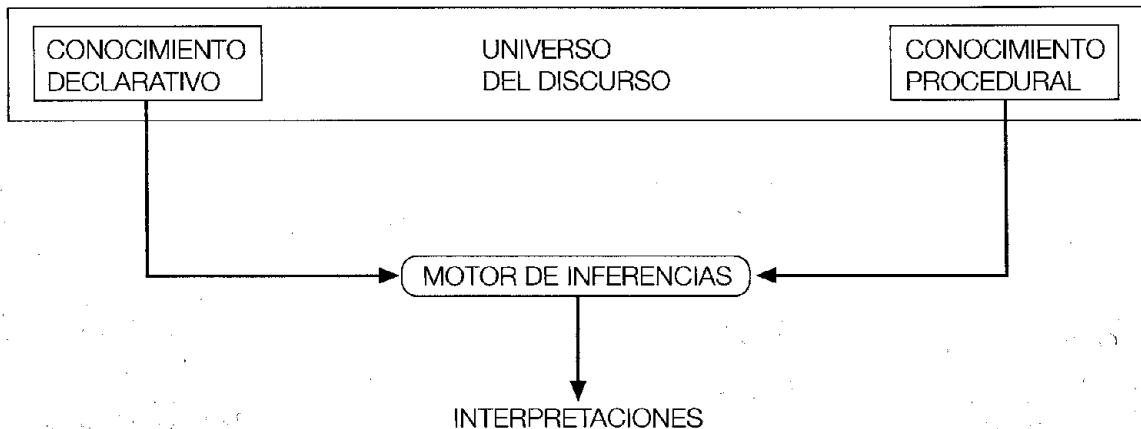


Fig. 3.1. Esquema general de un Sistema Experto.

Es importante insistir en el carácter *general* (esto es, no basado en un conocimiento científico específico) de este Motor de Inferencias, carácter que lo diferencia sensiblemente del contenido «práctico» del Sistema Experto. Se trata de un conjunto estructurado de instrucciones (y no de un programa declarativo) cuya misión es *utilizar* las reglas que, a su vez, representan el conocimiento procedural. El Motor de Inferencias es, pues, un programa que «activa» las reglas, si las condiciones de aplicación contenidas en el antecedente de las mismas están relacionadas con las unidades de conocimiento declarativo contenidas en la Base de Datos o Hechos, y que ejecuta las operaciones que aparecen en el consecuente de las mismas. Esas instrucciones de «activación» y de «ejecución» no dependen del contenido específico de cada regla, sino que siempre son las mismas sea cual sea el conocimiento procedural empleado.

### Motores de Inferencia

Hemos visto que un Sistema Experto se compone, esencialmente de tres módulos: una Base de Hechos, una Base de Reglas y un Motor de Inferencias, esto es, un conjunto ordenado de conocimiento Declarativo y Procedural (específico a un problema concreto), y un programa capaz de *usar* esos dos tipos de conocimien-

to para encontrar la solución a un problema definido en ese dominio concreto; por *uso* del conocimiento, entenderemos aquí la combinación del conocimiento Declarativo y el Procedural para producir nuevo Conocimiento Declarativo.

Analicemos ahora el *motor de inferencias*. Su función es la de *ejecutar* el Conocimiento, combinando Hechos con Reglas. Para ello, el motor de inferencias actúa de la siguiente manera:

1. DETERMINA LAS REGLAS SUSCEPTIBLES DE SER ACTIVADAS
2. ELIGE UNA REGLA A EVALUAR DE ENTRE LAS REGLAS ACTIVABLES
3. EJECUTA DICHA REGLA Y ACTUALIZA LA BASE DE HECHOS

Para determinar la aplicabilidad de las reglas, el motor de inferencias compara su enunciado con las cláusulas (scan del tipo que sean: encuadres, expresiones relacionales, etc.) existentes en la base de Hechos. Obsérvese que la base de hechos es *siempre* particular a un caso, y que la base de Reglas es particular a un problema o dominio. Lo que establece el motor de inferencias es si el caso representado por la base de datos es relevante a la base de Reglas; en otras palabras, si ambos mencionan las mismas unidades. No olvidemos que una Regla no es más que una operación lógica realizada con dos expresiones *formuladas en la misma notación que las unidades existentes en la base de hechos*. En definitiva, una Regla de Producción se limita a poner en relación distintas unidades de Conocimiento Declarativo.

El programa realiza esa comparación por medio de una operación denominada *unificación*. En general, dos expresiones pueden unificarse si son idénticas o si pueden llegar a ser idénticas una vez sustituidas sus variables. Por ejemplo, podemos unificar las expresiones

«001 **es** Fíbula de Codo»  
«X **es** Fíbula de Codo»

Sustituyendo 'X' (una variable) por el individuo «001». Para validar cualquiera de los componentes de la regla (tanto el antecedente como el consecuente) en función de lo que contiene la base de hechos, el Motor de Inferencia busca, en primer lugar si las expresiones tienen:

- un predicado (verbo) con el mismo nombre [la partícula subrayada]
- la misma cantidad de argumentos
- las mismas constantes

En el caso contrario, la regla no será activable a partir de esa base de hechos concreta.

Una vez activadas, esto es, seleccionadas con el fin de resolver un problema, las reglas han de ser ejecutadas por el Motor de Inferencias. Este dispone de dos modos distintos de ejecución:

#### A. Encadenamiento hacia adelante («Forward chaining»)

Activa aquella regla o unidad de conocimiento procedural cuya condición existe en la Base de Conocimiento Declarativo, es decir, es «idéntica» a una de las expresiones que allí existen. El programa deduce, entonces, la consecuencia de dicha regla. En otras palabras, el motor de inferencias evalúa la validez de la premisa o antecedente de la regla a partir del Conocimiento Declarativo: si esa premisa *está definida* entonces el conocimiento asociado a la misma (el consecuente de la regla) es agregado al Conocimiento Declarativo. Por ejemplo (libremente adaptado de Guillaume 1987):

##### *REGLA 1*

- SI            a) los objetos en metal encontrados en Penya Negra (Alicante, Bronce Final) son objetos **de uso corriente**  
               b) se han encontrado en Penya Negra útiles de obreros metalúrgicos, lingotes de metal, crisoles, moldes de fundición,  
 ENTONCES Los objetos en metal de Penya Negra eran fabricados *in situ*.

El motor de inferencias buscará en el módulo que contiene el Conocimiento Declarativo (la base de Hechos) si existen o no las proposiciones a) y b); en caso afirmativo, añadirá el predicado: «Los objetos en metal

de Penya Negra eran fabricados *in situ*.» Este hecho deducido por el Sistema será utilizado, a su vez, para continuar con las inferencias:

**REGLA 2**

SI

- a) los objetos en metal encontrados en Penya Negra son **muy numerosos**
- b) los objetos en metal de Penya Negra eran fabricados *in situ*.

ENTONCES

Existía un artesanado metalúrgico **muy desarrollado** en el Sudeste de la Península Ibérica en la época del Bronce Final e Inicios de la Edad del Hierro.

Fijémonos que la premisa b) en la Regla 2 no existía previamente en la base de Hechos, sino que ha sido «inferida» gracias a la activación de una regla anterior.

El mecanismo general de este procedimiento es muy sencillo. Consideremos una Base de Hechos que contiene dos unidades de Conocimiento Declarativo (A y B), y las siguientes reglas que representan el conocimiento procedural que necesitamos:

- 1) Si A y C, entonces E
- 2) Si F y C, entonces H
- 3) Si B y E, entonces H
- 4) Si B entonces C
- 5) Si H entonces X

Nuestro objetivo es saber si es posible alcanzar X, partiendo de A y B; en otras palabras, evaluar la verosimilitud de la Hipótesis «A y B son X». El mecanismo de inferencia por encadenamiento hacia adelante actúa del siguiente modo:

**1.-** Sabemos que A y B son verdaderos (es decir, existen en la base de Hechos, por lo que forman parte del Conocimiento Declarativo).

**2.** Si lo que hemos de demostrar es X, entonces X debe figurar en la conclusión de alguna Regla (es decir, debe existir alguna unidad de conocimiento procedural que permita «producir» X).

**3.** Generamos nuevos hechos, mediante el uso de aquellas unidades de conocimiento procedural que contienen A ó B como premisas

**4.** Continuamos añadiendo nuevos hechos a la Base de Hechos combinando repetidamente la progresivamente creciente Base de Hechos a la base de Reglas, hasta que X entre a formar parte de la base de Hechos, o hasta que la Base de Hechos alcance su mayor tamaño (es decir, que no se le puedan agregar nuevos hechos).

Obsérvese cómo:

- la regla 4) produce el hecho C, con lo que se modifica el conocimiento declarativo, que ahora contiene las Unidades A, B y C.
- la regla 2) genera al hecho F, con lo que se modifica el conocimiento declarativo, que ahora contiene las Unidades A, B , C y E.
- la regla 3) genera el hecho H, con lo que se modifica el conocimiento declarativo, que ahora contiene las Unidades A, B , C, E y H.
- la regla 5) asigna X al conocimiento declarativo, con lo que se resuelve el problema.

**B.- Encadenamiento hacia atrás (backward chaining ó «backchaining»)**

El Sistema no usa las proposiciones que permiten activar ciertas reglas, existentes en la base de datos, sino que empieza a partir de una proposición (Hipótesis) formulada por el usuario. A continuación va buscando paulatinamente en la base de Reglas si existe alguna que contenga en su conclusión esa proposición u objetivo. El mecanismo de funcionamiento es el siguiente:

1. Supongamos que X es la proposición que queremos demostrar, al igual que en el ejemplo anterior
2. Existe una regla que contiene la proposición X en su consecuente (Regla 5)
3. La premisa de esa Regla se convierte ahora en nuevo objetivo a demostrar, pues H no existe en la Base de datos (A, B). Por lo tanto, se busca una regla que contenga H en su consecuente.
4. Existe una regla que contiene la proposición H en su consecuente (Regla 3)

5. La premisa de esa Regla se convierte ahora en nuevo objetivo a demostrar, pues E no existe en la Base de datos (A, B). Por lo tanto, se busca una regla que contenga E en su consecuente.
6. Existe una regla que contiene la proposición E en su consecuente (Regla 1)
7. La premisa de esa Regla se convierte ahora en nuevo objetivo a demostrar, pues C no existe en la Base de datos (A, B). Por lo tanto, se busca una regla que contenga C en su consecuente.
8. Existe una regla que contiene la proposición C en su consecuente (Regla 4)
9. La premisa B de esa regla existe en la base de datos, por lo que X queda demostrada: existe una conexión lógica continua entre X y (A,B), que permite afirmar «A y B son X».

La Hipótesis que deseamos demostrar es, por ejemplo, «Este útil lítico ha sido importado». El Sistema Experto iniciará la búsqueda, entre las reglas que han sido implementadas en él, de aquella que pueda confirmar la Hipótesis, por ejemplo:

- |    |   |
|----|---|
| SI | a) el útil lítico está fabricado en sílex                 |
|    | b) el origen de ese tipo de Sílex se encuentra en Francia |
|    | c) el útil ha sido encontrado en Polonia                  |
- ENTONCES Este útil lítico ha sido importado

Para que el motor de inferencias sea capaz de aplicar esta regla es preciso que el Sistema Experto disponga de información acerca de la materia prima del útil lítico en cuestión y acerca del lugar del hallazgo. Si esta información no existe en la Base de Conocimientos (es decir, ya en la Base de Datos, ya en la Base de Reglas como consecuente de otra regla) siempre será posible programar el Sistema para que pida al usuario que introduzca información sobre ello (cf. Voorrips 1990).

#### C.- Diseños avanzados de Motores de Inferencia

El principal problema con los motores de inferencia que trabajan por encadenamiento hacia adelante es el tiempo que requieren para encontrar y aplicar las reglas que corresponden a específicas instancias de los datos. Cuando el Sistema Experto es grande (de un orden superior a 2000 reglas), esa operación es prácticamente intratable por el ordenador. Para resolver esa dificultad existen diversas alternativas, en especial el algoritmo RETE y los motores de inferencia basados en la teoría de las probabilidades. Aunque no excesivamente difíciles, el estudio de estos mecanismos excede los límites de este libro. El lector interesado puede encontrar suficiente material en las referencias bibliográficas.

#### Reglas de Producción y coeficientes de verosimilitud

Hasta ahora hemos utilizado reglas «exactas»; esto es, el programa ha asignado a cada una de las proposiciones o expresiones de conocimiento declarativo uno (y sólo uno) de los siguientes valores: *verdadero*, *falso*. Sin embargo, prácticamente nunca los arqueólogos están seguros de sus interpretaciones; por consiguiente, en la mayoría de las ocasiones habremos de permitir al sistema experto proponer soluciones *inexactas*. La *inexactitud* o grado de confianza en una solución se suele expresar numéricamente mediante una cifra de 0 a 100:

- 100, significa «verdadero»
- 0, significa «falso».

Mejor dicho, «100» podría traducirse por *verdadero definitivamente*, en tanto que un valor de 80, por ejemplo, indicaría que el sistema está seguro tan sólo en un 80 % de su validez.

Estos valores numéricos suelen recibir el apelativo de *factores de Certidumbre* (FC), para diferenciarlos de los «valores de verdad» de la Lógica de Predicados, los cuales tienen una connotación formal (y, en consecuencia, universal) totalmente ajena a la programación de Sistemas Expertos (que por definición son particulares y heurísticos). La función de estos índices es distinta según el tipo de unidad de conocimiento al que estén asociados.

En el caso de la Base de Hechos (conjunto de datos empíricos), el factor de certidumbre suele señalar la confianza que tenemos en un dato empírico. Por ejemplo, como resultado de una prospección en dos yacimientos,

A y B, estamos bastante seguros que en A aparece cerámica tipo 1, mientras que tenemos nuestras dudas acerca de la posible presencia del tipo 2; la presencia del tipo 2 en el yacimiento B es prácticamente segura y la datación de esas cerámicas, por su parte, parece bastante fiable. La expresión de esa «confianza» o seguridad por medio de Factores de Certidumbre (FC) aparece en la tabla:

YACIMIENTO A	<i>bay</i>	CERÁMICA TIPO 1	FC=75
YACIMIENTO A	<i>bay</i>	CERÁMICA TIPO 2	FC=50
YACIMIENTO B	<i>bay</i>	CERÁMICA TIPO 2	FC=90
CERÁMICA TIPO 2	<i>datación</i>	SIGLO VIII	FC=90
CERÁMICA TIPO 1	<i>datación</i>	SIGLO VII	FC=80

¿Qué significa, en términos arqueológicos «estar seguros al 75 ó al 50 % de un hecho? Se trata del grado de «verosimilitud» de ese hecho. En Teoría de la ciencia se ha utilizado bastante este modo de evaluar la verdad de los enunciados científicos, asimilándolo, por lo general, a una *medida de la probabilidad de error*. En otras palabras, la expresión

CERÁMICA TIPO 2      *datación*      SIGLO VIII      FC=90

debe entenderse en los siguientes términos: «la probabilidad que la Cerámica del Tipo 2 date en el Siglo VIII a.C. es del 90%».

Dos son las principales ventajas de esa «lectura probabilística» de los factores de certidumbre. Por un lado, el mecanismo de activación de hechos y reglas en un sistema experto se aproxima a la inferencia estadística clásica (factores de certidumbre = intervalos de confianza); por otro lado, permite utilizar los axiomas de la Teoría de Probabilidades para desarrollar el motor de inferencias del programa, de modo que lo que en páginas anteriores hemos denominado *espacio del problema* pasa a convertirse en un experimento probabilístico (cf. Capítulo 3).

Ahora bien, el uso de probabilidades en los mecanismos de resolución automática de problemas plantea una dificultad mayor: *la asignación de probabilidades iniciales*. Supongamos que una vez concluida una excavación arqueológica, deseamos «valorar» los hechos arqueológicos descubiertos con ayuda de sus correspondientes factores de certidumbre. ¿Cómo hacerlo? Desgraciadamente existen diversas interpretaciones de la Teoría de la Probabilidad (frecuentista, subjetiva) y lo que las diferencia es, precisamente, la manera de asignar probabilidades a los acontecimientos. Una manera efectiva, aunque no la mejor, puede ser la siguiente:

**D** = cerámica con decoración a base de espirales, pintada en rojo

**T** = cerámica encontrada en unidades topo-estratigráficas datadas por C-14 en el siglo IX a.C.

Para calcular la probabilidad de la expresión

«La cerámica con decoración a base de espirales pintada en rojo, se puede datar en el siglo IX a.C.»

Necesitamos una muestra de cerámicas lo suficientemente completas (o bien restauradas) que permitan la descripción del esquema decorativo, y para las cuales se dispongan de suficientes dataciones de C-14. La costumbre de los estadísticos es que una muestra de esas características ha de tener, al menos, 30 individuos. En el caso de disponer esa muestra, calcularíamos:

$$P(T|D) = \frac{\text{cantidad de cerámicas deco. datadas en el s. IX a.C. por C-14}}{\text{cantidad de cerámica deco. identificada en el yacimiento}}$$

Suele definirse la noción de *regla probabilística* del siguiente modo: aquella expresión del tipo SI...ENTONCES cuya característica principal es la propiedad siguiente: « si la proposición *y* existe en la base de Conocimientos, entonces hay una probabilidad *p* de que una proposición distinta, *x*, sea verdadera, y una probabilidad *1-p* que su opuesto sea falso. En resumidas cuentas:

*Si existe y, entonces existe x con probabilidad p.*

Resulta fácil de ver que una regla probabilística se limita a asociar entre sí hechos probabilísticos. En este caso hay que diferenciar entre los factores de certidumbre de las premisas y los de la conclusión. Los que aparecen en la premisa de las reglas actúan, en realidad, como «límites» o condiciones de activación, es decir, la certidumbre mínima que ha de tener una premisa para que la conclusión de la regla sea activada. Supongamos que la base de hechos contiene la siguiente proposición

**YACIMIENTO A tiene CERÁMICA DE RETÍCULA BRUÑIDA (FC= 70)**

y disponemos de la siguiente regla:

**Si en el yacimiento aparece cerámica de retícula bruñida (FC=30)  
ENTONCES el yacimiento data en el siglo VIII a.C.**

En este caso, la datación del yacimiento A será, efectivamente, «siglo VIII a.C.» porque la certidumbre que tenemos en ese hecho es superior a la certidumbre mínima estipulada en la premisa de la regla. Si, por el contrario, la presencia de esa cerámica estuviese asociada a un FC de 25, la conclusión no hubiese sido activada.

Consideremos a continuación el caso del factor de certidumbre asociado a la conclusión de una regla:

**Si en el yacimiento aparece cerámica de retícula bruñida  
ENTONCES el yacimiento data en el siglo VIII a.C [FC = 60]**

**Si en el yacimiento aparece cerámica fenicia  
ENTONCES el yacimiento data en el siglo VIII a.C. [FC=90]**

Este FC mide, en realidad, la intensidad de la relación entre premisa y conclusión. Dada la misma conclusión, uno de los antecedentes tiene más fuerza predictiva que el otro porque proporciona resultados más fiables. En otras palabras, tenemos más confianza en la cerámica fenicia como elemento de datación que en la cerámica de retícula bruñida.

El factor de certidumbre asignado a la conclusión de una regla depende, a su vez, de los factores de certidumbre de sus premisas. Habitualmente se dice que la certidumbre que tengamos en las premisas se *propaga* a la conclusión de la regla que las contiene. Supongamos, por ejemplo, que el Sistema Experto que contiene las dos reglas anteriores funciona por medio de un mecanismo de encadenamiento hacia atrás. Como no hay Base de Hechos inicial, el sistema planteará diversas preguntas al usuario (al igual que hacía el programa ESTELAS). Por ejemplo:

**¿APARECE EN ESTE YACIMIENTO CERÁMICA FENICIA?**

Si el usuario está seguro de ello contestará con una cláusula como la siguiente:

**CERÁMICA FENICIA PRESENTE [FC=100]**

y el Sistema proporcionará la solución esperada:

**LA CRONOLOGÍA DEL YACIMIENTO ES SIGLO VIII [FC=90]**

Ahora bien, ¿qué hubiese sucedido si el usuario del Sistema Experto no estuviese muy seguro de la existencia de Cerámica Fenicia en el yacimiento? Quizás el yacimiento fue excavado a principios de siglo, con una metodología muy dudosa, quizás la estratigrafía original haya sido removida por clandestinos,... Supongamos que se haya respondido:

**CERÁMICA FENICIA PRESENTE [FC=50]**

Como resultado, el factor de certidumbre de la conclusión descenderá hasta  $50 * 90 / 100 = 45\%$

En general, podemos decir que el factor de certidumbre de una conclusión es el factor de certidumbre de la premisa multiplicado por el porcentaje de certidumbre de esa regla. La fórmula general será, por tanto:

$$\text{FC (conclusión)} = \text{FC (premisa)} * \text{FC (conclusión)} / 100$$

Dado que lo que en una regla es premisa en otra puede ser conclusión, los factores de certidumbre se propagan de unas reglas a otras siguiendo los canales de activación de éstas. Ello implica que la fórmula anterior se aplique iterativamente a todo el Sistema. Por ejemplo, supongamos las reglas siguientes:

- (1) SI la decoración de la cerámica que aparece en el Yacimiento B incluye motivos en zig-zag [FC=20]  
ENTONCES la cerámica del yacimiento B procede del yacimiento A [FC=50]
- (2) SI la decoración de la cerámica que aparece en el Yacimiento B incluye motivos en banda [FC=70]  
ENTONCES la cerámica del yacimiento B procede del yacimiento A [FC=100]
- (3) SI la cerámica del yacimiento B procede del yacimiento A  
ENTONCES existe una relación económica entre ambos [FC=100]

Las dos primeras reglas afirman que no toda la cerámica decorada encontrada en el yacimiento B procede del yacimiento A. La tercera regla, por su parte afirma que *siempre* que la cerámica de un yacimiento proceda de otro, existe una relación económica de algún tipo entre ambos (yobviol). De la activación conjunta de las tres reglas se deduce que la existencia de la misma cerámica en dos yacimientos no obliga a suponerlos contemporáneos. El factor de certidumbre asociado a la premisa de la primera regla nos indica que no tenemos por qué estar muy seguros de la decoración de esa cerámica para que la regla se active. Quizás el 80 % de la cerámica encontrada en B muestre motivos de ese tipo, o bien que ambos yacimientos están tan cerca el uno del otro que la hipótesis más sencilla es suponer que el más importante exporta su cerámica al menos importante; precisamente porque la hipótesis es muy simple se le ha otorgado un factor de certidumbre muy bajo. El caso contrario aparece en la segunda regla.

El objetivo del sistema es responder a la pregunta:

¿QUÉ RELACIÓN EXISTE ENTRE EL YACIMIENTO B Y EL YACIMIENTO A, BASÁNDOSE EN LA PRESENCIA O AUSENCIA DE CERÁMICAS?

Si el usuario introduce el siguiente dato empírico:

PRESENCIA DE CERÁMICA CON DECORACIÓN DE BANDAS EN EL YACIMIENTO B. (FC=80)

la solución será:

EXISTE UNA RELACIÓN ECONÓMICA ENTRE AMBOS (FC=100),

pues el motor de inferencias ha propagado el FC asociado a la conclusión en la segunda regla (FC=100). Si el dato empírico introducido por el usuario hubiese sido

PRESENCIA DE CERÁMICA CON DECORACIÓN A BASE DE ZIGZAG EN EL YACIMIENTO B (FC=100)

la respuesta sería:

EXISTE UNA RELACIÓN ECONÓMICA ENTRE AMBOS (FC=50)

Automáticamente, la certidumbre en la conclusión final se ha reducido porque el Sistema Experto está utilizando información poco fiable para llegar a ella. El factor de certidumbre en este caso es el resultado de la aplicación de la fórmula general.

Si el factor de certidumbre de todas las unidades de conocimiento contenidas en nuestra Teoría Computacional es «exacto» (0 ó 100), su propagación no plantea problema alguno: obtendremos resultados «definitivamente válidos» únicamente en el caso en que las premisas también sean «definitivamente válidas». No obstante, si éstas manifiestan factores de certidumbres inexactos (un número de 0 a 100), el resultado nunca será válido al 100 %, sino que su validez dependerá de la validez de las premisas.

Hasta aquí hemos visto el caso, relativamente sencillo, de reglas con una única premisa. Para calcular el factor de certidumbre de la conclusión en una regla con diversas premisas, cada una de ellas con diferentes factores de certidumbre, utilizaremos alguno de los siguientes mecanismos lógicos de equivalencia. [A = Primera Premisa, B = Segunda Premisa]

- el factor de certidumbre de «A y B» es el mínimo de los factores de certidumbre de A y de B.
- el factor de certidumbre de «A ó B» es el máximo de los factores de certidumbre de A y de B.
- el factor de certidumbre de «no A» es el opuesto al factor de certidumbre de A.

Así, en la regla mencionada en el ejemplo anterior, el factor de certidumbre de su conclusión sería de 60.

La mayoría de generadores de sistemas expertos utilizan el método llamado E-Mycin (por ser éste programa el primero en usarlo) para calcular la «acumulación de certidumbre procedente de las distintas premisas y obtener el factor de certidumbre de la solución al problema. Se caracteriza por el siguiente cálculo:

Supongamos que existen dos reglas que soportan una misma solución:

$$\text{Factor de Certidumbre (Regla 1)} = c1$$

$$\text{Factor de Certidumbre (Regla 2)} = c2$$

$$\text{Factor de Certidumbre(Solución)} = c1 + c2 - (c1 * c2) / 100$$

Si hay más de dos reglas que soportan la hipótesis (esto es, la solución al problema aparece en la conclusión de más de dos reglas), se aplicará la misma fórmula incrementalmente. El algoritmo general es el siguiente: si A y B son los factores de certidumbre de las conclusiones de aquellas reglas que activan la solución, la certidumbre combinada será:

$$\text{Combinada (A,B)} = A + B - (A * B) / 100$$

Si la solución está activada también por una tercera regla, cuyo factor de certidumbre es C,

$$\text{Combinada}(C(A \text{ y } B)) = C + \text{Combinada (A,B)} - (C * \text{Combinada (A,B)}) / 100$$

y así sucesivamente. El orden en el que se combina la evidencia no afecta el resultado final, puesto que podemos calcular, indistintamente:

$$\begin{aligned} \text{Combinada}(C(A \text{ y } B)) &= [C + A + B - (A * B) / 100] - [(C * (A + B - (A * B) / 100))] = \\ &= [(A + B + C) + (A * B * C) / 100] - [(A * B + A * C + B * C) / 100] = \\ &= \text{Combinada (A, (C y B))} \\ &= \text{Combinada (B, (A y C))} \end{aligned}$$

El cálculo de una solución inexacta aparece, pues, como una alternativa a los algoritmos que ya hemos visto de encadenamiento hacia adelante y encadenamiento hacia atrás, los cuales sólo son válidos en el caso de problemas con solución exacta (o verdadera o falsa). Los pasos sucesivos de este algoritmo son los siguientes:

- a) se compara el objetivo o solución a las conclusiones de todas las reglas inexactas. En todas y cada una de las reglas en las que esa comparación sea positiva, se prueban las premisas. Si todas las cláusulas son validables con arreglo al estado actual de la Base de Hechos, el factor de certidumbre de la regla así activada podrá utilizarse para calcular el de la solución.
- b) combinar los coeficientes de validación obtenidos en la evaluación de todas las reglas apropiadas.
- c) cálculo del FC de la solución:
- d) 1) definir el FC de cada cláusula en el antecedente de las reglas.
- 2) definir el FC de la conclusión de dicha regla usando las equivalencias: «y» = mínimo; «ó» = máximo.
- 3) calcular el FC de esa regla usando la fórmula anterior:

$$\text{FC (conclusión)} = \text{FC (premisa)} * \text{FC (regla)} / 100$$

En el curso de una sesión de trabajo con un Sistema Experto pueden aparecer resultados sorprendentes si aplicamos este mecanismo «probabilístico» hasta sus últimas consecuencias. No olvidemos que las premisas no son más que conclusiones de reglas anteriores, y por tanto arrastran consigo los factores de certidumbre que se les ha asignado en un momento anterior. Cuando la base de Reglas es bastante grande, el usuario pierde completamente el control de la «verosimilitud» de las conclusiones, siendo el programa el que se encarga de todo. Desgraciadamente, el programa puede incurrir en errores, por ejemplo, FC menores que cero o mayores que la unidad (mayores que 100), que por definición son imposibles. Habitualmente esos errores se achacan a la fórmula general de propagación

$$\text{Factor de Certidumbre(Solución)} = c1 + c2 - (c1 * c2) / 100$$

la cual no deriva de ninguna Teoría Axiomática, y que por tanto viola en ocasiones los principios de la Teoría de Probabilidades. Para la mayoría de investigadores el mecanismo de propagación tipo E-Mycin necesitaría, o bien una nueva definición rigurosa de las fórmulas y ecuaciones de la propagación de la certidumbre, o bien, su sustitución por un algoritmo distinto, pero que cumpliese el mismo objetivo: evaluar la transferencia de información entre una regla y otra.

Ambos enfoques están siendo abordados por distintos proyectos de investigación, sin embargo, aún no han llegado al ámbito de la informática comercial. Los programas generadores de Sistemas Expertos que se pueden adquirir en el mercado siguen implementando el mecanismo E-Mycin, aunque en los manuales se señale que *puede* conducir a error. En muchos casos ese mecanismo es efectivo y muy útil; los casos en los que el factor de certidumbre final propuesto por el programa no coincide con el que se deduciría de la Teoría están mal caracterizados. Es decir, no se sabe por qué falla la propagación de la certidumbre *en determinadas* circunstancias. Todo lo más que ofrecen los programas comerciales para solventar esas dificultades son distintas variantes de la fórmula general dejando en manos del usuario la elección entre una y otra, en el caso que la fórmula general fallase.

### **Un ejemplo más complejo**

En esta sección se desarrollará un Sistema Experto desarrollado en el Centre National de la Recherche Scientifique (U.P.R. 315) por H.P. Francfort como «experto» y M.S. Lagrange y M. Renaud como «ingenieros del conocimiento», bautizado con el nombre PALAMEDE (Francfort 1987, 1990, 1991, Francfort, Lagrange y Renaud 1989). Este complejo Sistema contiene seis módulos independientes, divididos en dos grupos:

- DOXOGRAFÍA: análisis de las Teorías sobre Urbanismo, Sociedad y Estado (USE) y análisis de las Teorías sobre la Civilización (CIV).
- FISIOGRAFÍA: Topografía de las actividades (TOP), Tecnología de los artefactos (TEC), análisis de los elementos arquitectónicos (ARCHI) y Síntesis de las Conclusiones (SYN).

Todos ellos han sido programados para estudiar, por un lado la coherencia interna de las teorías al uso sobre el origen del Estado, y por el otro para interpretar en esos mismos términos los hallazgos procedentes de las excavaciones arqueológicas. En esta sección nos limitaremos a describir el módulo TOP.

#### *1. Definición del Problema*

Lo que se pretende es inferir la presencia de clases sociales diferenciadas en un yacimiento arqueológico a partir de las asociaciones espaciales detectables en el registro arqueológico, fundamentalmente, la detección de las áreas especializadas (asociadas a actividades artesanales o propias de ciertas élites sociales).

#### *2. La solución «experta» al problema*

El Sistema Experto contiene los resultados de diversos análisis sobre la economía del yacimiento de Shor-tugaï (ca. 2200-1700 a.C.) en el Nordeste de Afganistán (Francfort 1989), aunque puede aplicarse a cualquier yacimiento protohistórico, en especial del área mediterránea. En ese yacimiento se detectaron dos fases cronológicas, A y B, topográficamente circunscritas, puesto que corresponden a las dos colinas colindantes en las que se realizó el asentamiento en momentos sucesivos (el cambio de hábitat se produciría entorno al 2000 a.C.). El arqueólogo ha efectuado un análisis de la influencia de la estructura socioeconómica en la topografía del asentamiento, investigando si las distintas unidades topográficas definidas sobre el terreno se diferencian o no en tres grandes categorías, según cual sea la función social dominante: Doméstica, Artesanal o de Prestigio. La inferencia se basa en evaluaciones *cuantitativas* de la importancia y significación de los distintos vestigios. Para ello se calcularon diferentes índices, que miden la importancia de cada una de esas funciones en cada una de las subunidades topo-estratigráficas (denominadas «emplazamientos»). Comparaciones ulteriores de esos índices permitirán inferencias acerca de la evolución social en ese asentamiento. En otras palabras, el arqueólogo ha definido un conjunto de conceptos del tipo:

FUNCTION DOMESTICA = vestigios arqueológicos relacionados con actividades de subsistencia

FUNCTION ARTESANAL = vestigios arqueológicos relacionados con la producción de bienes de consumo

FUNCTION DE PRESTIGIO = vestigios arqueológicos relacionados con élites sociales.

El Conocimiento Declarativo, pues, está constituido por esos conceptos interpretativos (y otros semejantes a ellos, como por ejemplo la noción de Comercio a Larga Distancia) y por los datos procedentes de la excavación (descripción de las unidades topo-estratigráficas, en términos de las estructuras halladas y el tipo, cantidad y materia prima de los artefactos encontrados en ellas). El Conocimiento Procedural es el conjunto de conexiones que permiten poner en relación los datos empíricos con los conceptos interpretativos, por ejemplo:

Si una unidad topo-estratigráfica (x) contiene cerámica de cocina,

Entonces el índice de Función Doméstica de esa unidad será elevado.

### 3. El Sistema Experto

Sus componentes son: a) una BASE DE HECHOS que contiene los hallazgos arqueológicos, junto con sus coordenadas espaciales; b) La BASE DE REGLAS, que enuncia en qué condiciones y en qué medida esos hallazgos son representativos de alguna de las tres funciones (calcula los índices). c) El DIAGNOSTICO final del sistema, que muestra si se manifiesta o no el predominio de una de las tres funciones (Doméstica, Artesanal o de Prestigio) en una u otra fase de ocupación del asentamiento. El Sistema, al igual que el «experto» humano, enumera los índices necesarios, declara su valor inicial («inicializa los índices») y les asigna valores específicos según el hallazgo de cierto tipo (o cantidad) de vestigios.

La Base de Hechos contiene numerosas unidades de conocimiento declarativo, por ejemplo:

«NIV-A4 es un nivel arqueológico que pertenece a la fase A»

«P1-A4-N es un emplazamiento que pertenece al Nivel NIV-A4»

«P1-A4-N tiene como cerámica dominante una cerámica de almacenaje»

«en P1-A4-N aparece un hogar»

«en P1-A4-N aparecen instrumentos de molienda»

«en P1-A4-N aparece un objeto de adorno en sílex de procedencia no local»

La Base de Reglas usará ciertas unidades de conocimiento procedural para averiguar si emplazamientos como P1-A4-N tienen una función predominantemente doméstica (economía de subsistencia), artesanal (producción de bienes de consumo) o de prestigio (carácter monumental o lujoso). Todas las reglas están construidas a partir de la noción general de «acumulación de índices».

El Sistema cuenta con un total de 100 reglas, repartidas en 36 etapas.

#### ETAPA 1 (3 reglas)

En esta etapa se crean 21 índices que se inicializan a 0. Serán utilizados más adelante:

##### ÍNDICES ASOCIADOS A LOS EMPLAZAMIENTOS:

índice de función doméstica

índice de función artesanal

índice de función de prestigio

##### ÍNDICES ASOCIADOS A LOS NIVELES ESTRATIGRAFICOS

acumulación de los resultados obtenidos con los índices asociados a emplazamientos

##### ÍNDICES ASOCIADOS A LAS FASES DE OCUPACIÓN

acumulación de los resultados obtenidos con los índices asociados a emplazamientos y niveles estratigráficos

#### ETAPAS 2-5 (37 reglas)

Estas reglas permiten asignar ciertos valores a los tres índices de emplazamiento. Son de la forma:

Si (x) es un Emplazamiento,

y en (x) aparece un objeto que sirve para pulir cuentas de collar,

Entonces, incrementar en 10 puntos el índice de función artesanal de ese emplazamiento.

o bien

Si (x) es un Emplazamiento,

y (x) tiene como cerámica dominante la cerámica de almacenamiento,

y el índice de prestigio de (x) es mayor o igual a 40,

Entonces, incrementar en 20 puntos el índice de prestigio de ese emplazamiento.

El valor específico que se asigna a cada emplazamiento tiene tan sólo un valor heurístico. Por ejemplo, ¿por qué hay que asignar 30 puntos al índice doméstico si aparecen instrumentos de molienda? El Sistema Experto es ajeno totalmente a esta pregunta; se limita ejecutar el cálculo que un arqueólogo humano considera apropiado. Se trata de un valor heurístico porque el arqueólogo lo considera como un artificio para obtener una solución *possible*.

#### **ETAPAS 6-7 (2 reglas)**

Estas reglas sirven para imprimir los resultados de los índices asociados a un Emplazamiento.

#### **ETAPA 8 (3 reglas)**

Cálculo de la función dominante en cada emplazamiento. Se usan para ello los índices calculados en las etapas anteriores. Una regla-tipo podría ser la siguiente:

- Si           (x) es un Emplazamiento,
- y           el índice artesanal de (x) es distinto de cero ó mayor o igual que su índice doméstico,
- Entonces, la función predominante de (x) es artesanal.

En otras palabras, se calcula la función predominante de un emplazamiento teniendo en cuenta la importancia de las otras dos funciones en ese mismo asentamiento (se disponen de los tres índices para cada unidad topo-estratigráfica).

#### **ETAPA 9 (4 reglas)**

Con ayuda de los índices asociados a niveles definidos en la Etapa 1, se suman por niveles los resultados de los índices asociados a Emplazamientos, lo cual totaliza el conjunto de resultados por nivel. Por ejemplo:

- Si           (x) es un Emplazamiento,
- y           (x) pertenece al Nivel (y),
- y           (y) pertenece a la fase cronológica (z),
- Entonces   – agrega el valor del índice doméstico de (x) a (y)
- agrega el valor del índice artesanal de (x) a (y)
- agrega el valor del índice de prestigio de (x) a (y)
- suma los tres índices de (y) y calcula el índice general de (y)

#### **ETAPAS 10-24 (15 reglas)**

Estas etapas contienen reglas de dos tipos: las que calculan los porcentajes (por niveles y por fases de ocupación) y las que imprimen los resultados de los índices y de dichos porcentajes. Por consiguiente, para cada nivel y cada fase de ocupación, se obtiene:

- el valor de los índices de cada una de las funciones
- para cada uno de esos índices, su porcentaje en relación a la suma de los tres (índice general).
- la cantidad de emplazamientos dedicados a cada una de las funciones predominantes.
- el porcentaje de emplazamientos dedicados a cada una de las tres funciones, en relación con la totalidad.

#### **ETAPA 25 (10 reglas)**

Comparación entre los índices asociados a cada una de las fases de ocupación, y creación de nuevos índices que midan las diferencias así registradas. El valor que se asigna a estos nuevos índices procede del cálculo de las diferencias observadas entre los índices que describen una fase y los que describen la otra. Por ejemplo:

- Si           (A) y (B) son fases de ocupación,
- y           (A) es anterior a (B),
- Entonces, calcular los siguientes índices, asociados a (B):
  - *Diferencia Doméstica* = índice doméstico de (B) - índice doméstico de (A)
  - *Diferencia Artesanal* = índice artesanal de (B) - índice Artesanal de (A)
  - *Diferencia de Prestigio* = índice de Prestigio de (B) - índice de Prestigio de (A)

Otras reglas en la misma etapa definen tres intervalos de los valores en esos nuevos índices, para interpretarlos como *Aumento*, *Disminución* o *Estabilidad* de los resultados de una fase a la siguiente. Así:

- Si (A) y (B) son fases de ocupación,
- y (A) es anterior a (B),
- y el valor de la Diferencia de Prestigio es superior al 5 %,
- Entonces el Prestigio AUMENTA en la fase (B)

#### **ETAPA 26 (5 reglas)**

En esta etapa se aborda la noción de «especialización de las áreas de actividad», que puede definirse como resultante de los tres índices anteriores. El nuevo índice se utiliza para expresar el sentido de la variación resultante entre las fases que se están comparando; es decir, no se obtiene una descripción de las áreas especializadas, sino que se pone de manifiesto el grado de la variación o diferenciación funcional entre las unidades topo-estratigráficas del yacimiento. Por ejemplo:

- Si (A) y (B) son fases de ocupación,
- y (A) es anterior a (B),
- y el índice de prestigio de (B) indica un Aumento,
- y el índice doméstico de (B) indica una disminución,
- Entonces el grado de especialización de las áreas en (B) aumenta

#### **ETAPAS 27-28 (5 reglas)**

Estas reglas permiten indicar diferentes grados en la «amplitud de la variación» de una fase a la siguiente. La amplitud se calcula por medio de la diferencia absoluta de los índices domésticos de (A) y (B). Para averiguar el grado de variación en ese índice se usa:

- Si el índice de amplitud de las diferencias de una entidad cualquiera (B) tiene un valor igual o superior a 25 e inferior a 50,
- Entonces, la amplitud de las diferencias entre una fase y otra varía *sensiblemente*.

#### **ETAPA 29 (3 reglas)**

En este momento se interpreta la combinación de las informaciones que proceden del índice de variación en la especialización de las áreas y del índice de amplitud de esa variación. Las reglas se ajustan al modo siguiente:

- Si (A) y (B) son fases de ocupación,
- y (A) es anterior a (B),
- y el índice de especialización de las áreas de (B) disminuye,
- Entonces el índice de variación de la amplitud de las diferencias adopta un valor negativo.

#### **ETAPA 30 (1 regla)**

Se crea el índice Comunicación Lejana:

- Si existe una fase de ocupación (A) y un nivel (B),
- y el nivel (B) pertenece a la fase de ocupación (A),
- y existe un emplazamiento (C) que pertenezca a la fase de ocupación (A),
- y (C) contiene cuentas de collar u objetos de adorno cuya materia prima es de origen lejano,
- Entonces, incrementar en un punto el valor del índice de Comunicación Lejana

#### **ETAPA 31 (1 regla)**

Se crean dos nuevos índices: Comercio a Larga Distancia y valor absoluto del Comercio a Larga Distancia, que servirán para comparar las dos fases de ocupación desde el punto de vista de los intercambios con otras regiones:

- Si (A) y (B) son fases de ocupación,
- y (A) es anterior a (B),
- Entonces, índice de comercio a Larga Distancia =  
[índice de Comunicación lejana de (B) - índice de comunicación lejana de (A)]

**ETAPA 32 (4 reglas)**

Estas reglas describen los grados de amplitud de la variación de Comercio a Larga Distancia entre una fase y otra:

- Si        el índice del valor absoluto de Comercio a Larga Distancia de (X) tiene un valor igual o superior a 0,50 e inferior a 0,75,  
 Entonces, la variación en dicho índice es *elevada*.

**ETAPA 33 (3 reglas)**

Estas reglas indican cual es el sentido de la variación entre fases:

- Si        el valor del índice de Comercio a Larga Distancia es inferior a cero,  
 Entonces, el Comercio a Larga Distancia *disminuye*.

*4. Funcionamiento del Sistema Experto*

Ya hemos visto en las secciones anteriores el mecanismo general de la operación de instancia (o unificación): una regla es aplicable si es posible unificar un hecho contenido en los antecedentes de la regla con un hecho existente en la Base de Hechos. Esta operación se repite tantas veces como antecedentes tenga la regla (*encadenamiento hacia atrás*). Veamos su funcionamiento con más detenimiento, usando ejemplos propios a PALAMEDE.

Sea el estado siguiente de la base de Hechos:

«El índice de Emplazamientos Artesanales de la fase de ocupación A es igual a 11. NIV-A6 es un nivel estratigráfico, asociado a la fase A, con un índice de Emplazamientos Artesanales igual a 4. P1-A6-N es un emplazamiento, que pertenece a NIV-A6, cuya función predominante es la doméstica. P2-A6-N, emplazamiento que también pertenece a NIV-A6, tiene una destinación artesanal, etc.»

Expresado en el formato «computable» en el que deben expresarse esos conocimientos para que el ordenador sea capaz de procesarlos (PALAMEDE está implementado en SNARK, un lenguaje de alto nivel bastante alejado de las convenciones del lenguaje escrito habitual), se obtiene

f1	FASE-A	NATURALEZA	FASE_OCUP
f2	FASE-A	CEM-ART	1
f3	NIV-A6	PERTENECE	FASE-A
f4	NIV-A6	NATURALEZA	NIVEL
f5	NIV-A6	CEM-ART	4
f6	P1-A6-N	PERTENECE	NIV-A6
f7	P1-A6-N	NATURALEZA	EMPLAZAMIENTO
f8	P1-A6-N	F_DOMIN	ARTESANADO
f9	P2-A6-N	PERTENECE	NIV-A6
f10	P2-A6-N	NATURALEZA	EMPLAZAMIENTO
f11	P2-A6-N	F_DOMIN	ARTESANADO
f12	NIV-A5	PERTENECE	FASE-A
f13	NIV-A5	NATURALEZA	NIVEL
f14	NIV-A5	CEM-ART	3
f15	P2-A5-N	PERTENECE	NIV-A5
f16	P2-A5-N	NATURALEZA	EMPLAZAMIENTO
f17	P2-A5-N	F_DOMIN	ARTESANADO

Es preciso aclarar que no es ésta la Base de Hechos original, sino el estado que ha adoptado después de la aplicación de diversas reglas y que los consecuentes de las mismas hayan entrado a formar parte del conocimiento declarativo de la base de Hechos.

Consideremos a continuación la regla:

«Si en un emplazamiento que pertenece a un nivel estratigráfico y a una fase de ocupación cualquiera, predomina la función artesanal, entonces, añadir 1 punto al índice de emplazamientos artesanales del nivel y de la fase de ocupación a los que pertenezca».

He aquí el aspecto de la misma regla tal y como debe implementarse para que el ordenador sea capaz de entenderla:

C1	SI	NATURALEZA	(X)	= EMPLAZAMIENTO
C2		PERTENECE	(X)	= (Y)
C3		NATURALEZA	(Y)	= NIVEL
C4		PERTENECE	(Y)	= (Z)
C5		NATURALEZA	(Z)	= FASE_OCUP
C6		F_DOMIN	(X)	= ARTESANADO
	ENTONCES			
a1		CEM-ART	(Y)	$\leftarrow +1$ CEM-ART (Y)
a2		CEM-ART	(Z)	$\leftarrow + 1$ CEM-ART (Z)

La instanciación de la regla se ajusta al esquema siguiente:

\* la Condición C1 está cumplida por f7, f10 y f16, con lo cual:

$$X = P1-A6-N \text{ ó } P2-A6-N \text{ ó } PA-A5-N$$

\* la condición C2 está cumplida por f6, f9, f15. Podemos realizar, por tanto, tres instanciações distintas:

$$x = P1-A6-N, y = NIV-A6 (f6)$$

$$x = P2-A6-N, y = NIV-A6 (f9)$$

$$x = P1-A5-N, y = NIV-A5 (f15)$$

\* la condición C3 está cumplida por f4 y f13, con las mismas instanciações posibles para (Y) que en el caso anterior.

\* la condición C4 está cumplida por f3 y f12, con:

$$Z = FASE-A$$

sin que cambien las instanciações precedentes.

\* la condición C5 está cumplida por f1. Igual que antes:

$$Z = FASE-A$$

\* la condición C6 elimina la posibilidad: X = P1-A6-N, pero como está cumplida a la vez por f11 y f17, existen aún dos sistemas de instanciación posibles:

$$X = P2-A6-N, Y = NIV-A6, Z = FASE-A$$

$$X = P1-A5-N, Y = NIV-A5, Z = FASE-A$$

En otras palabras, esta regla es activable porque todas sus condiciones existen de un modo u otro en la Base de Hechos. Una vez aplicada:

a1 modificará f5, que se convertirá en NIV-A6 CEM-ART 5

a2 modificará f2, que se convertirá en FASE-A CEM-ART 12

Las 100 reglas que componen el Sistema Experto están agrupadas en «paquetes» (las 36 etapas). Cada uno de esos paquetes dispone de un orden de prioridad (inscrito en la Agenda interna del Motor de Inferencias), lo que permite el examen individualizado de cada uno de ellos y su desactivación una vez acabadas las inferencias. La estrategia de control utilizada es la búsqueda exhaustiva, es decir, que el motor de inferencias busca todas las instanciações posibles de una regla antes de pasar a la siguiente. Esas instanciações son memorizadas para evitar que la regla sea aplicada varias veces con el mismo juego de unificaciones.

##### 5. Comparación de los resultados obtenidos por el arqueólogo y los calculados por la máquina

La conclusión propuesta por el Sistema Experto es la siguiente: «al pasar de la fase de ocupación A a la fase de ocupación B, el índice doméstico aumenta, el índice artesanal permanece estable y el índice de prestigio disminuye; por consiguiente, en la fase B del yacimiento de Shortugaï, la especialización de las áreas de actividad disminuye sensiblemente, y el comercio a larga distancia disminuye enormemente». Un módulo independiente de PALAMEDE utiliza estos resultados y una nueva base de reglas para obtener, como resultado final:

«De la fase A a la fase B el yacimiento de Shortugaï experimenta una tendencia simple hacia un sistema económico de tipo rural, menos técnico, menos productor y menos organizado».

Este resultado es comparable a las hipótesis arqueológicas formuladas por el equipo que excavó el yacimiento (cf. Francfort 1989): el asentamiento evolucionó de una «colonia» harapense (fase A), a un lugar de habitación «rural» con una cultura bactriana local. La variación del comercio a larga distancia confirma esta evolución. Es importante destacar que la importancia de las actividades artesanales no disminuye, como se había supuesto en un principio (Francfort 1984): el Sistema Experto pone a la luz un detalle importante, la estabilidad de esas áreas. Un análisis detallado de los resultados de la excavación muestra que esta estabilidad artesanal es el resultado de una transformación en el dominio de las distintas actividades, es decir, que pasan de la producción de bienes de prestigio a la de útiles domésticos, o bien, característicos de otras formas de prestigio.

Con el fin de validar los resultados del Sistema Experto, se utilizó una Base de reglas ligeramente modificada para que pudiera adaptarse a una nueva Base de Hechos. Esta vez se trataba de la excavación del yacimiento de Shahr-i Sokhta, situado en el Nordeste de Irán, cuya ocupación debió tener lugar entre el 2900 y el 1700 a.C. La interpretación de los materiales de esa excavación (Tosi 1984) insistió en el desplazamiento progresivo y a la concentración de la actividad artesanal de los talleres domésticos originales (al Este del yacimiento) a barrios especializados (al Oeste y al Sur). Esta transformación topográfica y económica iría pareja a un mayor control político: «La centralización física de la producción artesanal y el grado de especialización en aumento son procesos que se pueden dividir en dos fases: [la primera se caracterizaría por] el control de los especialistas trasladándolos de sus talleres domésticos y *resituandolos* en una nueva sección» (Tosi 1984: 35).

El Sistema Experto concluye, sin embargo, que «entre el periodo 1 y el periodo 2 en Shahr-i Sokhta el índice doméstico aumenta, el índice artesanal disminuye, el índice de prestigio disminuye. Conclusión: la especialización de las áreas disminuye sensiblemente». Estos resultados son opuestos a la hipótesis inicial de Tosi, que consideraba que la especialización de las áreas aumentaba con el tiempo. Esa disparidad puede explicarse recurriendo a tres criterios distintos:

- Los bienes de prestigio y las élites desaparecen casi totalmente en la descripción de las fases recientes -la publicación de Tosi se limita a detallar el sector artesanal durante la segunda fase-, mientras que su distribución espacial era homogénea en la fase anterior. Esta ausencia desequilibra la Base de Hechos.
- Los artesanos, diseminados en el asentamiento en la fase antigua, son sustituidos por concentraciones de talleres cuya importancia no es valorable ante la falta de cuantificación. Alguno de esos nuevos talleres, estaban en funcionamiento en la fase antigua.
- Ante la falta de información sobre la función doméstica en la publicación de referencia (Tosi 1984), ésta ha tenido que ser restituida al definir la base de Hechos.

En resumidas cuentas, el resultado del Sistema Experto en el caso de Shahr-i Sokhta no invalida la interpretación de Tosi, pues la documentación existente sobre este yacimiento no es comparable con la documentación disponible sobre Shortugaï (los datos utilizados están orientados exclusivamente hacia la demostración de la concentración espacial de la función artesanal, dejando de lado problemas tales como la ambigüedad cronológica y la cuantificación de los hallazgos).

### **Aplicaciones de los Sistemas Expertos en Arqueología**

Qué duda cabe, muchos lectores habrán considerado demasiado complejo el ejemplo anterior, totalmente alejado de su forma de trabajar, y, por lo tanto muy poco «útil» para la práctica arqueológica. El propósito de esta sección es desterrar esos tópicos infundados.

Empecemos por el problema de la «complejidad»; es fácil observar que lo que unos llaman complejidad no es más que el resultado de un elevado grado de abstracción. El cálculo de los índices para cada una de las funciones no es complicado, matemáticamente hablando, pero el uso de esos índices sí que puede plantear problemas a muchos arqueólogos. Pero, no es esto lo importante: el Sistema Experto no *exige* el uso de índices como los anteriores, sino tan sólo el mismo mecanismo de razonamiento usado por el arqueólogo. Es un experto humano el que ha considerado que el uso de tales índices tenía interés para poder interpretar los datos, lo único que *hace* el Sistema Experto es «representar» ese método de inferencia.

Por consiguiente, el lector debiera tener bien presente que lo complicado, lo realmente difícil, es interpretar los datos arqueológicos, y no programar un Sistema Experto. El ordenador ejecutará los procedimientos

de interpretación que un arqueólogo considere relevantes, es decir, su inteligencia se debe, tan sólo, a que reproduce las operaciones usadas por un agente inteligente.

En este orden de cosas ¿cuál es la utilidad «real» de un Sistema Experto? En ningún caso podemos proponerlo como sustituto del arqueólogo, sino como una mera máquina auxiliar. El proceso de interpretación es muchísimo más complicado de lo que la mayoría de arqueólogos creen. Los Sistemas Expertos ayudan a dominar esa complejidad. ¿Cree el lector que hubiese sido posible calcular y utilizar de forma coherente todos esos índices sin ayuda de PALAMEDE?

Evidentemente, el uso de índices no es el único procedimiento interpretativo utilizable por un Sistema Experto. El mismo programa PALAMEDE tan sólo lo utiliza en tres de sus módulos (TOP, TEC, ARCHI). Los restantes (USF, CIV), no detallados aquí, son muy distintos, pues simulan diferentes Teorías sobre el origen del Estado (Carneiro, Wittfogel, Claessen, entre otras) bajo la forma de reglas de producción. Esos argumentos traducidos en forma de Base de Reglas son contrastados con diferentes Bases de Hechos, cada una de las cuales representa el conocimiento declarativo que se dispone acerca de alguna cultura arqueológica. Evidentemente esos casos prácticos han sido elegidos convenientemente: algunos son sociedades con una organización típicamente estatal, pero otros son sociedades tribales o, incluso, comunidades de insectos. El objetivo es aplicar una misma Teoría a diferentes ejemplos concretos para evaluar la *predicción* resultante con ayuda de lo que se sabe acerca del ejemplo. Las conclusiones son sorprendentes: ¡en la mayoría de los casos, la comunidad de insectos se considera que tiene una organización estatal! Evidentemente, algo falla en esas teorías, y es el Sistema Experto el que ha puesto de manifiesto dicho error.

PALAMEDE es un buen ejemplo de programa destinado a la *comprobación de hipótesis en arqueología*; Es posible que haya quien crea que esa no es una actividad que deba dejarse en manos de una máquina, pues es una de las principales características del razonamiento científico. Sin embargo, programas semejantes a PALAMEDE, como por ejemplo SUPERIKON (Lagrange y Renaud 1983b, 1984, 1985, 1987. Un ejemplo muy semejante es el de Monteiro 1993) han demostrado lo confuso de las hipótesis arqueológicas expresadas en lenguaje natural, y lo apropiado de los sistemas expertos para procesarlas evitando la ambigüedad. La dificultad estriba en el excesivamente difundido desprecio por el formato lógico del razonamiento científico: a diferencia de las explicaciones narrativas usuales, que oscurecen y ocultan el mecanismo de inferencia usado por el investigador para resolver un problema, un Sistema Experto enfatiza la estructura propia de la inferencia, eliminando lo accesorio. Profundizaremos en esta cuestión en las secciones siguientes.

Pero los Sistemas Expertos también pueden usarse para facilitar operaciones más simples y bastante engorrosas. Así, por ejemplo, se pueden usar como interfaces «inteligentes» entre programas técnicos muy específicos y usuarios no entrenados en esas técnicas. Por ejemplo, entre un Sistema de Información Geográfico y el usuario. Los Sistemas de Información Geográficos (GIS: «Geographic Information Systems»), la última moda en arqueología computacional, suelen ser programas muy complejos y difíciles de usar; un sistema experto puede ayudar al arqueólogo en la formulación y modelización de su problema y en encontrar la técnica de GIS más adecuada para resolverlo (cf. Webster 1990). Existe un proyecto en el Instituto de Prehistoria y Protohistoria de la Universidad de Amsterdam para realizar un programa de este tipo (Voorrips 1991, comunicación personal). Algo semejante existe en el campo de la Estadística: sistemas expertos que ayudan a seleccionar la técnica más adecuada a un problema concreto (cf. Gale 1986). El programa VANDAL, citado en el capítulo anterior, también entraña en esa categoría, puesto que ayuda al arqueólogo a interpretar los análisis arqueométricos. Otro ejemplos típicos, esta vez en zooarqueología y osteología serían los de Brough y Parfitt (1984), Maícas (1989), González y Maícas (1991).

En relación a esas aplicaciones puede citarse un estudio piloto realizado en el Centro de Investigaciones de IBM en el Reino Unido, que desarrolla un sistema a base de reglas para analizar conjuntos de piedras y decidir cómo disponerlas para construir un muro (cf. Reilly 1989). Programas como éste pueden ser muy útiles para el estudio de las técnicas constructivas de la antigüedad e, incluso, para ayudar en la reconstrucción de estructuras arquitectónicas partiendo de los restos descubiertos durante la excavación.

En el ámbito del procesamiento de imágenes por medio de Sistemas Expertos cabe citar el proyecto de Raquel y Josep Miquel Piqué (1993). Su objetivo es crear un programa capaz de analizar las microfotografías de las muestras antracológicas (carbonos) encontradas en excavaciones arqueológicas, determinando la clasificación biológica de la muestra (la madera de la cual procede el carbón). El programa consta de dos partes principales, un conjunto de algoritmos de detección de contornos y atenuación de imagen, capaz de detectar la estructura de la muestra. Interesa especialmente determinar la conformación de los poros y traqueas presentes en la imagen. El segundo componente del programa obtiene las coordenadas X e Y de dichas características mor-

fológicas de los carbones, y mide el radio de las traqueas. Esta información extraída automáticamente de la imagen es utilizada a continuación por el sistema experto propiamente dicho, que contiene una clasificación de especies botánicas y reglas de producción cuyos antecedentes están definidos por las características medidas.

Monique H. van den Dries (1994) ha diseñado un Sistema Experto que analiza las huellas de uso de los artefactos líticos. La idea básica era desarrollar un programa que facilitase a los estudiantes de prehistoria el estudio de esta técnica; por eso el programa no lee directamente la imagen, sino que interroga al usuario acerca de la presencia de determinadas huellas, y otras características de la pieza, como morfometría, estado de conservación, etc. El programa contiene numerosas fotografías para auxiliar al estudiante en esta fase de interrogación. Una vez que el programa dispone de la información descriptiva inicial, verifica si esta información es válida desde el punto de vista informático (no genera conflictos con la información que dispone) y arqueológico (no está fragmentada o alterada por procesos post-depositacionales). A continuación se activan las reglas por encadenamiento hacia adelante hasta producir la interpretación funcional deseada, esto es, la actividad que supuestamente ha producido las huellas de uso descritas por el usuario.

Muy semejante es el sistema *FAST* ("Functional Analysis of Stone Tools"), diseñado por Roger Grace (1993). El usuario debe introducir una descripción de los materiales líticos, siendo misión del programa proporcionar una interpretación funcional de esos materiales. Los rasgos descriptivos necesarios para esa interpretación son muy diversos. Así, por ejemplo, algunas existen reglas como:

Si                   (ángulo del filo < 30°)  
Entonces       (cortar material blando) [2]

El coeficiente de confianza de la conclusión es utilizado, a su vez, para lanzar nuevas reglas:

Si                   cortar ≤ 4  
y                   raspar > 8  
entonces       seleccionar raspar

Con ayuda de diversas reglas como ésta, *FAST* es capaz de distinguir cinco acciones básicas (cortar, raspar, girar, percibir y proyectil) y tres distintas categorías de dureza (blanco, medio, duro). La aplicación de este programa a los útiles líticos encontrados en el yacimiento mesolítico inglés de Thatcham permitió la interpretación del lugar en tanto que unidad doméstica, antes que campamento de caza.

VANDAL (Vitali y Lagrange 1988, Lagrange 1988, 1989b, Lagrange y Vitali 1992) es un programa de ordenador que tiene por objeto ayudar al arqueólogo a interpretar los resultados de análisis arqueométricos, en el marco de los estudios de procedencia de artefactos. Se trata de un sistema automatizado que produce uno (o varios) «diagnósticos» según sea la procedencia del artefacto, a partir de un corpus de datos proporcionado por el usuario. VANDAL ha sido concebido, esencialmente para exponer de forma clara el potencial y los límites de la información técnica en los estudios arquicométricos de procedencia, tarea que suele ser bastante complicada para el arqueólogo.

En el ejemplo que aquí se comenta, VANDAL trata el problema específico de la producción local y el comercio de las cerámicas Calcolíticas en la zona central de los Zagros (Asia Central). La información inicial se compone, por un lado, de los análisis químicos de la cerámica combinado con un Análisis de Datos, y por el otro informaciones arqueológicas apropiadas: yacimiento en el que se hallaron esas cerámicas, su tipo (es decir, una combinación del tipo de pasta y de atributos morfológicos y estilísticos), y su datación. Se supone que estos datos constituyen una *información válida* (coeficiente de verosimilitud = 1). El propósito es comparar la composición química de diferentes grupos de cerámicas arqueológicas para poder decidir si son de origen local o importadas.

El usuario debe proporcionar al sistema los datos empíricos, esto es, una descripción de las cerámicas y de su materia prima. Por ejemplo, supongamos que el usuario introduce el siguiente conocimiento:

(1)	G2-DESC-CERAM1	FUNCTION	DESC-OBJETO
(2)	G2-DESC-CERAM1	TIPO	GEOMÉTRICO
(3)	G2-DESC-CERAM1	ID-QUIM-COMPOS	G2-REF-CERAM2
(4)	G2-REF-CERAM2	FUNCTION	REF-OBJETO
(5)	G2-REF-CERAM2	TIPO	GEOMÉTRICO

Es decir,

- (1) G2-DESC-CERAM1 es un objeto desconocido .

- (2) su tipo es «geométrico».
- (3) Tiene la misma composición química que G2-REF-CERAM2
- (4) G2-REF-CERAM2 es un grupo de cerámicas de referencia (es decir, de procedencia conocida)
- (5) su tipo es «geométrico».

En otras palabras, «existe un objeto de cerámica con la misma tipología y la misma composición química que las cerámicas del conjunto de referencia». Veamos ahora qué «conocimientos» han sido usados para definir el conjunto de cerámicas de referencia:

G2-REF-CERAM3	NATURALEZA	CERÁMICA
G2-REF-CERAM3	FUNCIÓN	REF-OBJETO
G2-REF-CERAM3	PORCENTAJE	7
G2-REF-CERAM3	TIPO	CORDADA
G2-REF-CERAM3	YACIMIENTO	YAZ-DEPE
G2-REF-CERAM3	PERÍODO	CALCOLÍTICO MEDIO 1
G2-REF-CERAM3	CANTIDAD	3
G2-MAT-PRIM-32	FUNCIÓN	MATERIA PRIMA
G2-MAT-PRIM-32	YACIMIENTO	YAZ-DEPE
G2-MAT-PRIM-32	NATURALEZA	ARCILLA
.../...		

VANDAL contiene reglas del tipo:

REGLA: PERTENECE-DFSC		
SI	(1) FUNCIÓN	(A) = DESC-OBJ
	(2) FUNCIÓN	(B) = REF-OBJ
	(3) TIPO	(A) = TIPO (B)
	(4) ID-COMP-QUIM	(A) = (B)
ENTONCES	PERTENECE (A) ← (B)	

Cuya traducción sería: «Si (A) es una cerámica desconocida, (B) es un grupo de cerámicas de referencia, (A) y (B) tienen el mismo tipo y (A) tiene la misma composición química que (B), entonces (A) pertenece al mismo grupo de referencia que (B)».

La eficacia de estos programas nos permite intuir que la aplicación por excelencia de los Sistemas Expertos en Arqueología es la implementación de Tipologías Automáticas. Cualquier arqueólogo se ha dado cuenta de lo engoroso que es clasificar artefactos cuando no se es, precisamente, un experto en la materia. Por ejemplo, un arqueólogo especializado en paleolítico encuentra en las capas superficiales de su yacimiento cerámica romana o de la Edad del Bronce. Para clasificarla correctamente debe consultar libros, muchos de ellos no actualizados, y los que sí lo están remiten constantemente a obras antiguas ilocalizables. Otro caso muy frecuente es el de la aparición de cerámica de importación en un nivel estratigráfico; lo imprescindible de su correcta clasificación está reñido con la ausencia de especialistas en cerámicas micénicas, griegas, fenicias,... En estas circunstancias un Sistema Experto puede sustituir con ventaja a un libro o, incluso, a un experto: *su contenido es el mismo que el que aparece en la obra de referencia, pero más claro y mejor ordenado*; en el Sistema Experto los criterios de clasificación son, por definición, explícitos, mientras que en la obra de referencia, lamentablemente, suele ser muy difícil averiguar por qué razón ese arqueólogo ha clasificado una pieza determinada en un grupo y no en otro. Finalmente, y, sobre todo, un Sistema Experto resulta muy fácil de actualizar, dada la modularidad de la arquitectura de la Base de Conocimientos. En resumidas cuentas, una Tipología Automática permitiría al arqueólogo introducir una Base de Datos relacional con la descripción morfométrica de sus hallazgos, y el Sistema Experto proporcionaría la cronología del yacimiento, debidamente ordenada por niveles y unidades espaciales, si es que se ha tenido la precaución de introducir las coordenadas de localización para cada artefacto (cf., por ejemplo, Ozawa 1989). Repito, no hay nada de Ciencia Ficción en esto; la tecnología actual permite hacerlo sin problemas, y a un coste accesible a cualquier Universidad o Centro de Investigaciones.

No obstante, a pesar de todas estas ventajas, aún no existen Tipologías Automáticas operativas. Podemos citar el trabajo pionero de Bishop y Thomas (1984) sobre la cerámica campaniforme británica, el de Joel Mourre (1985) sobre industria lítica, el de Ganascia et al. (1986) sobre hachas de bronce, el de Markel (1987) sobre pipas iroquesas y el de Herman (1987) sobre figurillas culturales chipriotas. Proyectos interesantes son los de Ross (1989),

sobre fibulas anglo-sajonas, el de Gegerun et al. (1990), sobre orientación de tumbas y, muy especialmente, el ensayo de análisis de morfometrías cerámicas desarrollado por Steckner (1993).

Sin embargo, conviene tener presente que los Sistemas Expertos Arqueológicos no están limitados a resolver problemas tipológicos. PALAMEDE es un buen ejemplo de programa capaz de simular el "razonamiento" empleado por los arqueólogos.

### **Sistemas Expertos y Lógica Proposicional**

La primera consecuencia que se puede extraer del uso de Sistemas Expertos para representar el razonamiento arqueológico es que la estructuración de las Teorías Arqueológicas es análoga a la noción de *Memoria Asociativa* o memoria accesible por su contenido, presentada en el capítulo 2: «está contenida esa unidad de información (los rasgos descriptivos) en la Memoria (la Base de Conocimiento)? Si la respuesta es afirmativa, averigua cuál y qué afirmaciones asociadas (expresiones conceptuales) pueden activarse». Veámos antes que, en las memorias asociativas, el conocimiento está almacenado en forma de pares asociados Estímulo-Respuesta, del tipo:

- |          |                               |
|----------|-------------------------------|
| SI       | •Presencia de Fibula de Codo. |
| ENTONCES | •Cronología: siglo IX a.C.    |

donde los estímulos no son más que un conjunto específico de atributos-clave, y las respuestas son los conceptos interpretativos asociados a esos atributos. El mecanismo lógico subyacente es el siguiente:

- |          |   |
|----------|---|
| SI       | F1 y F2 y ... y Fn son verdaderos               |
| y        | si se sabe que la regla:                        |
|          | 'SI F1 Y F2 Y ... Y Fn ENTONCES F' es verdadera |
| ENTONCES | se puede deducir que F es verdadero»            |

Como se considera que los elementos que se encuentran en la memoria de trabajo son verdaderos y que las reglas existentes en la base de Reglas son también verdaderas, este principio -el *Modus Ponens* de la Lógica Proposicional- permite utilizar las reglas cuya parte condición o antecedente corresponda al estado actual de la memoria de trabajo, con el fin de añadir nuevos elementos (el consecuente de la regla) a dicha memoria.

Por consiguiente, la modalidad de razonamiento impuesta por la peculiar estructura de los Sistemas Expertos puede caracterizarse en los siguientes términos: dados ciertos datos empíricos (observaciones) acerca de un caso arqueológico particular, y una Base de Conocimientos (hipótesis e interpretaciones consideradas válidas en una Teoría Social, Antropológica o Histórica determinada), hay que interpretar el caso particular [el problema arqueológico] en términos de la Base de Conocimientos. Esta modalidad de razonamiento suele denominarse *abducción*. Josephson et al. (1987) sugieren adoptar el siguiente pseudo-silogismo para representar las inferencias abductivas:

- D es una colección de datos (observaciones, descripciones, imágenes digitalizadas, etc.)
- H explica D (Si H fuese verdadero, entonces implicaría a D)
- Ninguna de las Hipótesis conocidas explica D mejor que H.
- Entonces, H es correcto.

Por consiguiente, dos unidades abductivamente asociadas son dos unidades entre las cuales se ha establecido una relación heurística; es decir, una relación que no está basada en la naturaleza profunda de las unidades, sino en un criterio externo establecido por el investigador. Consideremos el siguiente ejemplo: deseamos saber por qué un artefacto *a* tiene la propiedad *P* (una cronología determinada); disponemos de cierta información previa (todos los *F* son *P*), por lo tanto estableceremos una asociación abductiva entre «*a* tiene *P*» y «*a* es *F*», y utilizaremos la segunda unidad de conocimiento (*a* es *F*) para explicar por qué *a* tiene *P*. Obsérvese que la conexión entre ambas unidades no es segura; la hemos *conjeturado* sobre la base de un conocimiento general previo («todos los *F* son *P*»); es decir, el razonamiento abductivo propone ciertas hipótesis plausibles acerca de una situación inicial, con el fin de explicar unas observaciones.

Esta manera de «razonar» es la que se encuentra en el fundamento del Sistema Experto presentado al principio de este capítulo (ESTELAS), caracterizado por una secuencia finita de reglas de producción que permiten la asignación de expresiones conceptuales a las observaciones empíricas; por ejemplo, la expresión conceptual «Cronología: siglo IX a.C.» es asignada a aquellas Estelas en las que se ha representado una fibula de codo. Esa

«asignación» implica la existencia de una «asociación» entre ambas unidades de conocimiento, el concepto y el dato. Una representación gráfica del mecanismo de decisión en el caso anterior aparece en la figura 3.2. Este tipo de grafo recibe el nombre de *árbol de decisión*. Se interpreta de la siguiente manera: Supongamos que deseamos probar el objetivo «Siglo IX a.C.», para ello tendremos que averiguar si existe un objeto empírico capaz de «activar» ese concepto, por ejemplo fibulas de codo, espadas de lengua de carpa y/o cascos de cresta. Dado que estos son objetos «genéricos», es imprescindible buscar si existe un artefacto arqueológico relacionado de un modo u otro con esos objetos genéricos.

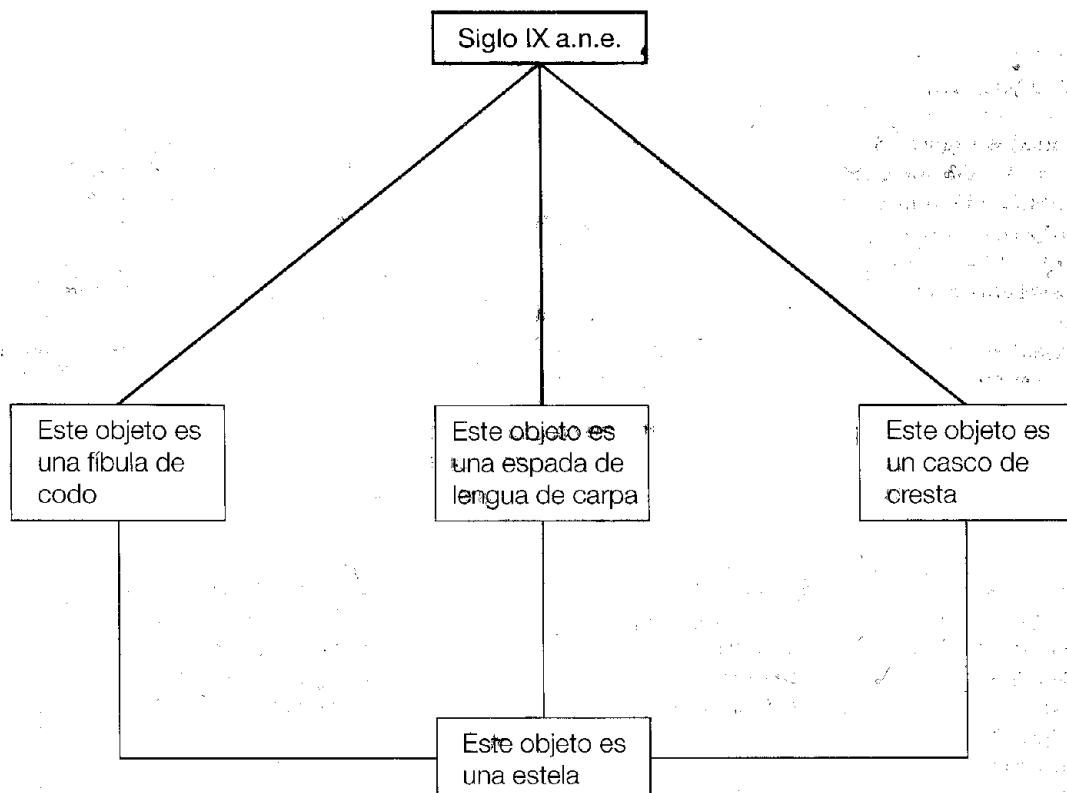


Fig. 3.2. Esquema de asociaciones abductivas en el Sistema Experto «Estelas v. 1.0».

De todo ello se deduce que para activar una unidad de conocimiento, éste situada donde esté en el árbol de decisión propio a ese problema, se necesitan dos cosas: una unidad de conocimiento y una asociación entre ésta y aquella que queremos probar (sea objetivo o subobjetivo). En otras palabras, toda inferencia se reduce a establecer asociaciones entre unidades de conocimiento predefinidas. Si no disponemos de conceptos interpretativos (esto es, si no los hemos definido previamente), jamás llegaremos a interpretar los datos empíricos: para saber si las Estelas datan o no en el siglo IX a.C. es preciso que definamos previamente el concepto «siglo IX a.C.», porque de esta definición (usando, por ejemplo, las características de la «Cultura del Bronce Atlántico III») surgirán las conexiones que permitirán asociar («activar») el estado inicial con el estado final del problema.

La formalización estricta de la noción de *activación* es muy importante para poder comprender la forma de representar «inferencias» por medio de programas de ordenador. Hemos visto en páginas anteriores que una regla contiene un antecedente y un consecuente. El antecedente enumera aquellas situaciones en las que la regla es aplicable; cuando esas condiciones son «verdaderas», es decir, existen en la Base de Hechos, bien en tanto que conocimiento directamente introducido por el usuario, bien como resultado de una regla anterior, diremos que la unidad de conocimiento representada en el consecuente de la regla ha sido *activada*. Activación e

Instanciación parten de principios distintos: en tanto que la Instanciación de las reglas puede ser definida como el proceso por el cual una regla estática es utilizada durante la inferencia y a la que se le asignan ciertos valores, la Activación es, en realidad, una función lógica que mide el grado de asociación entre el hecho que queremos interpretar (Condición de una Regla de Producción) y el concepto interpretativo (Consecuente de una Regla de Producción). La asociación entre dos o más conceptos o unidades de conocimiento ya viene dada en la formulación misma de la regla de producción:

A y B, entonces C

Donde C está asociado con A y con B, ya que su «activación» depende de la activación inicial de A y B; sin embargo, la regla no precisa qué tipo de asociación existe entre A y B.

Por otro lado, hemos visto que un Sistema Experto contiene varios centenares (o miles) de reglas. Las asociaciones entre las distintas unidades de conocimiento pueden ser extremadamente complejas: los consecuentes de determinadas reglas sirven de condición de activación de otras. Por ejemplo:

Si A y B, entonces C

Si C y D, entonces E

Si B, entonces D

Supongamos que *A* y *B* son atributos definidos en la Base de Hechos. La primera regla será, por tanto, «verdadera», con lo que el sistema *activará* la unidad *C*. El efecto de esta unidad y de la unidad *D*, activada a su vez mediante otra regla no es interpretar *A* y *B*, sino activar un nuevo concepto, *E*, cuya función será, bien activar una nueva unidad, bien interpretar el conjunto de datos descrito por *A* y *B*.

Así pues, por medio de «asociaciones», la «activación» de un concepto provocará, automáticamente, la activación de aquellos conceptos con los que el primero está asociado, y estos, a su vez, provocarán la activación de nuevos conceptos. El resultado es muy similar al de una reacción en cadena, pues la función de activación se extiende paulatinamente por todo el sistema a través de la asociación entre los conceptos que se ha implementado declarativamente (bajo la forma de reglas de producción). En cierto sentido, pues, podemos considerar que esa función hace las veces de la «energía» necesaria para que el Sistema Informático funcione y resuelva el problema que se le ha planteado, esto es, para que pueda buscar la heurísticamente mejor solución -esa analogía ha sido propuesta por Anderson (1983)-; para ello las distintas hipótesis que contenga nuestro Sistema Experto se activarán según sea la relación que mantengan con las posibles fuentes de activación (hipótesis previamente activadas y con las que está asociada).

El mecanismo de expansión de la función de activación suele ser bastante complejo, computacionalmente hablando. En realidad, la «activación» de una regla es mucho más que la mera instanciación de ciertos conocimientos implementados con anterioridad. Si el procedimiento de reconocimiento de los conceptos (hipótesis) estuviese basado en una comparación directa entre todos los componentes de dicho concepto y los del dato empírico que queremos usar para activarlo, necesitaríamos una extraordinaria capacidad de memoria en el ordenador para almacenar y calcular todas las asociaciones posibles. Esa instanciación, pues, no debe estar definida por una función directa del tipo:

«Si                      Objeto  
Entonces              Concepto»

sino por una evaluación heurística (es decir «posible» antes que «verdadera») del contexto en que esa asociación pueda llegar a producirse. El formato de la regla sería,

«Si                      (x,y,z) definen adecuadamente al Objeto F1  
y                      (v, w) definen adecuadamente el Concepto F  
y                      existe una asociación (A) entre F y F1  
Entonces              F1 activa F».

A esta complejidad de la función de activación debemos añadirle ahora las dificultades inherentes al mecanismo de expansión de la misma. Para que el sistema Experto «funcione», su motor de inferencias ha de ser capaz de provocar un movimiento en cascada o reacción en cadena entre los distintos conceptos, el cual, a su vez, permite la búsqueda y, consiguientemente, la resolución del problema. Si esa reacción en cadena o búsqueda es demasiado estrecha, posiblemente el sistema tendrá muy poca utilidad, pues difícilmente lograremos soluciones que, a primera vista parezcan innovadoras o, cuando menos, distintas a lo esperado. Si la búsqueda

da es demasiado amplia, demasiadas interpretaciones podrán ser válidas al mismo tiempo, y el sistema caerá en la incoherencia. La resolución de esta paradoja, sin embargo, está más allá del alcance de la tecnología informática, por lo que debe ser el arqueólogo el que decida, por sí mismo, el nivel de complejidad de aquel Sistema Experto que «represente» los conocimientos específicos que tiene acerca de un tema de su especialidad.

### Sistemas Expertos como «Emulación Cognitiva»

En la sección anterior hemos abordado el tema de las relaciones entre los Sistemas Expertos y la Lógica Proposicional clásica. A partir de ahora nos enfrentaremos a una de las cuestiones que más discusiones han provocado en la comunidad científica: ¿hasta qué punto un Sistema Experto reproduce la «inteligencia» de un Experto humano?

Lo cierto es que las limitaciones de estos programas son muchas e importantes, debidas, fundamentalmente a las limitaciones derivadas de la forma en que debe realizarse la implementación informática del conocimiento:

- las decisiones relevantes han de depender exclusivamente de un conjunto reducido y bien definido de variables o factores.
- los valores de esas variables han de ser conocidos (o inferibles), por lo que debe existir un método para especificarlos y expresarlos adecuadamente
- la forma precisa en que los resultados (diagnósticos o predicciones) dependen de los valores de las variables ha de conocerse y resultar «computable», al menos con una cierta aproximación
- la interrelación entre los factores que determinen el resultado han de ser lo suficientemente complejas para que valga la pena construir dicho Sistema Experto.

Limitaciones de este tipo, y aún otras no mencionadas aquí, han movido a muchos arqueólogos a criticar la tecnología de los Sistemas Expertos y su posible aplicación en la resolución de problemas arqueológicos (Hugget y Baker 1986, Wilcock 1986, 1990, Doran 1988, Gallay 1989, Shennan y Stutt 1989, Stutt y Shennan 1992). M.S. Lagrange (1989c) ha agrupado todas esas críticas:

- a) Los procedimientos de representación del conocimiento (reglas, encuadres, expresiones relacionales, etc.) son reduccionistas
- b) las reglas de producción no representan adecuadamente el mecanismo de razonamiento, porque ningún experto encuentra natural, ni tan sólo posible, proporcionarlas. Se trataría de un formalismo esencialmente pedagógico, usado por debutantes y no por los auténticos especialistas.
- c) Crítica inversa de (b): las reglas de producción son tan fáciles de formular, que resulta posible comprobar cualquier afirmación mediante un Sistema Experto.
- d) Los Sistemas Expertos obligan a fijar un estado concreto del conocimiento, por lo que rápidamente pierden actualidad.
- e) los programas actuales de Inteligencia Artificial no pueden aprender, ni generalizar, ni tratar con la incoherencia.
- f) el tratamiento de la incertidumbre en los Sistemas Expertos no está lo suficientemente perfeccionado.
- g) hasta hoy, sus aciertos concretos son poco satisfactorios (en el ámbito de la Arqueología, yo añadiría: inexistentes).

¿Debemos volver la espalda a una tecnología por el mero hecho de que no se usa? Si así lo hiciéramos aún trabajaríamos con los métodos de los anticuarios y buscadores de tesoros del siglo pasado. No cabe duda que aún faltan muchos estudios para decidir acerca de la utilidad y aplicaciones de los Sistemas Expertos; no obstante, quince años después de los primeros ensayos (Doran 1977), ya va siendo hora de que enfoquemos el problema en toda su amplitud.

Las limitaciones de la tecnología son evidentes, y en los capítulos sucesivos de este libro intentaré encontrarles solución. Por el momento, centrémonos en la que hace referencia a la capacidad de los Sistemas Expertos para representar el conocimiento científico. Prueba de lo infundado de las críticas es que para unos su formato es demasiado «rígido» (Shennan y Stutt 1989) y para otros demasiado «ambiguo» (Gallay 1989). No pongo en duda que las reglas de producción como formato de representación del conocimiento «fuerzan», en cierto sentido, el conocimiento que pretenden representar. En otras palabras, los científicos no razonan por medio de reglas; eso me parece obvio. Ahora bien, en ningún momento se ha afirmado que este formato de representación sea *análogo* al empleado por la mente humana. Recordemos lo que decíamos sobre la Teoría de la

representación: para que esa representación resulte adecuada, es preciso que exista una correspondencia que asocie la realidad exterior con la representación, de modo que cuando planteemos una pregunta acerca de la entidad real obtengamos la misma respuesta de su representación. Los estados de la entidad y de su representación estarán en correspondencia si para cada acción en el mundo real que transforme el estado actual de la entidad, existe una operación correspondiente que actualice la representación convenientemente. La existencia de una «correspondencia» entre el razonamiento científico (Descubrimiento de Regularidades Empíricas-formulación de Hipótesis-Contrastación de Hipótesis) y las reglas de producción (SI... ENTONCES...) no tiene por qué exigir la presencia de una analogía formal. Esa correspondencia puede existir a nivel de resultados, aunque la apariencia e incluso la naturaleza lógica de las mismas sea distinta.

De esta discusión hemos de aprender un principio de importancia fundamental: *la necesidad de no confundir el formato de representación con aquello que queremos representar*. La utilización de Sistemas Expertos no obliga al científico que quiera llegar a ser un buen investigador a abandonarlo todo y abrazar, como si de una fe mística se tratara, los Sistemas de Producción como panacea universal. Las reglas no son más que un útil para representar el conocimiento, y no un formato para reproducirlo exactamente, a la manera de una fotografía. La intención de los que trabajan con estos útiles no es la de producir en serie robots arqueológicos, sino analizar los mecanismos profundos del razonamiento. Si de ese análisis surgen ciertos útiles que contribuyen a automatizar ciertas tareas (Interfaces inteligentes, Tipologías Automáticas), tanto mejor.

### Comentarios bibliográficos

En este libro no se trata de la manera de construir Sistemas Expertos. Ningún libro servirá de ayuda; lo único que cabe recomendar es una lectura atenta de los Manuales de Utilización que acompañan los Programas Informáticos enumerados en el Anexo. La mayoría de ellos son lo suficientemente claros como para que el lector pueda construir un pequeño prototipo tras unas pocas horas de aprendizaje. La auténtica dificultad está en la manera de introducir el conocimiento experto en la máquina: ¡a veces, ni siquiera los arqueólogos son lo suficientemente buenos arqueólogos para saber cómo y por qué piensan como piensan! Entre la enorme cantidad de manuales de introducción hoy en día existentes, quizás debieran citarse los libros de Delahaye (1987), Keller (1987), Parsaye y Chignell (1988), Jackson (1989, segunda edición), Giarratano y Riley (1990), Benfer, Brent y Furbree (1991), Lucas y van der Graab (1991), Bundy (1990, tercera edición); en castellano, destacan el libro de Maté y Pazos (1988), la traducción del de Nebendhal (1991) y el de Cortés et al. (1993); todos ellos integran sencillez, claridad y profundidad.

## 4. DEDUCCIÓN AUTOMÁTICA

### ¿Es posible automatizar el mecanismo lógico de la Deducción?

Al empezar una sesión de trabajo con un Sistema Experto disponemos de un conjunto de unidades de conocimiento verdaderas (los datos empíricos, fruto de la observación directa del investigador, y almacenados en la Base de Hechos) y un conjunto de unidades de conocimiento «virtuales» (los conceptos interpretativos, que aparecen en la conclusión de las reglas de producción), cuyo valor verdad -en relación con la base de hechos- debe calcularse. Es decir, antes de iniciar una sesión con el Sistema Experto, los conceptos interpretativos no tienen valor verdad. Por ejemplo, un programa construido para evaluar la cronología de ciertos yacimientos arqueológicos contendrá entre sus conceptos interpretativos: «siglo IX a.C.», «siglo VIII a.C.», «siglo VII a.C.», etc. La base de Hechos contiene, pongamos por caso, la expresión «Presencia de cerámica fenicia en el yacimiento (x)». Esta expresión es verdadera, en caso contrario, no la hubiéramos incluido en la base de Hechos, a la que hemos incorporado tan sólo información cuya exactitud se conoce. Por el contrario, las expresiones «siglo IX a.C.», «siglo VIII a.C.», «siglo VII a.C.», no tendrán valor verdad hasta que sustituymos la expresión verdadera de partida por otra expresión que, incluyendo alguno de los conceptos interpretativos, también sea verdadera. A esta operación la denominaremos deducción.

Por consiguiente, «deduciremos» una proposición interpretativa, si dos (o más) unidades de conocimiento «verdaderas» pueden unirse en una nueva expresión que mantenga el valor verdad inicial. Por ejemplo, las expresiones:

«Presencia de cerámica fenicia en el yacimiento (x)» [VERDADERA]  
«la cerámica fenicia se fecha en el siglo VII a.C.» [VERDADERA]

Pueden combinarse deductivamente para formar la siguiente proposición:

«la Presencia de cerámica fenicia en el yacimiento (x) indica que éste se fecha en el siglo VII a.C.»  
[VERDADERA]

Esta operación de inferencia no es más que un silogismo de estructura clásica:

«En el yacimiento (x) hay cerámica fenicia»  
«la cerámica fenicia se fecha en el siglo VII a.C.»  
«El yacimiento (x) se fecha en el siglo VII a.C.»

Podemos utilizar los consecuentes de silogismos como el anterior para obtener proposiciones interpretativas progresivamente más abstractas, por ejemplo:

«Todos los yacimientos del siglo VII a.C. tienen construcciones defensivas»

«El yacimiento (x) tiene construcciones defensivas»

«Todos los yacimientos que reflejan una estructura política estatal tienen construcciones defensivas»

«El yacimiento (x) refleja una estructura política estatal»

La nueva proposición («El yacimiento (x) refleja una estructura política estatal») es más general que la que utilizábamos al principio («En el yacimiento (x) hay cerámica fenicia»), no sólo porque su contenido puede apli-

carse en mayor número de casos, sino porque, deductivamente hablando, la última proposición deducida «contiene» a la primera (o dato empírico), que es una «condición» para su activación.

Algunos autores han introducido, para explicar la relación deductiva que pudiese haber entre antecedente y consecuente, la noción de *similaridad literal* entre las condiciones de validez usadas para definir todos y cada uno de los conceptos (cf. Falkenhainer 1990): si y sólo si las condiciones de validez de la premisa o antecedente coinciden con las condiciones de validez del consecuente, diremos que el consecuente *ha sido deducido* de las premisas. Ahora bien, ¿cómo podemos llegar a conocer esas condiciones de validez? Muy sencillo, fijándonos en los atributos que definen el concepto.

Esta caracterización no tiene nada de extraño si tenemos presente la noción de significado defendida en la filosofía actual y que deriva, básicamente, del enfoque propuesto por Tarski: el significado no es una propiedad necesaria de la entidad, sino una propiedad del contexto en el que esa entidad tiene lugar. Son las características de ese contexto las que deciden la idoneidad de esa entidad en ese lugar del discurso preciso (cf. Johnson-Laird y Byrne 1991).

Por consiguiente, para que una regla como la siguiente:

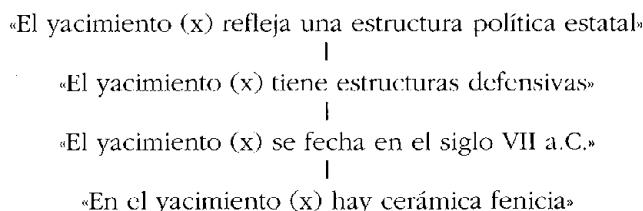
- Si           (x) se data en el siglo VII a.C.
- Entonces   (y) se data en el siglo VII a.C.

sea calificada de *deductiva* es preciso que exista similaridad literal entre *x* e *y*, esto es, que coincida la denominación y los valores de todos los atributos y subatributos que contengan.

De este enfoque se deriva el hecho que las inferencias nunca son idealmente deductivas; la *similaridad literal* no existe en el mundo real, todo lo más la relación será de una *similaridad parcial*, insuficiente para establecer una vinculación deductiva entre premisa y consecuente. Existe un continuo que sitúa en un extremo la asociación *abductiva* (cf. Capítulos 3 y 7) y en el otro la asociación *deductiva*, utilizando como criterio operativo la *similaridad parcial de las condiciones de validez*. Así, las inferencias serán *muy deductivas* o *poco deductivas* según el número de atributos que comparten el concepto o conceptos que hacen las veces de premisas y el concepto o conceptos que hacen las veces de conclusión. Por consiguiente, el *grado de deductividad* de una inferencia es una función matemática que depende del número y de la naturaleza de los atributos y subatributos compartidos por la premisa y el consecuente. Hay diferentes formas de calcular esa función, todas ellas basadas en las técnicas de la Taxonomía Numérica (Sneath y Sokal 1973), que no se detallarán aquí. Conviene citar, sin embargo, la posibilidad de utilizar el «modelo del contraste», que calcula la similaridad en función de la importancia subjetiva de los distintos atributos (Tversky 1977, Gati y Tversky 1984, Osherson 1987).

Como consecuencia de su fundamento matemático, la función «grado de deductividad» establece una jerarquía ultramétrica, no sólo entre las distintas inferencias (o asociaciones), sino entre los conceptos que toman parte en la inferencia deductiva. Es decir, siempre que razonemos deductivamente, estableceremos, consciente o inconscientemente, una relación de orden entre conceptos: *un concepto se sitúa por debajo de otro si y sólo si el conjunto de atributos que lo define está incluido en el conjunto de atributos del otro concepto* (De Boeck y Rosenberg 1988: 364). Dado que el número de condiciones de validez (atributos) del consecuente ha de ser, necesariamente, menor que los que definen las premisas (Johnson-Laird 1988), la relación de orden será, por definición, anti-simétrica: los atributos del consecuente están incluidos en los atributos de las premisas, ya que una deducción válida ha de ser verdadera en cada una de las situaciones en las que las premisas son verdaderas (Rips 1988). Dicho de otro modo, el consecuente no debe aumentar la información semántica contenida en las premisas.

Acabamos de introducir una idea de gran importancia, que nos va a permitir mejorar nuestros útiles de representación del conocimiento procedural: si las proposiciones interpretativas de un dato empírico concreto se ordenan jerárquicamente, la mayor de ellas «estará contenida» en el conjunto de todas las anteriores. Por ejemplo:



Lo que estamos estableciendo, en realidad, es un entramado de relaciones de orden entre los conceptos:  
**YACIMIENTO (X)**  
**SIGLO VII A.C.**  
**ESTRUCTURA POLÍTICA ESTATAL.**

SIGLO VII A.C. es una deducción de YACIMIENTO (X), ya que comparten (relación de similaridad) el atributo *Cerámica = Fenicia*. Del mismo modo, ESTRUCTURA POLÍTICA ESTATAL es una deducción de YACIMIENTO (X), ya que comparten el atributo *Estructuras Defensivas = Presencia*. Podemos establecer, también, una relación deductiva entre SIGLO VII a.C. y ESTRUCTURA POLÍTICA ESTATAL, considerando que el atributo *Estructuras Defensivas = Presencia* también es característico del concepto SIGLO VII a.C. El atributo que define el tercer concepto configura, claramente, un subconjunto de los atributos que definen los dos primeros conceptos, de ahí que la inferencia deductiva haya sido establecida en el siguiente orden:

[(YACIMIENTO (X)) y (SIGLO VII A.C.)] —> [ESTRUCTURA POLÍTICA ESTATAL]

Este grafo, de hecho, no es más que otra manera de implementar el silogismo general:

Si	A es verdadero para todos los B
B	es verdadero para todos los C
Entonces,	A es verdadero para todos los C

En nuestro ejemplo, disponemos de una propiedad general «Estructura política estatal», que es verdadera para un conjunto de yacimientos caracterizados por disponer de estructuras defensivas; todos los yacimientos arqueológicos con estructuras defensivas se datan en el siglo VII a.C. y en ellos aparece cerámica fenicia; en consecuencia, bastará con encontrar cerámica fenicia en un yacimiento, para *deducir* no sólo su datación en el siglo VII a.C., sino también la existencia en él de estructuras defensivas (aunque la excavación no las haya puesto de manifiesto) y la existencia de una estructura política estatal en la población que habitó dicho yacimiento.

### Relaciones Taxonómicas

El grafo presentado en la sección anterior es, en realidad, un árbol taxonómico. Se trata de una manera de representar aquella forma de razonamiento que permite inferir las propiedades de una clase o tipo basándose en las propiedades de aquellas clases o tipos situadas en un orden jerárquico más alto. Una definición formal de dicho mecanismo sería

Dado:

- un conjunto de clases
- un conjunto de propiedades para esas clases
- un conjunto de conexiones entre clases y propiedades

buscar:

- el valor más probable de la propiedad P, para la clase C
- la clase que mejor pueda describirse por medio de la propiedad P.

En nuestro caso podemos sustituir la noción de clase y propiedades por la de Hipótesis o Estado particular de un problema. Podemos emplear la modalidad de inferencia siguiente: dado un conjunto de hipótesis y una serie de conexiones entre las mismas (relaciones de orden o inclusión), encontrar el dato empírico que más verosímilmente esté asociado con la Hipótesis más general (o Solución). En otras palabras, la conexión entre Estado Inicial y Estado Final de un problema sería análoga a la relación entre un individuo o ejemplar particular y la clase general a la que pertenece.

El resultado de la ordenación jerárquica de las hipótesis es el siguiente: la más general (y, por tanto, la más abstracta) de las hipótesis contiene a las menores, las cuales tendrán acceso a un subconjunto de las propiedades que definen la hipótesis inmediatamente superior. Una vez dispuestas las clases en forma de árbol taxonómico, «exportaremos» las propiedades características de una Hipótesis (informaciones asociadas a la misma) a los datos o hipótesis particulares que forman parte de la misma. Esta operación se denomina *herencia de propiedades*.

Antes de seguir adelante analizaremos detenidamente un ejemplo muy simple. Consideremos el caso de las ánforas romanas del tipo Dressel 2-4; constituyen una «familia» o macro-tipo de ánforas que se distinguen morfológicamente de los otros macro-tipos (Dressel 1, Dressel 7-13, ... etc.), mientras que son muy similares entre ellas. La interpretación de esas ánforas no procede de su similaridad morfológica, sino del conocimiento «extraíntesco» (la función de esos contenedores de líquido) que se dispone para algunas de ellas (relacionadas con el comercio de vino). El problema pues, estriba en saber si resulta posible «exportar» a todos los miembros de la clase esa propiedad general. Consideremos que la respuesta es afirmativa: todas las ánforas Dressel 2-4 sirvieron para transportar vino; entonces, quizás fuese posible definir subclases (basadas en criterios tanto morfológicos, como de técnica de fabricación del ánfora) y exportar como propiedad, no tan general como la primera: el «lugar geográfico de producción», verosímilmente ligado con las áreas productoras de vino de calidad.

El árbol taxonómico adoptaría la forma que aparece en la Figura 5.1. Como sea que el individuo A2, por ejemplo, es miembro, simultáneamente de la Clase General y de la Clase D, «hereda» dos propiedades, la de ser un contenedor de vino y la de que ese vino procede de la provincia romana de la Narbonense. El ánfora A6, contendrá también vino, pero esta vez procedente de la provincia Tarraconense.

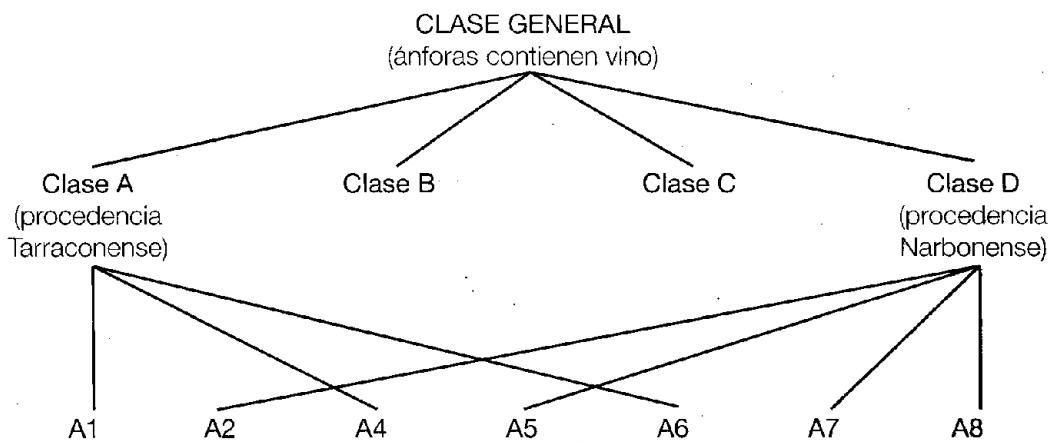


Fig. 4.1. Ejemplo de representación taxonómica de una inferencia deductiva.

Ahora bien, el Conocimiento Científico raramente se expresa en términos absolutos [todos los (x) son (y)]. Suele consistir en generalizaciones útiles, acompañadas de excepciones. Por ejemplo, si considerásemos de nuevo el caso de la datación de unos yacimientos arqueológicos a partir de la cerámica encontrada en ellos: «la cerámica de barniz rojo es característica del siglo VIII a.C., pero no es totalmente extraña en el siglo VII a.C.» Dado que en la mayoría de los casos esa cerámica es característica de los yacimientos del siglo VIII a.C., si la encontramos, deduciremos, inicialmente, que la fase de ocupación en cuestión puede datarse en el siglo VIII a.C., a no ser que haya alguna evidencia adicional que permita inclinarnos por una cronología en el siglo VII a.C., pues ha «heredado» esa propiedad de la clase general a la que pertenece «Yacimientos del siglo VII a.C.».

Por ese motivo, el árbol taxonómico suele ser conceptualmente más complejo que el que hemos visto anteriormente, permitiendo la *herencia múltiple* de propiedades. Imaginemos el tipo de relaciones que mantiene un artefacto importado (por ejemplo, la cerámica fenicia), encontrado en un yacimiento determinado (Fig. 4.2.): el Artefacto (y) hereda simultáneamente las propiedades de dos clases generales: Yacimiento A y Yacimiento B, porque ha sido encontrado en A, y fabricado en B. Supongamos que la cronología del Yacimiento A es siglo VI a.C., y la cronología del Yacimiento B, siglo VIII a.C.. Si y hereda la cronología de los dos yacimientos a los que pertenece, ¿cuál será su cronología? Una forma de resolver ese conflicto sería definiendo dos modalidades de cronología: una para la «fabricación» del objeto y otra para su uso. En el yacimiento A el artefacto y -más antiguo que los otros artefactos de ese lugar- sería una antigüedad, una pieza de museo.

En definitiva, las relaciones «taxonómicas» no son más que mecanismos para transferir información a través de los enlaces de asociación (relaciones de orden) definidos entre las unidades de conocimiento deductivo.

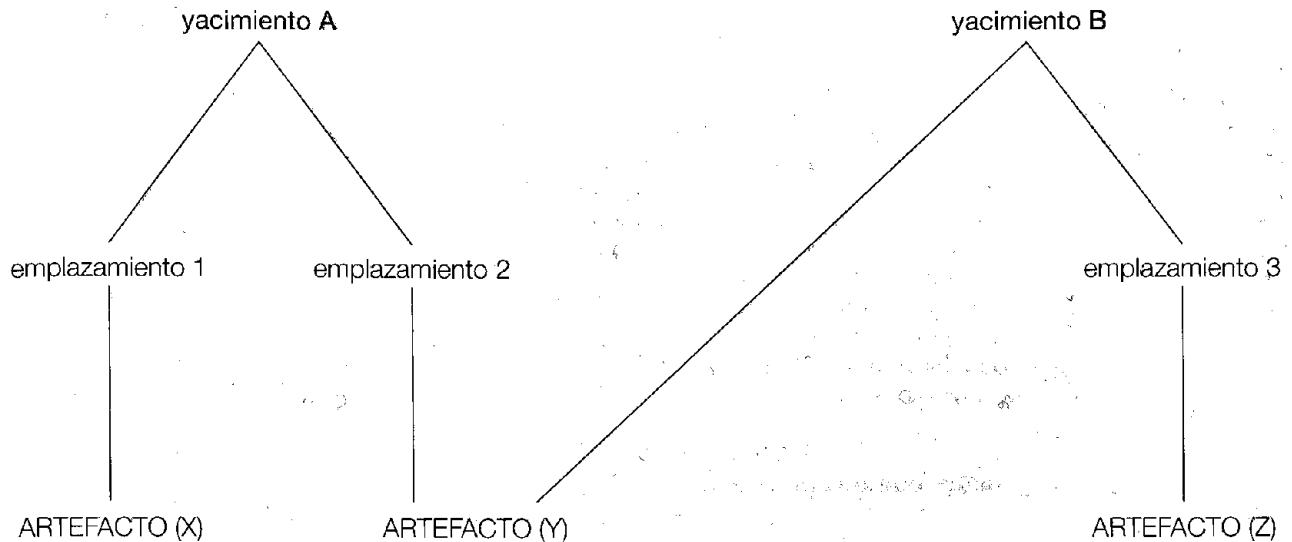


Fig. 4.2. Nuevo ejemplo de representación taxonómica de una inferencia deductiva.

vamente dispuestas. El mecanismo general puede representarse como sigue: si un artefacto **A** es un ejemplar de la clase **B** y la clase **B** es un ejemplar de la clase general **C**, entonces todas las propiedades genéricas asociadas con **C** están también asociadas con **B**, y todas las propiedades genéricas asociadas con **B** lo están también con **A**.

Suelen distinguirse dos tipos de *herencia de propiedades*. En primer lugar la «especificación» de propiedades: si una clase *a* es una especialización de una clase *b*, entonces cualquier miembro de *a* podrá usarse en lugar de un miembro de *b* (son idénticos). El segundo tipo es la «implementación» de propiedades: se crea una nueva clase *a* como modificación de una clase anterior *b*, de modo que *a* manifiesta algunas de las propiedades de *b*. Este segundo tipo permite la herencia *parcial* de propiedades, esto es, un individuo hereda *todas* las propiedades de la clase genérica a la que pertenece, excepto aquellas que se redefinan localmente.

La «implementación» de propiedades es un operador que permite crear conceptos a partir de conceptos preexistentes. La herencia de propiedades se usa entonces para definir una «modificación en incremento» del concepto inicial. Por defecto, el nuevo concepto «hereda» la representación de la clase inicial, si bien se puede extender esa representación, añadiendo nuevas variables u operadores, o bien cancelando algunos de los que caracterizaban el concepto inicial. De este modo, el concepto resultante, tendrá algunas de las propiedades del concepto inicial, más algunas de nuevo cuño. En realidad, la nueva clase no es un «subtipo» de la primera, sino una disyunción.

Recordemos lo que decíamos en la sección anterior a propósito de los Estados Finales de un problema: que «contenían» todos aquellos que le precedían y que era posible deducir la solución de un problema extrayendo el grafo (o árbol taxonómico) de las hipótesis contenidas en esa solución. De este modo podemos definir las unidades de conocimiento declarativo en términos de otras más generales (situadas más arriba en el árbol taxonómico), siempre y cuando esas unidades de conocimiento estén asociadas por medio de relaciones de orden entre las que sea posible la herencia de propiedades. Veamos algunos ejemplos,

La ordenación más sencilla sería la siguiente jerarquización de unidades espaciales :

CONTEXTO < UNIDAD DE HABITACIÓN < ESTRATO < YACIMIENTO < ÁREA < REGIÓN.

Algunas de las inferencias que hagamos sobre la región serán heredadas por todos los emplazamientos de los yacimientos en esa región. Qué subconjunto de propiedades de la clase general es el heredado por las entidades particulares, es algo que queda a la discreción del arqueólogo. El libro clásico de Clarke (1978) discute ampliamente la ordenación jerárquica de las unidades arqueológicas, y las consecuencias de la herencia de propiedades entre ellas.

Sin embargo, el caso o más interesante lo constituye la representación de la ordenación clásica de las Hipótesis Científicas según su componente observacional. Por ejemplo:

OBSIDIANA < INTERCAMBIO < ECONOMÍA

La «presencia de obsidiana» es un Dato Empírico fruto de una observación directa sobre el terreno. Estará enunciado, por tanto, en términos *observacionales*. La unidad de conocimiento «Intercambio», por el contrario, es una Hipótesis (o Estado transitorio de un problema), y se enuncia por medio de entidades *teóricas*, sin componente observacional. La herencia de propiedades, pues, se manifiesta como el útil de representación idóneo para estos casos: la proposición interpretativa resultante de la asociación entre el dato Empírico y el Concepto -esto es, asociación entre la obsidiana y la acción social responsable de su aparición en ese lugar (el Intercambio)- será «verdadera» porque la información contenida en el concepto (una forma económica particular) habrá sido asignada automáticamente al Dato. En otras palabras, el dato puede acceder directamente al conocimiento contenido en el Concepto.

### **La Herencia de Propiedades como representación del Conocimiento Procedural**

Hemos visto que cualquier operación de *inferencia* puede definirse como la combinación de dos unidades de conocimiento para producir nuevo conocimiento. Esta definición era análoga a la del mecanismo de la *deducción*, que es, fundamentalmente, un procedimiento de substitución de expresiones por expresiones lógicamente equivalentes. La forma general :

$$C \rightarrow P = Q$$

expresa la sustitución de P por Q, siempre y cuando se pueda probar que C es verdadero. En esta sección veremos cómo las inferencias pueden ser representadas también por medio de un árbol o grafo taxonómico: la «verdad» de C se substituiría entonces por la noción de «presente» en el grafo como arco (enlace) entre los nodos P y Q. La única condición para que Q sea una expresión lógicamente equivalente a P es que exista una relación taxonómica, es decir, una relación de orden que permita la herencia de propiedades, entre ambas. En otras palabras, una proposición interpretativa será verdadera si toda la información contenida en un Concepto Interpretativo fluye directamente al Dato Empírico que queremos interpretar. Denominaremos *inferencia* a ese «flujo de información», pues se trata de una noción perfectamente equivalente a la que hemos usado anteriormente.

En principio, toda operación de inferencia podrá representarse por medio de conexiones entre unidades de conocimiento que permitan la herencia de propiedades; esto es, el enunciado de las expresiones *deducibles* de una hipótesis equivaldrá a la especificación de los enlaces *permitidos* por un árbol taxonómico (en este caso los enlaces *permitidos* son los enlaces definidos explícitamente por el investigador). Esos enlaces están representados bajo la forma de las «ramas» del árbol taxonómico. Por consiguiente, afirmaremos que el propósito de un árbol taxonómico es especificar el conjunto de afirmaciones *permitidas* por un conjunto ordenado de Hipótesis o Conceptos Interpretativos; es decir, su función se reduce a averiguar las afirmaciones a las que se podría llegar usando esas Hipótesis como información inicial.

Supongamos que disponemos de un conjunto de Hipótesis Interpretativas, ya sean éstas términos cronológicos, categorías sociales, funciones de artefactos, etc. Para poder representar con éxito cualquier tipo de inferencia con esos materiales es preciso, en primer lugar, *estructurar* ese conjunto de unidades de conocimiento. Esto es algo que no debiera sorprender a nadie: cualquier Teoría Científica no es más que un conjunto debidamente estructurado de proposiciones interpretativas. Si utilizamos un árbol taxonómico para representar la deducción de todas las proposiciones interpretativas posibles a partir de esa Teoría, las unidades de conocimiento disponibles deberán disponerse en un orden jerárquico, que puede llegar a ser muy complejo. Este orden jerárquico se consigue estableciendo relaciones de inclusión o, como mínimo, enlaces del tipo  $\leq$  entre ellas.

Toda inferencia está representada por secuencias de unidades de conocimiento enlazadas. Resulta conveniente otorgar al último enlace la función de *razón* o *motivo* de la inferencia. Así por ejemplo, sea la cadena taxonómica que mencionábamos al final de la sección anterior:

OBSIDIANA < INTERCAMBIO < ECONOMÍA

Podremos interpretar el dato Empírico (la obsidiana) con ayuda del Concepto Interpretativo (el Intercambio como forma económica), si existe entre ambos algún tipo de enlace que permita a la entidad «obsidiana» heredar algunas de los atributos característicos de la entidad «Intercambio», algunos de los cuales, a su vez, proceden de la entidad «Economía».

Estas relaciones de orden entre unidades de conocimiento no son, en modo alguno, leyes empíricas. Las unidades de conocimiento no están ordenadas en el mundo real, sino que somos nosotros, investigadores a los que interesa estudiar el razonamiento científico, los que hemos establecido dicho orden taxonómico. Se trata de un artificio computacional que permite encontrar una solución más elegante (esto es, más breve y compacta) y en menos tiempo. El único requisito es establecer un criterio de ordenación y mantenerlo en todas las unidades de conocimiento implicadas en la inferencia. Es preciso tener bien presente, por tanto, que la «Herencia de Propiedades» y los enlaces «≤» son formatos de representación del conocimiento procedural y no el resultado de ciertas observaciones empíricas; tanto los enlaces como el mecanismo mismo de la «herencia de propiedades» pueden definirse como herramientas para trasladar información de una unidad de conocimiento a otra. La dificultad mayor durante la configuración de un árbol taxonómico será pues, la construcción de los diferentes enlaces necesarios.

¿Cuál es el enlace que existe entre unidades tales como «Obsidiana» e «Intercambio»? ¿Y entre «Espada de Lengua de Carpa» y «Siglo IX a.C.»? Esta clase de relaciones entre unidades suele recibir el nombre de ES-UN (en inglés «ISA»), es decir:

- «la obsidiana ES UN resultado del Intercambio»;
- «la Espada de Lengua de Carpa ES UN ejemplo característico del siglo IX a.C.».

Relaciones muy semejantes son las denominadas PARTE-DE, TIPO-DE y INSTANCIA-DE, por ejemplo:

- «borde=10cm.» es una PARTE-DE «ejemplar nº 12»;
- «el ejemplar nº 12» es una INSTANCIA-DE la clase «ánfora».

Los cuatro tipos de enlace son, en realidad, expresiones relacionales que representan un cierto tipo de conocimiento procedural. Este tipo de enlaces relationales, en realidad operadores, sustituyen con éxito en muchos casos a las Reglas de Producción.

La operación lógica ejecutada por estos enlaces es bastante exigente -en Lógica de Predicados suele expresarse por medio de implicaciones-:

si A es-un B, entonces todo lo que resulta verdadero en B ha de ser también verdadero en A.

Del mismo modo, cualquier cambio en B ha de repercutir directamente en A.

Este principio tiene consecuencias muy importantes. Sea cual sea el tipo de relación taxonómica en la que se base el sistema («especificación» o «implementación»), entre dos unidades unidas por una relación taxonómica no puede existir la inconsistencia. En otras palabras, aunque la unidad de nivel jerárquico inferior no herede todas las propiedades de la unidad general a la que está conectada, la definición local de sus propiedades no puede contradecir lo que ya contiene la unidad general. De hecho, no todas las entidades admiten este tipo de representación. Por ejemplo, en el estudio de una necrópolis no podríamos establecer ninguna relación taxonómica entre tumbas ricas y pobres, o entre tumbas femeninas y masculinas. Aunque pueda darse el caso que en las tumbas ricas y masculinas aparezca la totalidad del ajuar existente en la necrópolis y que las tumbas pobres (o las femeninas) aparezcan como subtipos especializados de las tumbas masculinas ricas, esa ordenación es tan sólo aparente, pues no toda hipótesis válida para unas será válida para las otras.

La segunda condición expresada en el enunciado de la propiedad de consistencia para la herencia de propiedades hace referencia a la Lógica No-Monótona, que trataremos brevemente en el capítulo 7. No obstante, podemos adelantar algunos puntos importantes. En primer lugar, un árbol taxonómico no es formalmente completo. Dado que el usuario se responsabiliza de la construcción de las relaciones taxonómicas, siempre se pueden añadir nuevos enlaces entre unidades de conocimiento previamente no asociadas. No obstante, al añadir nuevas vías de circulación a la información, el árbol taxonómico se mostrará incapaz de mantener las mismas inferencias que en su estado inicial. Es decir, las conclusiones a las que se puede llegar mediante un árbol taxonómico son aquellas inferibles en el estado actual del mismo. Si cambiamos cualquiera de los enlaces, el árbol taxonómico no reproducirá proposiciones deducidas con anterioridad.

Expongamos esas características por medio de un ejemplo:

1. la Cerámica de Retícula Bruñida data en los Siglos IX- VIII a.C.

2. la cerámica encontrada en el yacimiento de Atalaia (Alentejo, Portugal) tiene una cronología desconocida
3. en el Yacimiento de Ratinhos (Alentejo) hay cerámica de Retícula Bruñida
4. en el Yacimiento de Ratinhos (Alentejo) hay cerámica semejante a la de Atalaia

En muchas ocasiones, los arqueólogos han utilizado la presencia conjunta de cerámicas distintas en un yacimiento para imponer una de las cronologías, sin tener en cuenta las implicaciones de la inferencia. El caso anterior (Schubart 1975) es un buen ejemplo de ello. Como se desconoce la cronología de la cerámica de Atalaia, ésta «hereda» las propiedades cronológicas de la cerámica de Retícula Bruñida encontrada en el yacimiento de Ratinhos. Ahora bien, ¿están ambos tipos cerámicos unidos por una asociación que permita la herencia de propiedades? Se parte de la suposición que la asociación es estratigráfica, por lo que la herencia de propiedades tendrá validez en ese caso; sin embargo, esa asociación estratigráfica bien pudiera ser errónea (Ratinhos nunca ha sido excavado, así que no sabemos si ambos tipos cerámicos aparecen en el mismo estrato o pertenecen a fases de ocupación distintas). Bastantes años después de la formulación del silogismo anterior, se obtienen dataciones C-14 de unas cerámicas semejantes a las de Atalaia encontradas en la necrópolis de Pomar (Varela Gomes, comunicación personal, cf. Barceló 1991); esas dataciones se refieren a una cronología entorno el 1800-1500 a.C. ¿Debemos concluir automáticamente que el silogismo anterior es falso?

Obsérvese que lo único que cambia es la definición local de la propiedad «Cronología» en la unidad «Cerámica de Atalaia». No hemos introducido enlaces nuevos ni suprimido los anteriores. Sin embargo, la redefinición a nivel local de la propiedad genera una contradicción con las propiedades definidas a nivel general, y por ende, con la herencia de propiedades como mecanismo de inferencia. Podemos optar por dos soluciones:

- poner en duda el enlace entre ambas unidades: la asociación espacial en un mismo yacimiento no permite la herencia de propiedades.
- redefinir el mecanismo de la herencia de propiedades para que admita la gestión de *excepciones*.

Es evidente que, en el ejemplo anterior, la respuesta más adecuada es la no asociación cronológica entre ambos tipos de cerámica; es un caso claro de razonamiento no-monótono: la agregación de nuevo conocimiento invalida las afirmaciones conseguidas con anterioridad. Sin embargo, y a título tan sólo de ejercicio intelectual, veamos lo que pasaría si redefiniésemos el mecanismo de inferencia.

El árbol taxonómico debería aceptar la Herencia Múltiple. Su enunciado formal sería: una unidad A está relacionada taxonómicamente a B (es decir, A es menor que B y hereda las propiedades de B); esa misma unidad A también está relacionada taxonómicamente a C; si A hereda P a partir de B, y  $\neg P$  a partir de C, entonces, siempre y cuando exista un enlace taxonómico entre B y C (y no viceversa), concluir P; en el caso que exista un enlace taxonómico entre C y B (y no viceversa), concluir  $\neg P$ ; si no hay relación alguna, el resultado es ambiguo. En este último caso, lo habitual es *neutralizar* o inhibir el enlace conflictivo, de modo que la inferencia sea imposible.

El ejemplo arqueológico que estamos analizando se representaría del modo que aparece en la Figura 5.3. Es evidente que, en nuestro caso el razonamiento taxonómico produce una ambigüedad. La unidad «Cerámica de Atalaia» hereda los valores de la propiedad «Cronología» de dos unidades opuestas: «Necrópolis do Pomar (1500 a.C.)» y Ratinhos (800 a.C.). Una primera solución sería elegir uno de los dos enlaces conflictivos que exportan propiedades contradictorias a la unidad «Cerámica de Atalaia». D.S.Touretzky (1986, cf. también Harty et al. 1990) sugiere otorgar más importancia a los enlaces simples que a los compuestos. De este modo, la relación Cerámica de Atalaia/Pomar predominaría sobre la relación Cerámica de Atalaia/Ratinhos/Siglo IX, por la única razón que una cadena de inferencias es más breve que la otra.

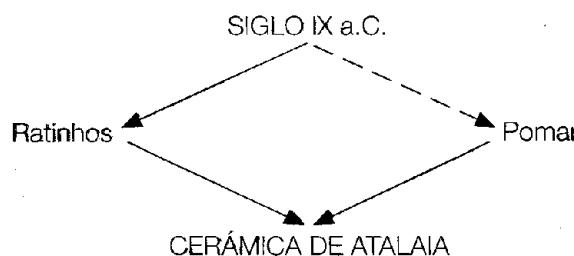


Fig. 4.3. Ejemplo de resolución de contradicciones en un árbol taxonómico.

L. Shastri (1988) proporciona otra solución a ese conflicto. Según este autor el problema se debe a la falta de cuantificadores lógicos («Existe algún...») en los árboles taxonómicos. En nuestro caso el conflicto se produce porque la herencia de propiedades da lugar a las siguientes proposiciones

*Toda la Cerámica de Ratinhos data en el 1500 a.C.*

*Toda la Cerámica de Ratinhos data en el 800 a.C.*

Para permitir enunciados existenciales del tipo:

*Existe alguna Cerámica de Ratinhos en el 1500 a.C.*

*Existe alguna Cerámica de Ratinhos en el 800 a.C.*

Shastri propone sustituir el cuantificador lógico por una medida precisa. Por ejemplo:

- el 70 % de la cerámica encontrada en Ratinhos (sea del tipo que sea) se puede datar en el siglo IX a.C.
- el 30 % de la cerámica encontrada en Ratinhos no se puede datar
- el 90 % de la cerámica encontrada en Pomar no data en el siglo IX a.C.

La evaluación probabilística

[total de la cerámica datada en el siglo IX a.C.]

[total de la cerámica que no data en el siglo IX a.C.]

de estas cantidades proporcionaría una estimación de la «intensidad» de la asociación con las clases generales: la herencia de propiedades circularía, por consiguiente, a través de los enlaces más «intensos», pero sin dejar de lado los restantes.

En definitiva, tres soluciones parecen posibles al problema de las excepciones en situaciones de Herencia Múltiple de propiedades:

- la eliminación sistemática de aquellos enlaces que produzcan inferencias ambiguas
- la «decisión» probabilística del mejor de los enlaces posibles
- la aceptación de la ambigüedad resultante como una inferencia posible; en ese caso el razonamiento taxonómico preservaría la naturaleza del conocimiento., pues variarían las características de las propiedades heredadas.

En nuestro ejemplo arqueológico, la ambigüedad de la inferencia final es probablemente significativa: «Cronología» es una propiedad *necesaria* para las unidades «Pomar» y «Ratinhos», ya que es esa propiedad la que discrimina entre ambas; sin embargo, en el caso de «Cerámica de Atalaia», la propiedad «Cronología» sería *no-necesaria*, pues es ambigua y no contribuye a definir esa unidad: si la cerámica de Atalaia tuviese una duración de más de setecientos años, podríamos encontrarla tanto en yacimientos característicos del año 1500 a.C., como del año 800 a.C.

Conviene no dejarnos llevar por los problemas inherentes al tratamiento de excepciones y conflictos. En la mayoría de las ocasiones no se trata de un «fallo» del mecanismo de herencia de propiedades, sino de un enlace situado donde no debe. No todas las unidades de un discurso están relacionadas taxonómicamente. Desgraciadamente, aún no existe un procedimiento para detectar los errores en dicho mecanismo de inferencia: en bastantes ocasiones, un resultado ambiguo es interesante para el investigador (como en el caso de la indeterminación cronológica), aunque rechazable para el lógico.

### Sistemas Expertos y Relaciones Taxonómicas

Así pues, el árbol taxonómico parece ser un modo muy útil para representar inferencias deductivas, gracias a las relaciones jerárquicas que impone a las hipótesis o proposiciones interpretativas. Ahora bien, aunque computacionalmente «efectivo», no se trata de un modo de representación tan «expresivo» como el de las reglas de producción. Es cierto que los Sistemas Expertos estudiados en el capítulo anterior manifiestan dificultades con la representación de relaciones jerárquicas (el orden de activación de las reglas es independiente del conocimiento implementado en ellas), pero ello no debiera llevarnos a sustituir completamente un modo de representación del conocimiento procedural por otro, sino a integrarlos, esto es, a usar ambos en un mismo caso. Afortunadamente, existen útiles informáticos de difusión comercial que permiten esa integración.

Tenemos dos opciones para llevar a cabo esa integración, una supone dar más importancia a las Reglas de Producción y la otra a las Relaciones Taxonómicas. En el primer caso, la herencia de propiedades sirve tan sólo para aumentar la flexibilidad de la Base de Hechos del Sistema Experto: las relaciones taxonómicas simulan en ese caso las *relaciones* de una Base de Datos Relacional, lo que permite al Sistema Experto acceder directamente al conocimiento declarativo escrito en ese formato, mucho más comprensible, completo y fácil de usar e implementar que una base de hechos construida mediante triplets <OBJETO><RELACION><VALOR>. En esos casos, la Herencia de propiedades es un mero recurso técnico para reducir la cantidad de información que hay que introducir en el ordenador: basta con la definición de las variables o atributos de las clases o prototipos más generales, pues sus valores aparecerán automáticamente en las instancias de dichas clases. El conocimiento procedural está expresado exclusivamente bajo la forma de reglas, cuyas conclusiones pueden acumularse en esa misma Base de Hechos taxonómica, o bien en una Base de Hechos clásica.

En el segundo caso, el predominio de las Relaciones Taxonómicas sobre las Reglas de Producción, la inferencia estará representada por el árbol taxonómico. El Motor de Inferencia usará la herencia de propiedades para localizar y/o calcular el valor de un atributo en un nodo concreto del árbol, siendo la función de las reglas la de controlar la ejecución y servir de estructura de control en momentos puntuales.

Podemos sustituir la representación del conocimiento declarativo por medio de triplets por otro formato de representación: los *encuadres* (en inglés "frames"). Se trata de un tipo de estructura de datos cuyo tamaño y alcance resulta mayor que el de una proposición, dado que representa no sólo el contenido puramente declarativo de la unidad, sino también una parte del procedural (las relaciones de pertenencia a una categoría más general). Para construir tales unidades computacionales habremos de asociar a cada unidad un conjunto de propiedades distintivas o *atributos* (en inglés se ha popularizado el término «slot»). Así, Patel y Stutt (1988) describen del siguiente modo los restos de una unidad de habitación:

#### CERCADO

*anchura*: 20  
*longitud*: 10  
*situación*: (10 30)  
*material*: BARRO  
*ubicación*: SUPERFICIE  
*radio*: [-]  
*forma* CUADRANGULAR

Fue M. Minsky (1975) el primero en utilizar el término «encuadre» para construir una base de datos estructurada y flexible que contuviese las cantidades enciclopédicas de conocimiento necesarias para el razonamiento por sentido común. La idea básica consistía en organizar la representación de un dominio específico por medio de una secuencia de «encuadres» del mismo; esas unidades fundamentales de representación intentaban capturar en un formato fijo lo que es común a ciertos hechos, situaciones o dominios de un mismo tipo. Hoy en día este tipo de representación del conocimiento se utiliza en todos los ámbitos; prácticamente cualquier objeto, hecho o fenómeno pueden ser descritos en términos de un encuadre, siempre y cuando usemos en su construcción los dos tipos de información siguientes:

- el valor de una propiedad particular de un objeto
- la categoría a la cual pertenece dicho objeto.

Los componentes básicos de este modo de representación son:

- NOMBRE.- cada encuadre tiene su propio nombre (por ejemplo «Ánfora», A12, ...)
- ANTEPASADO O TIPO.- cada objeto está relacionado, como mínimo a otro por medio de un enlace jerárquico que permite la *herencia* de propiedades.
- ATRIBUTOS.- el cuerpo de un encuadre está compuesto por cierto número de atributos. Estos atributos puede ser, a su vez, encuadres.
- PREDICADOS ASOCIADOS.- Cada atributo en un encuadre puede tener predicados asociados que serán invocados siempre que ese atributo sea leído o modificado.

En los Sistemas a base de Encuadres, hay varias maneras de declarar el valor propio a cada atributo:

- un VALOR: que contiene el valor actual declarado del atributo. Corresponde a una información que la entidad deberá poseer para poder ser adecuadamente representada por ese atributo.

- por DEFECTO: contiene el valor que se va a dar al atributo en el caso de que no se disponga de ninguna otra información.
- por REGLA: se trata de procedimientos ligados a cada atributo, que se activan en el momento de manipularlo. Dos modalidades:
  - reglas que son activadas cuando se introduce una nueva información en un atributo. Su función es propagar esa información a través de las relaciones taxonómicas y estructurales.
  - reglas que se activan en el momento en que resulta necesario dar un valor a un atributo y éste no existe. Su función es, pues, la de proporcionar un procedimiento de cálculo o adquisición de ese valor.

La función primordial de las reglas de producción en un árbol taxonómico es, pues, declarar localmente el valor de ciertos atributos. De este modo, se pueden programar en forma de regla de producción las condiciones que deben ser satisfechas antes de que se asigne un valor a esa propiedad en ese nodo particular del árbol. A cada una de esas condiciones está asociado uno (o varios) predicados con cierta información procedural, de forma que si se necesita el valor preciso de un atributo para activar una regla, entonces se ejecuta antes el predicado asociado a ese atributo. Igualmente, antes de fijar el valor de un atributo, el sistema ejecuta los predicados.

Un ejemplo idóneo lo constituyen los caracteres morfométricos que deben medirse en cada vasija.

*Encuadre: A12*

*Es-un: Ánfora*

*Propiedades: índice Anchura/Longitud:*

Ejecutar la función «Introducción de datos»

Ejecutar la función «Cálculo índice A/L»

La función «Introducción de Datos» está constituida por ciertas rutinas que solicitan del usuario la longitud total del ánfora A12 y su anchura; la función «Cálculo índice A/L» ejecuta ciertas operaciones aritméticas con los valores introducidos por el usuario, y las sitúa en la posición *valor* del atributo *índice Anchura/Longitud*. De este modo, los valores de las distintas propiedades podrán definirse localmente; es decir, serán calculados cada vez que se use el Sistema, y sólo serán válidos en la sesión en curso.

La idea básica es que siempre que accedamos a un nodo del árbol taxonómico accederemos a todos los nodos que dependen de él, esto es, a todos los que están relacionados taxonómicamente y con los cuales la herencia de propiedades está permitida. Interrogaremos el árbol taxonómico como si de una Base de Datos relacional se tratase, solicitando los valores que ciertos objetos han adoptado. Por ejemplo

**x := (tipo de A12)**

que solicita del Sistema la respuesta a una pregunta como ésta: «¿cuál es el tipo del individuo A12? Imaginemos que A12 es una instancia del objeto «Ánfora», el cual *es-una* «Cerámica». Uno de los atributos de este objeto es «Tipo», definido localmente en el objeto «ánfora». El Motor de Inferencias peculiar al sistema iniciará la búsqueda en A12 del atributo «Tipo» y de su valor. Si éste no ha sido definido a ese nivel, entonces debe existir, *necesariamente*, en alguno de los objetos de los que depende taxonómicamente. Ese es el caso del objeto «ánfora». La respuesta será, pues:

**ánfora = (tipo de A12).**

Ahora bien, hasta aquí sólo hemos usado las propiedades «declarativas» de la herencia de propiedades. Hemos visto que no sólo sirven para describir conceptos, sino que puede usarse también como representación procedural, esto es, representación del mecanismo de solución de un problema. Para ello, cada encuadre representará una de las soluciones posibles (o estados intermedios), siendo el encuadre situado en la cúspide del árbol taxonómico el que represente el Estado Final del Problema. En la parte inferior del árbol representaremos los Estados Iniciales, ya que un mismo problema puede enunciarse de muy distintos modos. Cada uno de esos Estados Iniciales es, a su vez, un encuadre unido taxonómicamente a los demás. Para interpretar el caso concreto que estamos analizando debemos encontrar un Estado Inicial predefinido que sea análogo a nuestro caso, y averiguar a qué Estado Final está unido, esto es, qué propiedades hereda de los encuadres situados más arriba en el árbol taxonómico.

Una estrategia de resolución de problemas como la que se acaba de enunciar exige construir una Teoría Científica bajo la forma de un árbol taxonómico de encuadres. Desgraciadamente, en la mayoría de los ca-

sos, las Teorías con las que trabajan los arqueólogos (las «Teorías de Alcance Medio») suelen ser, en el mejor de los casos, incompletas; pero para que la Herencia de Propiedades sea un mecanismo efectivo, el árbol taxonómico ha de estar completo. Esta contradicción puede resolverse integrando el sistema a base de Encuadres con una Base de Reglas de Producción.

La función de esas Reglas es «cubrir» los posibles huecos que existan en el árbol taxonómico, calculando el valor de algunas propiedades en el momento de la ejecución, o bien modificando los resultados del mecanismo de herencia simple o múltiple. En ocasiones, esas reglas permiten introducir coeficientes de verosimilitud, con lo que el Sistema será capaz de realizar evaluaciones probabilísticas para resolver conflictos en la Herencia de Propiedades. Las reglas se pueden usar, también, para solventar el problema de los cuantificadores: proporcionando un conjunto de condiciones locales que hacen que varíe el valor de una propiedad heredada. Se pueden crear, también, nuevos encuadres, nuevos atributos y nuevas relaciones taxonómicas a partir de reglas. Por ejemplo,

Si en un encuadre se produce la siguiente combinación de propiedades:

hallazgo = necrópolis

tipo morfométrico = Tipo 2

material asociado = cerámica campaniforme

Entonces «CREAR NUEVO ENCUADRE»:

Cronología = Calcolítico Tardío

Función = Enterramiento

En otras palabras, las Reglas de Producción, no representan el conocimiento procedural propiamente dicho, sino ciertas estructuras de control, especialmente la gestión de excepciones procedentes de la Herencia Múltiple de propiedades. En general, diremos que si la regla asigna un nuevo valor a un atributo, la denominaremos regla *deductiva*. Si la regla se limita a especificar las condiciones lógicas que han de satisfacer los valores de los atributos, será una regla *no deductiva*.

Las ventajas de la integración entre Sistemas a base de encuadres y Reglas de Producción son extraordinarias. En primer lugar, porque permite «corregir» el automatismo de la Herencia de Propiedades; el valor de los atributos no dependerá entonces exclusivamente de los objetos a los que está relacionado, sino también de los intereses del usuario a cada momento. En segundo lugar, las reglas pueden contener nuevo conocimiento que interesa aplicar al árbol taxonómico, con el fin de inducir en él las modificaciones que resulten de ciertos resultados concretos obtenidos al final de una sesión. Otra manera de utilizar las reglas de producción sería como puente entre dos árboles taxonómicos, el primero representando la descripción relacional de un yacimiento, el segundo representando la Teoría necesaria para interpretar ese yacimiento. La función de las reglas es calcular las relaciones taxonómicas posibles y el sentido de la Herencia de Propiedades entre uno y otro. Desarrollaremos algo más esta última posibilidad.

Imaginemos que nos interesa saber si un Dato Empírico cualquiera es interpretable por un Concepto Interpretativo bastante abstracto y situado, por tanto, en la parte superior de un árbol taxonómico distinto del que contiene la representación del dato Empírico. El objetivo que deseamos alcanzar será, por tanto:

#### OBJETO-1 *es un* OBJETO-2

Lo que hacemos, en realidad, es preguntar al sistema por el valor de las propiedades que el dato Empírico debiera de haber heredado del Concepto en el caso de estar relacionado taxonómicamente con él.

#### OBJETO-1 *tiene «atributo (x)»*

Si ese dato ha sido implementado previamente como instancia de cierto concepto ligado a su vez con el Concepto que suponemos interpreta el dato, el sistema se limitará a seguir las relaciones existentes y a registrar los resultados de la operación de búsqueda a lo largo del árbol taxonómico. Sin embargo, si no hemos implementado previamente ese dato empírico como instancia de ningún objeto en el sistema, entonces el sistema invocará un conjunto de reglas cuyo cometido será:

- crear un nuevo encuadre que represente el conocimiento disponible acerca de ese dato empírico.
- en función de ese conocimiento (descripción) calcular con cual de las unidades existentes en el árbol taxonómico puede estar relacionado. Es evidente que esta operación requiere un conocimiento espe-

cializado que no tiene por qué existir en el Sistema a base de encuadres que estemos usando. Un Sistema de Reglas, por su parte, puede actuar como módulo especializado para realizar esa tarea concreta.

Los intercambios entre los encuadres y las reglas están especificados por la Agenda interna del Motor de Inferencia. Se trata de un fichero en el que hemos descrito todos los objetivos que hay que validar a cada momento. En general, la agenda contiene las Tareas que ha de cumplir el sistema. El dominio de dichas tareas puede ser:

- un atributo (de un objeto)
- una acción procedural
- un objetivo condicional

Si la tarea (u objetivo) consiste en averiguar el valor de un atributo concreto, el Sistema usará el mecanismo de Herencia de Propiedades para calcularlo. Siempre que en la búsqueda encuentre una regla o predicado asociado a un atributo, la ejecutará en ese momento, lo que provocará, evidentemente modificaciones en los nodos del árbol taxonómico. Si el objetivo es una acción procedural (un operador), éste será ejecutado. Finalmente, en el caso de un objetivo condicional (una regla con encadenamiento hacia atrás, por ejemplo), el Motor de Inferencias analizará primero las condiciones, y si éstas se cumplen pasará a cumplir el objetivo, ya se trate del valor de un atributo o de una acción.

### Ejemplos Arqueológicos

La proliferación de programas informáticos no excesivamente caros que permiten ambos tipos de programación, no tardará en encontrar aplicación práctica en nuestra disciplina. En este capítulo examinaremos brevemente dos programas desarrollados por arqueólogos. Abordaremos en primer lugar el Sistema a base de Encuadres SILEX, programado por Joël Mourre utilizando como útil de desarrollo el programa CLASSIC, un Motor de Inferencia construido, a su vez, en una extensión orientada a objeto del lenguaje LISP (Mourre 1985).

La función de este programa es reconocer la forma (tipología) de ciertos útiles líticos: los buriles diedros. De lo que se trata es, por tanto, de proponer una ordenación de dichos buriles basándose en sus atributos extrínsecos (descripción morfométrica). El árbol taxonómico de partida aparece en la Figura 4.4.

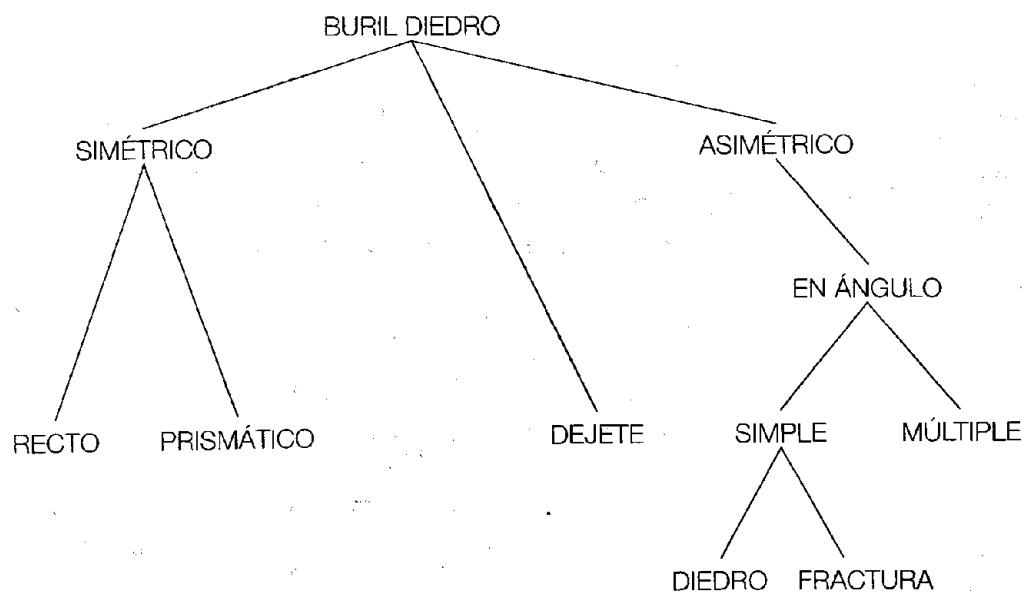


Fig. 4.4. Árbol taxonómico en el que se basa el programa silex (cf. Mourre 1985).

sos, las Teorías con las que trabajan los arqueólogos (las «Teorías de Alcance Medio») suelen ser, en el mejor de los casos, incompletas; pero para que la Herencia de Propiedades sea un mecanismo efectivo, el árbol taxonómico ha de estar completo. Esta contradicción puede resolverse integrando el sistema a base de Encuadres con una Base de Reglas de Producción.

La función de esas Reglas es «cubrir» los posibles huecos que existan en el árbol taxonómico, calculando el valor de algunas propiedades en el momento de la ejecución, o bien modificando los resultados del mecanismo de herencia simple o múltiple. En ocasiones, esas reglas permiten introducir coeficientes de verosimilitud, con lo que el Sistema será capaz de realizar evaluaciones probabilísticas para resolver conflictos en la Herencia de Propiedades. Las reglas se pueden usar, también, para solventar el problema de los cuantificadores: proporcionando un conjunto de condiciones locales que hacen que varíe el valor de una propiedad heredada. Se pueden crear, también, nuevos encuadres, nuevos atributos y nuevas relaciones taxonómicas a partir de reglas. Por ejemplo,

Si en un encuadre se produce la siguiente combinación de propiedades:

hallazgo = necrópolis

tipo morfométrico = Tipo 2

material asociado = cerámica campaniforme

Entonces «CREAR NUEVO ENCUADRE»:

Cronología = Calcolítico Tardío

Función = Enterramiento

En otras palabras, las Reglas de Producción, no representan el conocimiento procedural propiamente dicho, sino ciertas estructuras de control, especialmente la gestión de excepciones procedentes de la Herencia Múltiple de propiedades. En general, diremos que si la regla asigna un nuevo valor a un atributo, la denominaremos regla *deductiva*. Si la regla se limita a especificar las condiciones lógicas que han de satisfacer los valores de los atributos, será una regla *no deductiva*.

Las ventajas de la integración entre Sistemas a base de encuadres y Reglas de Producción son extraordinarias. En primer lugar, porque permite «corregir» el automatismo de la Herencia de Propiedades; el valor de los atributos no dependerá entonces exclusivamente de los objetos a los que está relacionado, sino también de los intereses del usuario a cada momento. En segundo lugar, las reglas pueden contener nuevo conocimiento que interesa aplicar al árbol taxonómico, con el fin de inducir en él las modificaciones que resulten de ciertos resultados concretos obtenidos al final de una sesión. Otra manera de utilizar las reglas de producción sería como puente entre dos árboles taxonómicos, el primero representando la descripción relacional de un yacimiento, el segundo representando la Teoría necesaria para interpretar ese yacimiento. La función de las reglas es calcular las relaciones taxonómicas posibles y el sentido de la Herencia de Propiedades entre uno y otro. Desarrollaremos algo más esta última posibilidad.

Imaginemos que nos interesa saber si un Dato Empírico cualquiera es interpretable por un Concepto Interpretativo bastante abstracto y situado, por tanto, en la parte superior de un árbol taxonómico distinto del que contiene la representación del dato Empírico. El objetivo que deseamos alcanzar será, por tanto:

**OBJETO-1 es un OBJETO-2**

Lo que hacemos, en realidad, es preguntar al sistema por el valor de las propiedades que el dato Empírico debiera de haber heredado del Concepto en el caso de estar relacionado taxonómicamente con él:

**OBJETO-1 tiene «atributo (x)»**

Si ese dato ha sido implementado previamente como instancia de cierto concepto ligado a su vez con el Concepto que suponemos interpreta el dato, el sistema se limitará a seguir las relaciones existentes y a registrar los resultados de la operación de búsqueda a lo largo del árbol taxonómico. Sin embargo, si no hemos implementado previamente ese dato empírico como instancia de ningún objeto en el sistema, entonces el sistema invocará un conjunto de reglas cuyo cometido será:

- crear un nuevo encuadre que represente el conocimiento disponible acerca de ese dato empírico.
- en función de ese conocimiento (descripción) calcular con cual de las unidades existentes en el árbol taxonómico puede estar relacionado. Es evidente que esta operación requiere un conocimiento espe-

cializado que no tiene por qué existir en el Sistema a base de encuadres que estemos usando. Un Sistema de Reglas, por su parte, puede actuar como módulo especializado para realizar esa tarea concreta.

Los intercambios entre los encuadres y las reglas están especificados por la Agenda interna del Motor de Inferencia. Se trata de un fichero en el que hemos descrito todos los objetivos que hay que validar a cada momento. En general, la agenda contiene las Tareas que ha de cumplir el sistema. El dominio de dichas tareas puede ser:

- un atributo (de un objeto)
- una acción procedural
- un objetivo condicional

Si la tarea (u objetivo) consiste en averiguar el valor de un atributo concreto, el Sistema usará el mecanismo de Herencia de Propiedades para calcularlo. Siempre que en la búsqueda encuentre una regla o predicado asociado a un atributo, la ejecutará en ese momento, lo que provocará, evidentemente modificaciones en los nodos del árbol taxonómico. Si el objetivo es una acción procedural (un operador), éste será ejecutado. Finalmente, en el caso de un objetivo condicional (una regla con encadenamiento hacia atrás, por ejemplo), el Motor de Inferencias analizará primero las condiciones, y si éstas se cumplen pasará a cumplir el objetivo, ya se trate del valor de un atributo o de una acción.

### Ejemplos Arqueológicos

La proliferación de programas informáticos no excesivamente caros que permiten ambos tipos de programación, no tardará en encontrar aplicación práctica en nuestra disciplina. En este capítulo examinaremos brevemente dos programas desarrollados por arqueólogos. Abordaremos en primer lugar el Sistema a base de Encuadres SILEX, programado por Joël Mourre utilizando como útil de desarrollo el programa CLASSIC, un Motor de Inferencia construido, a su vez, en una extensión orientada a objeto del lenguaje LISP (Mourre 1985).

La función de este programa es reconocer la forma (tipología) de ciertos útiles líticos: los buriles diedros. De lo que se trata es, por tanto, de proponer una ordenación de dichos buriles basándose en sus atributos extrínsecos (descripción morfométrica). El árbol taxonómico de partida aparece en la Figura 4.4.

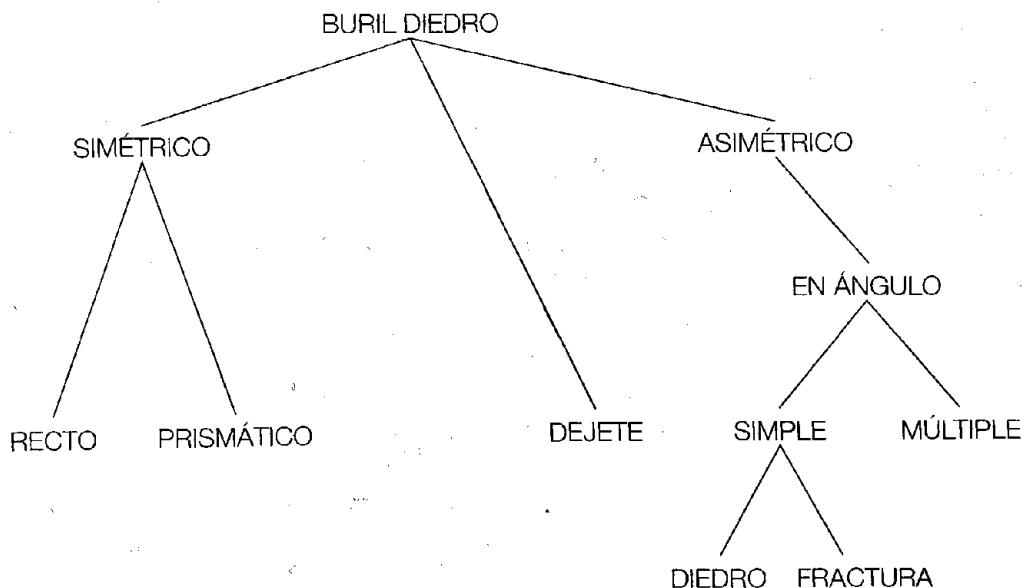


Fig. 4.4. Árbol taxonómico en el que se basa el programa silex (cf. Mourre 1985).

Los atributos del encuadre superior (Buril Diedro) son los siguientes:

ATRIBUTOS	ESTADOS (valores)
Cantidad de retoques a la izquierda	simple / múltiple
Cantidad de retoques a la derecha	simple / múltiple
Parte	Distal presente / ausente
Posición Distal	izquierda / der. / centro
Parte Proximal	presente / ausente
Posición Proximal	izquierda / der. / centro
Angulo	agudo / recto
Eje	simétrico / asimétrico
Tipo de retoque para el buril de ángulo	?
Forma	?

Cada encuadre constituye una estructuración de los datos correspondiente a un individuo típico de esa clase; constituye, por tanto, un prototipo del mismo. El encuadre BURIL DIEDRO está situado en la raíz del árbol taxonómico. Corresponde a la clase global de los buriles que se desean clasificar. Posee todos los descriptores que definirán los prototipos que descienden de él, de ahí que esos descendientes hereden los atributos de la raíz. Sólo se especifican, por tanto, los valores de aquellos atributos que diferencian unos buriles de otros. Los demás atributos se calculan en el encuadre correspondiente a la clase general y se transmiten a las distintas instancias de esa clase por medio de la herencia de propiedades. Por ejemplo:

#### ENCUADRE BURIL NORMAL

superclase: buril diedro  
 clase: buril normal  
 posición distal: centro  
 posición proximal: centro  
 ángulo: agudo  
 eje: simétrico

#### ENCUADRE BURIL RECTO

superclase: buril normal  
 clase: buril recto  
 Cantidad de retoques a la izquierda: simple  
 Cantidad de retoques a la derecha: simple  
 posición distal: centro  
 posición proximal: centro

Las reglas de producción permiten describir las nodos del árbol taxonómico que representan los distintos tipos de buriles múltiples. Tanto la premisa como el consecuente de esas reglas contiene una expresión de la forma:

[DESCRIPCIÓN](valor)]

Por ejemplo:

Si                   distal presente,  
                       posición distal izquierda,  
                       posición distal derecha,  
                       parte proximal ausente  
 entonces:          forma doble distal  
                       «BURIL MULTIPLE DOBLE DISTAL»

Todas las reglas de producción se refieren, única y exclusivamente a la clase de buriles múltiples, y sirven para asignar el valor adecuado al atributo «forma», según los descriptores que esa clase ha heredado:

**ENCUADRE BURIL MULTIPLE**

*superclase:* buril en ángulo

*clase:* buril múltiple

*cantidad de ángulos:* múltiple

*Forma:* doble-distal, doble-proximal, opuesto-derecha, opuesto-izquierda, alterno-derecha, alterno-izquierda, triple-distal-derecho, triple-distal-izquierdo, triple-proximal-derecho, triple-proximal-izquierdo, cuadruple.

El objetivo del sistema es la identificación taxonómica del buril diedro. Una sesión clásica de consulta sería la siguiente, en la que el usuario introduce los siguientes datos:

## – INTRODUCCIÓN DE DATOS:

- ? buril diedro BURIL-15
- ? Cantidad de retoques a la izquierda: simple
- ? Cantidad de retoques a la derecha: simple
- ? Parte Distal: presente
- ? Posición Distal: izquierda
- ? Parte Proximal: presente
- ? n° de diedros: múltiple
- ? Ángulo: recto
- ? Eje: asimétrico

y el ordenador responde con el siguiente resultado:

## – INTERPRETACIÓN DE DATOS:

BURIL-15 clase: buril en ángulo

BURIL-15 clase: buril múltiple

Forma: triple-proximal-izquierda

## – Descripción del objeto Buril-15:

Posición Distal: izquierda (Válido)

Parte Proximal: presente (válido)

Posición proximal: izquierda (válido), derecha (válido)

nº de diedros: múltiple (válido)

Ángulo: recto (válido)

Eje: asimétrico (válido)

Forma: triple-proximal-izquierda (válido)

A excepción de la deducción de la forma de ese buril; el resto e las inferencias (conclusión BURIL MULTIPLE) ha sido realizada explotando las relaciones taxonómicas entre los prototipos: el programa compara las propiedades en el orden que aparecen, identificando primero el tipo de Buril más general (Buril en ángulo) e investigando a continuación sus descendientes, con propiedades más particulares (Buril Múltiple). Como en la Base de Conocimiento no existe información acerca de la forma del Buril Múltiple, se utilizan ciertas reglas:

Si	parte distal presente posición distal izquierda, parte proximal presente posición proximal derecha posición proximal izquierda
Entonces	forma triple-proximal-izquierda.

que permitirán asignar ese valor al atributo «forma», sin valor declarado (y por tanto, no heredado) en el árbol taxonómico.

Otro ejemplo sencillo de integración Herencia de Propiedades/Reglas de Producción es el Sistema KIVA, programado para interpretar la función de las áreas de actividad en un yacimiento arqueológico (Patel y Stutt 1988). En este programa, los encuadres se utilizan tan sólo por su mayor riqueza expresiva a la hora de repre-

sentar el conocimiento declarativo (si los comparamos a las tripletas <Objeto><Objeto><Objeto> de los Sistemas Expertos de Primera Generación).

Esos encuadres contienen la descripción de los yacimientos arqueológicos. Los autores distinguen tres categorías:

#### HALLAZGOS:

##### - ARTEFACTOS :

hueso — instancias...  
cerámica: — instancias...  
piedra : — instancias...  
madera: — instancias...

##### - ECOFACTOS :

hueso : — instancias...  
polen: — instancias...  
madera: — instancias...

##### - YACIMIENTO:

hoyo: —instancias...  
acumulación: — instancias...  
estructuras de demarcación:  
    muro: — instancias...  
    empalizada: — instancias...  
    círculo de piedras: — instancias...

Las reglas están organizadas en varios conjuntos, cuyo objetivo es:

- singularizar las áreas de actividad, basándose en las características de cada yacimiento (situación, localización de las estructuras de demarcación)
- interpretación del uso de los artefactos
- análisis del contenido (ecofactos) de las áreas singularizadas
- sobre la base de esos contenidos, identificación de la función socio- económica del área.
- sobre la base del uso de los artefactos encontrados en cada área, identificación de la función socioeconómica del área.

Es fácil de apreciar que este sistema prácticamente no explota la Herencia de propiedades como representación del Conocimiento procedural, sino que emplea los encuadres para representar el conocimiento declarativo, tanto aquél introducido por el usuario (descripción), como aquél producido por las reglas de producción (interpretaciones). Esas interpretaciones adoptan la forma de nuevos atributos asociados a los anteriores, o bien, crean nuevos encuadres. Así, por ejemplo, un encuadre que represente una instancia concreta de artefacto, con los atributos:

Instancia de Cerámica  
base quemada  
tipo cuchara

Por medio de la regla:

Si                   (x) es un Artefacto  
                      (la base de (x) está quemada)  
                      (el tipo de (x) es «cuchara»)

Entonces: Añadir el atributo Uso, con el valor «cocina» al encuadre (x) ...

interpretaremos que ese artefacto ha sido utilizado para actividades de cocina. Si artefactos de ese tipo aparecen en un área de actividad, el área de actividad en la que hayamos encontrado el artefacto adoptará la misma función socioeconómica.

## Lenguajes Orientados a Objeto

En páginas anteriores hemos desarrollado una representación particular de las Inferencias Declarativas, basándonos en las relaciones de similaridad existentes entre dos unidades de conocimiento y la ordenación de las mismas que resulta del axioma ultramétrico, consustancial a la relación de similaridad. Hemos visto que la noción de implicación lógica puede representarse mediante el mecanismo de «Herencia de Propiedades». Ahora bien, la implicación lógica es una característica de las inferencias con un *grado de deductividad* máximo. En modo alguno podemos limitarnos a este tipo de razonamiento, por lo demás poco útil en la investigación científica debido a la necesidad que el consecuente de una deducción no aumente la información semántica contenida en las premisas. Precisamos, pues, otros tipos de programación, más sofisticados que la Herencia de Propiedades y las Relaciones Taxonómicas, para representar inferencias con distintos grados de deductividad.

Los lenguajes orientados a objeto son programas capaces de dividir funcionalmente la memoria disponible en el ordenador en paquetes independientes, con la suficiente «consciencia» de su identidad como para impedir o permitir la actividad en ellos de ciertos procedimientos. Un «objeto» es, por tanto, una parcela, debidamente singularizada, de la memoria del ordenador, que no tiene por qué corresponder a una entidad en el mundo real. Su representación computacional es bastante parecida a la de los encuadres (atributos y valores), aceptando incluso la herencia de propiedades con el fin de reducir al mínimo la cantidad de información que hay que introducir. A diferencia de los encuadres, sin embargo, esos «objetos» computables son *autosuficientes*, pues incluyen, además, una buena cantidad de conocimiento procedural, no limitado a la Herencia de Propiedades.

Ahora bien, en realidad la diferencia declarativo/procedural se mantiene, sólo que en otro nivel: dentro de esa parcela de memoria que hemos denominado «objeto» se distinguen dos componentes, que coinciden con las unidades declarativas y procedurales: el estado particular de esa memoria y aquellos métodos capaces de manipular ese estado particular, de modo que sólo los métodos propios a un objeto tienen acceso a su estado. La estructura interna de los «objetos» coincide, a grandes rasgos, con una estructuración de los conceptos basada en la existencia de un núcleo declarativo (o estado particular) y unos procedimientos de manipulación de la información contenida en ese núcleo.

El estado privado o núcleo declarativo del objeto sirve como memoria local que es compartida por todas las operaciones de manipulación que se realizan sobre él. Comprende aquella información peculiar a esa entidad, así como todos los enlaces que la unen a las demás objetos o «paquetes de memoria» existentes en el sistema. Los procedimientos de manipulación suelen denominarse métodos, y constituyen la parte dinámica que describe el comportamiento de esa entidad, es decir, representan el conjunto de operaciones que son aplicables a ese objeto en particular, así como sus reacciones ante lo que sucede en el entorno en el que se halla.

En la práctica, se utiliza un formato de representación bastante parecido al de los encuadres. El núcleo declarativo se define por medio de un conjunto de campos (llamados también variables o atributos), mientras que las operaciones o métodos aparecen detallados como campos especializados.

Un esquema típico sería:

Nombre-del-Objeto

Relacionado taxonómicamente a (Hereda propiedades de) [OBJETO]

propiedades locales:

a

b

c

métodos locales:

AñadeNuevoValor

ImprimeResultado

BorraResultado

En un lenguaje Orientado a Objeto, los objetos están ordenados por clases. Los objetos situados en una misma clase poseen operaciones comunes y, en consecuencia, un comportamiento uniforme. La clase de un objeto suele corresponderse con lo que en programación clásica se entiende por tipo de dato, esto es: carácter alfanumérico, número entero, decimal, matriz, cadena,... Los tipos suelen ser predicados que permiten definir los datos, mientras que las clases debieran considerarse como «plantillas» para la definición de los objetos. Una clase es un molde o plantilla que sirve para generar los objetos (instancias). Por ejemplo, si deseáramos progra-

mar un Sistema Experto por medio de un lenguaje Orientado Objeto, tendríamos que definir, como mínimo, las siguientes clases: «Reglas», «Hechos», «Motor de Inferencias», cada una de las cuales comprendería, a su vez, multitud de instancias.

Hemos visto que cada «objeto» tiene entidad propia (es decir, la información que contiene permanece oculta a los demás objetos; se dice que está *encapsulada*). El universo del discurso consiste, pues, en una colección de objetos independientes que se comunican entre sí intercambiándose «mensajes», los cuales activarán los métodos propios a cada objeto. De hecho, los objetos son las entidades activas pues aceptan o rechazan los mensajes o procedimientos que se pasan de unos a otros. En general el objeto acepta el mensaje que reconoce (cuyo selector activa un método existente) y rechaza los que no conoce.

La noción de mensaje es una metáfora usada en informática para referirse a una manera de programar: dadas dos estructuras de datos («objetos»), un «mensaje» es aquella función o conjunto de operadores que activa las rutinas contenidas en un objeto y genera ciertas transformaciones en él. En este caso, las rutinas incluidas en el objeto, constituyen su método, por lo que podremos diferenciar entre ambos tipos de conocimiento procedural según sean internos o externos a los objetos que definen el Sistema: varios objetos pueden compartir el mismo mensaje, pero no el mismo método, que es particular a cada objeto. En definitiva, los «mensajes» no explican cómo manipular un objeto, tan sólo dan la orden de efectuar cierto tipo de manipulación, la cual está especificada por el método residente en ese objeto. Por ejemplo, si estamos programando un sistema informático que sea capaz de dibujar una línea y un círculo, dispondremos de los siguientes componentes:

- un objeto «Línea»
- un objeto «Círculo»
- un mensaje «Dibuja»

cada objeto contiene un estado particular (el conjunto de coordenadas de puntos en la pantalla del ordenador que configuran una Línea o un Círculo) y unas rutinas de programación que permiten mostrar en la pantalla esos motivos; obviamente, el método para trazar una línea es distinto al método necesario para trazar un círculo. Finalmente, el mensaje «Dibuja» no es más que una orden, la misma en ambos casos, capaz de activar las rutinas de programación específicas a cada uno de los objetos.

La ejecución de un mensaje depende siempre del receptor. Por ese motivo, en toda invocación de una operación (un «mensaje») habrá que especificar:

- el objeto al que va destinado ese mensaje
- el nombre de la operación a realizar
- los argumentos de esa operación
- una llamada al código en el que se ha implementado esa operación.

Por ejemplo, supongamos que queremos realizar la siguiente operación aritmética:

$$2 + 3$$

ello equivaldrá a enviar el mensaje «+» al objeto «2» con el argumento «3»:

$$2 := + (3)$$

Para que el programa ejecute esa suma es preciso que en la definición del objeto «2» hayamos tenido en cuenta la implementación de los métodos «adicción» y «resultado». El anterior es un ejemplo simple, la totalidad de los lenguajes orientados objetos disponen de la clase «Entero» (y sus instancias) y el método «adicción» implementados como primitivas. Los argumentos son las variables contenidas en el objeto, o bien ciertas variables globales reconocidas por todos los objetos.

¿Qué sucede cuando se envía un mensaje a un objeto? Cada objeto tiene una parte global (compartida con otros) y una parte local, específica a él mismo. La parte global está almacenada en la clase, entidad superior con la que mantiene una relación taxonómica. La Herencia de Propiedades controla parcialmente el destino de los mensajes, haciendo que esos procedimientos sean automáticamente aplicados a las clases y a los objetos. Cuando un mensaje llega a un objeto éste busca si existe un método asociado al selector (el nombre) de ese mensaje. Si no lo encuentra puede ser debido a que ese método está definido a otro nivel, por lo que utiliza la conexión taxonómica para buscar la clase de la cual ese objeto es una instancia y busca el método en la

clase (obviamente, toda instancia hereda todo método definido a nivel de la clase a la que pertenece). Los objetos similares –los objetos situados en la misma clase– experimentarán, por consiguiente, las mismas transformaciones, puesto que reaccionan ante los mismos mensajes.

En ocasiones los objetos se envían mensajes a sí mismos. A veces puede resultar indispensable llamar a otro método definido en la misma clase, lo que equivale a remitir un mensaje al objeto que acaba de recibir uno. Podemos representar verbalmente esos mensajes como mandatos imperativos en primera persona. Dado que es la existencia de mensajes lo que da la impresión de «actividad» en los objetos, la existencia de *automensajes* permite representar entidades capaces de reaccionar por sí mismas, en ausencia de cualquier otro agente. Esta característica, que diferencia nítidamente los «objetos» de los encuadres, es fundamental para representar ciertos tipos de razonamiento científico.

En resumen, nos encontramos ante una herramienta de representación muy distinta a la expuesta hasta ahora. En los sistemas basados en relaciones taxonómicas, el conocimiento declarativo era, por definición, pasivo, mientras que el conocimiento procedural, activo, se definía de manera totalmente independiente del declarativo. Las inferencias, por tanto, se originaban en la combinación de unidades de naturaleza distinta. Los sistemas basados en el intercambio de «mensajes», por su parte, pueden ser comparados a un conjunto de chips o microprocesadores, en cada uno de los cuales se han definido las instrucciones necesarias para llevar a cabo ciertas operaciones. Cada objeto, por tanto, es una célula, un elemento de base para construir un sistema de gran complejidad. A diferencia de un Sistema Deductivo clásico, no existe un control global (sintáctico) de las inferencias: las interacciones entre los objetos (y por tanto la gestión de los conflictos derivados de esas interacciones) se definen localmente, en términos de la combinación de comportamientos individuales que proporciona el intercambio de mensajes.

Lo que más debiera interesarnos de los Lenguajes Orientados a Objeto es, precisamente, la noción de mensaje. Hemos visto que la definición de concepto coincide, a grandes rasgos, con la estructura interna de un objeto. En consecuencia, la noción de «mensaje» quizás pudiera utilizarse con provecho para representar las operaciones con conceptos, esto es, los mecanismos de la inferencia y del razonamiento científicos. Desgraciadamente, en tanto que sí abundan las investigaciones sobre emulación cognitiva en otros tipos de lenguajes informáticos (PROLOG, LISP, etc.), no son muchos los trabajos existentes, que se refieren a la representación del razonamiento mediante Lenguajes Orientados a Objeto.

El mecanismo de funcionamiento de los «mensajes» en un sistema orientado objeto concuerda perfectamente con la definición de inferencia propuesta en este libro: *la combinación de dos unidades de conocimiento para producir nuevo Conocimiento*. Consideremos que las dos unidades de conocimiento iniciales están representadas por medio de objetos; el mensaje cuyo selector se «interpreta» activará ciertos métodos implementados en los objetos de partida y creará un nuevo objeto, con el resultado de la inferencia. Por ejemplo, si el primer objeto es la representación de un *Dato Empírico* (una tumba, en la que aparece una espada) y el segundo objeto una hipótesis sociológica (características generales de las *Élites Sociales*), el resultado del mensaje será una instancia de la clase *«Élite Social»* caracterizada por la presencia de espadas. Obviamente, para que esa inferencia sea válida es preciso que el objeto *Élite Social* contenga un método capaz de utilizar la información declarativa contenida en el objeto *Tumba*, en otras palabras, para que la inferencia sea verdadera es preciso que en el sistema exista la siguiente información: «la presencia de espadas es una condición necesaria para interpretar la existencia de *Élites Sociales*». El silogismo que pretendemos representar es:

- a) En esta tumba aparece una espada
- b) la presencia de Espadas implica la existencia de una *Élite Social*

- c) El individuo enterrado en esta tumba pertenecía a una *Élite Social*

Los objetos usados son *«Tumba»* y *«Élite Social»*, no las distintas proposiciones que configuran el silogismo. El segundo objeto contiene un método constituido por una regla de producción:

Método *Interpreta*

Si	presencia de Espada
Entonces:	Crea Nuevo-Objeto
	Nuevo-Objeto hereda las propiedades de:
	«Élite Social»
	«Tumba»

A su vez, el mensaje podrá expresarse del siguiente modo:

Élite Social := Interpreta (Tumba)

Es decir, enviamos el mensaje «Interpreta» al objeto «Élite Social» con el argumento «Tumba», en términos verbales; pedimos a un Concepto Interpretativo que «interprete» un Dato Empírico. Para ello, «Élite Social» ejecuta el método que corresponde a ese mensaje, buscando la presencia de un tipo determinado de información (presencia de espadas) en el objeto que hace las veces de argumento del mensaje. En el caso de cumplirse la condición, creará un nuevo objeto (Interpretación) que heredará las propiedades del dato Empírico y del Concepto Interpretativo.

En definitiva, lo que hemos hecho es asociar el mecanismo de Herencia Múltiple de propiedades y el de la transmisión de mensajes. Un Sistema Orientado Objeto es, en realidad, una variante del Árbol Taxonómico que hemos detallado en este capítulo. La única diferencia es la posibilidad de enviar mensajes a los distintos nodos con el fin de crear automáticamente nuevos nodos y nuevas relaciones taxonómicas.

El sistema resultante adopta la apariencia de un Sistema Multi-experto, en el que cada objeto contiene un Sistema Experto propio, con una Base de Hechos y una Base de Reglas particular, la cual puede estar formada, a su vez, por objetos. Los mensajes harán las veces de estructuras de control. En estas condiciones, el conocimiento procedural estará totalmente contenido en cada objeto y será particular a ese objeto; con ello evitamos en gran medida los conflictos. Son los métodos los que representan esa parte del conocimiento. Su función es prácticamente la misma que los operadores especializados situados en el consecuente de las reglas de producción. La riqueza expresiva posible dependerá del lenguaje que utilicemos o de lo buenos programadores que scamos; un lenguaje como Smalltalk-80, por ejemplo, dispone de unas 500 funciones preprogramadas disponibles; el usuario siempre puede crearlas a la medida, si bien ello exige un considerable dominio del lenguaje de programación..

### Más allá de la «Orientación a Objeto»

El espacio distribuido de un problema está configurado por un conjunto de estados semiautónomos (elementos de procesamiento) capaces de interactuar cooperativamente y resolver el problema «compartiendo» información. En esta definición destacan dos puntos fundamentales:

- El carácter *semiautónomo* de cada uno de los elementos que contribuyen a la solución
- El grado de *cooperación* entre todos esos elementos.

Cada uno de los elementos, estados o nodos es, en realidad, un sofisticado mecanismo de resolución de problemas capaz de funcionar por sí solo; no obstante, lo que caracteriza el sistema «distribuido» es su capacidad de comunicarse con otros programas tan independientes como él mismo, y de «cooperar» en la ejecución de una tarea más compleja, modificando su conducta según el tipo de cooperación que sea necesaria.

Estas dos características justifican la expresión «Lenguajes a base de Actores», con la cual se denomina un tipo particular de lenguajes de programación, derivados de los programas orientados-objeto que han sido presentados en este mismo capítulo. Cada uno de los elementos que «cooperan en la solución del problema» reciben la categoría de *actores*, ya que son capaces de «actuar» por sí mismos. Cada actor adopta una *conducta* particular, definida en su guión. Todas las acciones a desarrollar se programan adoptando la apariencia de «mensajes» entre un actor y otro, por ese motivo, cada uno de los actores debe conocer, no solo el *nombre* de los actores con los que va a comunicarse, sino también la manera de comunicarse y los tipos de mensajes que puede enviar, o los mensajes cuya ejecución puede delegar en otro actor.

Ahora bien, ¿qué es un «mensaje», computacionalmente hablando? En realidad, se trata de una simple instrucción. Los mensajes entre actores pueden ser muy simples:

(ORDENA actor-N (SUMA 2 + 2))

en la que «actor-N» aparece como *cliente* del actor en el cual ha nacido el mensaje, o bien extraordinariamente complejos (cf. capítulo 6). Una manera sencilla de comprender el funcionamiento de un sistema como éste es recurriendo a la figura de un Sistema Experto «modular»: cada actor sería, en realidad un Sistema Experto especializado en una tarea muy concreta. Las reglas que contiene dicho Sistema Experto harían las veces de men-

sajes, cuya función no sería tan sólo ejecutar tareas específicas (SUMA 2 + 2), sino activar mensajes:

SI            condiciones a, b, c  
 ENTONCES,    Activa mensaje B

SI            mensaje B está activo  
 ENTONCES,    Activa actor-N

La implementación de cada actor tiene asociados los siguientes elementos:

- su propia base de datos de «creencias» acerca de las acciones posibles que puede llevar a cabo, sus propiedades y las de los otros agentes.
- un conjunto de meta-reglas cuya parte condición se ejecuta si su antecedente existe en la base de datos del agente.
- la planificación o estrategia general de funcionamiento de todo el sistema, implementada de manera independiente a cada actor individual y a la cual todos los actores deben amoldarse.

La comunicación de mensajes entre actores exige una arquitectura en paralelo, en la que multitud de mensajes compiten entre sí por llegar a un actor. La estructura de control que regula los conflictos que inevitablemente surgirán en ese sistema competitivo está basada en la lista de nombres de actores a los que se tiene acceso partiendo de un actor determinado. La suma de todas las listas particulares a los distintos actores del sistema define la *topología de conexiones* posible en el sistema. Dado que esas listas de direcciones son modificables (como respuesta a un mensaje específico), el sistema es capaz, en cierto modo, de «modificarse a sí mismo», recurriendo a la modificación de su estructura de control. Es precisamente este rasgo el que diferencia los lenguajes orientados-objeto de los basados en Actores: en los primeros la conexión entre objetos (enlaces taxonómicos) son extremadamente simples y de bajo nivel, mientras que en los segundos, los mensajes cruzados entre actores y la simulación con el modo de razonamiento humano, es mucho más directa.

Propiamente hablando, los lenguajes a base de actores no deben clasificarse como técnicas de inteligencia artificial. Ahora bien, sus propiedades en tanto que sofisticados lenguajes de programación los convierten en el mecanismo idóneo para representar la *resolución distribuida de problemas*. Se trata de problemas cuya resolución sería imposible secuencialmente, esto es, disponiendo los estados intermedios uno después de otro. Si el número de estados en el espacio del problema fuese extraordinariamente grande, la disposición secuencial de los mismos en un programa declarativo no sería efectiva, porque exigiría demasiado tiempo de computación; en ése caso sería mucho mejor dividir las tareas en módulos (actores), y coordinando las distintas tareas por medio de mensajes, de forma que el conjunto de esas tareas ejecutadas parcialmente en paralelo, nos condujese a la solución en menos tiempo.

Otra aplicación posible concierne a la teoría de la simulación. En el capítulo 6 se presenta un ejemplo de aplicación de los Lenguajes a base de Actores en Arqueología: el proyecto EOS dirigido por James Doran.

### Comentarios bibliográficos

Para encontrar información adicional acerca de la Deducción Automática, el lector deberá remitirse a las publicaciones fundamentales de Johnson-Laird y Byrne (1991) y la de Bibel (1993), o bien a las publicaciones «informáticas», en las que los ejemplos son particularmente banales («Clyde es un Elefante», «Tweedy es un pájaro») y la notación lógica bastante complicada. Con todo, cabe citar Touretzky (1986), Shastri (1988), y el libro compilado por Lenzerini, Nardi y Simi (1991). Los libros de introducción a la Inteligencia Artificial (Capítulo 2) y los dedicados a la Representación del Conocimiento suelen incluir secciones acerca de la herencia de propiedades, por lo que el lector puede empezar por ellos, antes de pasar a la bibliografía especializada. La formalización del «razonamiento taxonómico» que aparece en este capítulo se basa en la de Harty (1991), Harty, Thomason y Touretzky (1990). La integración entre Sistemas Expertos, Relaciones Taxonómicas y «encuadres» suele estar bien expuesta en la documentación específica a los programas especializados en tales tareas (cf. Anexo). Libros que tratan exclusivamente este problema, describiendo numerosos programas comerciales, son: Tello (1989), Payne y McArthur (1990). Acerca de los lenguajes de actores, resulta fundamental el libro de Agha (1986), y el de Hewitt y Agha (1990). Tello (1989, capítulo 9) puede servir como introducción sencilla.

## 5. LA LÓGICA DEL DESCUBRIMIENTO CIENTÍFICO

### Observación y Descubrimiento Científico

El objetivo de este capítulo es presentar una serie de técnicas informáticas capaces de descubrir la regularidad que pudiera haber en la base de datos que contiene los resultados de una excavación arqueológica:

«REGULARIDAD: una limitación en el espacio de todas las situaciones posibles; existe ‘regularidad’ si algún hecho es imposible o menos probable que otro» (Zytkow y Baker 1991: 34).

Habitualmente, esta tarea ha sido encomendada a la Estadística (Carnap 1950, Carnap y Jeffrey 1971), definiéndose un principio general a partir de las observaciones iniciales (DATOS), y expresándolo en términos de la covariación detectada entre las propiedades que las describen. En Arqueología, D. Clarke (1978) fue uno de los principales defensores de esta manera de abordar las inferencias inductivas. Propuso utilizar como mecanismo de descubrimiento la correlación estadística: los tipos arqueológicos, culturas, tecnocomplejos, etc.,... se definirían en tanto que «paquetes» de atributos o propiedades estrechamente correlacionados. Así, por ejemplo, un artefacto lítico pertenece al *tipo Punta de La Gravette*, si los rasgos morfométricos que definen ese artefacto están correlacionados estadísticamente de igual modo que en la definición del *tipo Punta de La Gravette*. A su vez, este tipo de artefacto se definía por medio de la comparación de todas las puntas morfológicamente semejantes halladas en un lugar concreto (La Gravette), calculando a continuación la correlación de sus rasgos morfométricos y usando el valor de esa correlación como regla discriminante del tipo en cuestión.

En este capítulo los operadores estadísticos son substituidos por programas informáticos que han sido utilizados con éxito en otras disciplinas, tales como medicina y geología. Estos programas suelen agruparse bajo el epígrafe «Aprendizaje Automático» ó *Machine Learning*, y aunque utilizan cálculos estadísticos como fondo, pretenden imitar la manera en que los seres humanos aprenden conceptos a partir de ejemplos: dado un espacio que contiene *todos* los hechos posibles (una Base de Datos con todas las combinaciones posibles de los atributos que describen esos ejemplos), se extrae el hecho o los hechos que sean *más* posibles que los demás, utilizando para esa extracción criterios heurísticos (por ejemplo, el más «probable», matemáticamente hablando).

El mecanismo de funcionamiento de estos programas «de inducción automática» puede formalizarse del siguiente modo: Sea  $U$  el conjunto universal, es decir, todos los artefactos que podemos encontrarnos en el transcurso de una excavación; por ejemplo, todos los útiles líticos que pudiera haber en una cueva ocupada por cazadores paleolíticos. No hay limitación alguna para la cantidad de elementos en  $U$ . Un principio general  $G$ , -buriles del Magdalenense final- puede describirse, entonces, en tanto que subconjunto de  $U$ :

$$G \supset U$$

«Descubrir  $G$  significa poder reconocer los objetos que aparecen en  $G$ ; esto es, aquellos útiles líticos que sean buriles y que puedan datarse en el Magdalenense final. Una vez que  $G$  haya sido descubierto, el sistema será capaz de reconocer si un objeto cualquiera  $X$  pertenece o no a  $G$ . Obviamente, si la base de datos no contiene «ejemplos» de esa categoría, el programa será incapaz de obtener un resultado interpretable en términos arqueológicos.

Así pues, el propósito de un descubrimiento científico es poder llegar a describir una entidad general, partiendo de lo que sospechamos son instancias de esa entidad. Un *ejemplo* para aprender la entidad  $G$  es un par (objeto, clase)

donde *objeto* es la descripción de un artefacto arqueológico y *clase* un operador que puede tomar dos valores: «+» ó «-». Si *objeto* pertenece a *G*, entonces *clase* = + ; de otro modo *clase* = - . En el caso de los artefactos líticos citado al principio, el par

(CV12569, buril)

donde «CV12569» es el número de inventario de un objeto y «buril» una adscripción tipológica. Al haber definido *G* en tanto que «buril», diremos que el par es *un ejemplo positivo*; de otro modo -si la adscripción tipológica fuese «raedera», diríamos que el par es *un ejemplo negativo* del principio *G*.

Dado un conjunto *E* de ejemplos positivos y negativos, constituido por aquellas piezas que sabemos con total certidumbre que son o no son buriles y que datan o no datan en el Magdaleniense final, debemos encontrar una fórmula *F*, expresada en un lenguaje descriptivo específico, tal que:

para todos los objetos *X*

- (1) Si *X* es un ejemplo positivo en *E*, entonces *X* coincide con *F*.
- (2) Si *x* es un ejemplo negativo en *E*, entonces *X* no coincide con *F*.

como resultado, *F* constituirá la manera en que el sistema «comprende» la entidad *G*. Esta fórmula constituye el equivalente *simbólico* de los discriminadores estadísticos. La calificamos de «simbólica» porque no está basada en la covarianza o correlación de atributos, sino en la *co-ocurrencia* de los mismos. Lo habitual es representarla mediante *reglas de producción* del tipo siguiente:

Si                el objeto **X** tiene las propiedades **a, b, c, d, e**  
 Entonces        el objeto **X** es una instancia del TIPO **Y**.

Utilizaremos este formato -relacionado estrechamente con los Sistemas Expertos- para expresar los resultados que obtengan los programas de descubrimiento automático. Como veremos, esos programas actúan sobre una base de datos, buscando el esquema de asociaciones y co-ocurrencias entre los atributos que la definen. Precisamente por ello son potencialmente muy útiles en arqueología: ponen de relieve o «descubren» asociaciones entre los artefactos y las variables que describen su morfología o su forma de deposición. Su empleo será tanto más recomendable cuanto mayor sea la base de datos.

## Mecanismos fundamentales de descubrimiento

Desarrollemos el esquema formal adelantado en la sección anterior: si alguna entidad pertenece al conjunto *G* y tiene la propiedad *P*, entonces, todas las entidades que pertenezcan a *G* manifestarán la misma propiedad *P*. Por ejemplo, a partir de las siguientes evidencias arqueológicas:

- En el yacimiento *Y* se encontraron huesos de reno.  
 La función económica del yacimiento *Y* (establecida mediante el examen de la industria lítica) es «caza mayor»
- En el yacimiento *Z* se encontraron huesos de reno.  
 La función económica del yacimiento *Z* (establecida mediante el examen de la industria lítica) es «caza mayor»

En este ejemplo tenemos dos entidades (los yacimientos *Y, Z*) que pertenecen al conjunto *G* (yacimientos en los que se practicaba la Caza Mayor), y ambos comparten una misma propiedad *P* (hallazgo de huesos de reno). *Descubrir* significa, en este contexto, *generalizar* la propiedad *P*, para lo cual afirmaremos lo siguiente: «en todos los yacimientos donde se practicó la Caza Mayor aparecen huesos de reno». Expresándolo mediante una regla de producción:

Si                *X* es un yacimiento con huesos de reno  
 Entonces        FUNCIÓN ECONÓMICA (*x*) = Caza Mayor

Es fácil observar que nuestras evidencias arqueológicas de partida (Yacimiento *Y*, Yacimiento *Z*) eran dos ejemplos positivos de un mismo término general «Función Económica = Caza Mayor». Lo único que hemos hecho ha sido proponer una descripción de ese mismo término general que cubre todos los ejemplos positivos y

ninguno de los posibles ejemplos negativos. Este tipo de generalización suele aparecer designado como *generalización por suma de variables*, ya que la nueva variable «X» sustituye a las constantes «yacimiento Y», «yacimiento Z» que aparecían en las instancias.

El ejemplo puede parecer trivial, y realmente lo es; no obstante, el lector puede imaginar fácilmente la apariencia que adoptaría un caso real: una base de datos que contendría las tipologías de útiles líticos encontrados en cada yacimiento y los restos de fauna asociados a esos utensilios. Dada una hipótesis inicial (FUNCIÓN ECONÓMICA), el programa de descubrimiento debe encontrar los atributos asociados con cada uno de los valores que pueda adoptar esa hipótesis. La definición de esa hipótesis inicial es el momento más delicado de todo el proceso: la hipótesis ha de aparecer como un atributo más, de modo que el programa calcule todas las asociaciones posibles. En algunos casos (la CRONOLOGÍA, por ejemplo) esto es relativamente sencillo, pues siempre podemos incluir en la base de datos una nueva variable con los resultados de las dataciones absolutas o relativas que se han llevado a cabo en los distintos yacimientos. En otros casos (la FUNCIÓN ECONÓMICA, sin ir más lejos), la adscripción de los valores de esta nueva variable es mucho más discutible, ya que no existe, aparentemente, una definición universal de la misma. Una solución aceptable sería utilizar los resultados de una inducción previa, en la que los ejemplos positivos y negativos fuesen sociedades conocidas etnológicamente y en las que se asociaría su modo de subsistencia paradigmático con el utilaje lítico abandonado en sus asentamientos. En el Capítulo 6 volveremos a plantear esta importante cuestión, lo que nos interesa ahora es describir la manera en que, para definir las asociaciones y co-ocurrencias que pudiera haber en una base de datos, el ordenador tiene que basarse, necesariamente, en ejemplos positivos y negativos del concepto que se quiere «descubrir».

T. Mitchell (1977, 1982) propuso que el algoritmo de generalización se describiese en términos de la búsqueda heurística en un espacio abstracto que contuviese reglas potenciales: a partir de una serie de ejemplos positivos y negativos, el algoritmo impone un proceso secuencial, generando, refinando y rechazando hipótesis sucesivas que le irán acercando paulatinamente al concepto o regla que se quiere obtener. Cada hipótesis representa una aproximación al concepto final, y es el resultado del procesamiento de una cantidad determinada de ejemplos. A medida que aumenta el número de ejemplos, la hipótesis experimenta nuevas modificaciones. El algoritmo puede describirse del siguiente modo (véase Bratko 1989: 271):

Para aprender un concepto *C* a partir de una serie dada de ejemplos  $E_1, E_2 \dots E_n$  (en donde  $E_i$  ha de ser, necesariamente, un ejemplo positivo de *C*)

1. adoptar  $E_1$  como la hipótesis inicial,  $H_1$ .
2. procesar los restantes ejemplos

Para cada  $E_i (i = 2, 3, \dots)$  hacer:

- 2.1 Comparar la hipótesis actual  $H$  con  $E_i$ ; el resultado de la comparación es una descripción,  $D$ , de las diferencias entre  $H$  y  $E$ .
- 2.2 Modificar  $H$  de acuerdo con  $D$ , según sea  $E_i$  un ejemplo positivo o negativo de *C*. El resultado de esa modificación es una Hipótesis refinada  $H_j$ , acerca de *C*.

Consideremos el siguiente ejemplo: en un área geográfica bien delimitada se conocen cinco asentamientos prehistóricos con una única fase de ocupación. Además de su cronología, la información que disponemos de cada yacimiento (P1, P2, P3, N1, N2) es la siguiente:

P1 – 2 cabañas

- cabaña A: 2 hogares, 2 silos, presencia de restos de fauna
- cabaña B: 1 hogar, 1 silo, ausencia de restos de fauna

P2 – 2 cabañas

- cabaña C: 2 hogares, 1 silo, ausencia de restos de fauna
- cabaña D: 1 hogar, 2 silos, presencia de restos de fauna

N1 – 2 cabañas

- cabaña E: 2 hogares, 1 silo, presencia de restos de fauna
- cabaña F: 1 hogar, 1 silo, presencia de restos de fauna

N2 – 2 cabañas

- cabaña G: 1 hogar, 2 silos, ausencia de restos de fauna
- cabaña H: 2 hogares, 1 silo, presencia de restos de fauna

P3 – 2 cabañas

- cabaña I: 1 hogar, 2 silos, ausencia de restos de fauna
- cabaña J: 1 hogar, 1 silo, ausencia de restos de fauna

En la argumentación que sigue, las características de cada yacimiento se expresarán del modo siguiente:

$$P1 = \{[2, 2, \text{presencia}] [1, 1, \text{ausencia}]\}$$

en donde cada tripleta debe leerse en el siguiente orden [ $n^{\circ}$  hogares,  $n^{\circ}$  silos, presencia/ausencia de restos de faunal]; es importante tener presente que el orden de las dos tripletas es indiferente. Nuestro objetivo es calcular una fórmula general para adscribir una cronología a futuros yacimientos que puedan descubrirse en ese área. Consideramos probada la cronología de los asentamientos P1, P2, P3 en una fase Mesolítica; los yacimientos N1 y N2, por el contrario, son claramente posteriores. Tal y como indica el algoritmo, seleccionamos el primer ejemplo positivo y definimos una hipótesis inicial en sus mismos términos:

$$\text{MESOLÍTICO} = \{[2, 2, \text{presencia}] [1, 1, \text{ausencia}]\}$$

A continuación comparamos esta hipótesis con el segundo ejemplo (*P2*). Como se trata de un ejemplo positivo, intentaremos extraer los rasgos comunes. Tenemos varias opciones, según sea el orden en el que comparemos cada una de las dos cabañas (Fig. 5.1.).

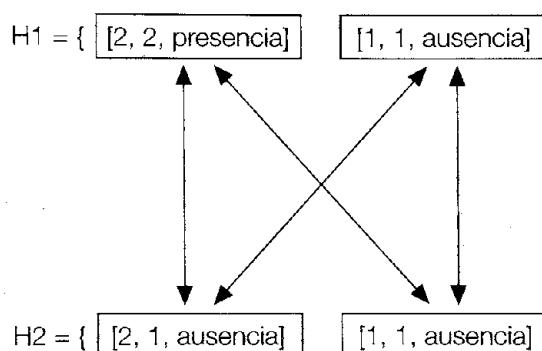


Fig. 5.1. Análisis de las asociaciones y correspondencias entre una Hipótesis y un Ejemplo Positivo.

Esta operación genera cuatro modificaciones posibles:

- H2a = (2, ?, ?)(1, ?, ?)
- H2b = (?, ?, ?)(?, 1, ausencia)
- H2c = (2, ?, ?)(?, ?, ?)
- H2d = (?, 1, ausencia)(1, 1, ausencia)

en donde el signo «?» indica que la característica en cuestión es irrelevante, por ser distinta en dos ejemplos positivos de un mismo concepto. Cada una de estas cuatro hipótesis debe ser corregida con ayuda del siguiente ejemplo; en este caso N1 es un ejemplo negativo (no es datable en el Mesolítico), por lo que eliminaremos todas las posibles coincidencias. H2a coincide con N1 (en ambos casos aparece la especificación dos hogares en una cabaña y uno en la otra), por lo que la eliminaremos como hipótesis candidata; el criterio a seguir, por tanto, es el siguiente: de las dos cabañas del asentamiento, una ha de tener dos hogares y la otra uno; no se especifica cuál de las dos cabañas es la que tiene dos hogares. De la confrontación entre N1, H2b y H2d se infiere que la única modificación posible es:

$$H3 = \{[?, ?, ?][?, 1, ausencia]\}$$

Como los ejemplos siguientes P3 y N2 no modificarían esa afirmación, H3 será la hipótesis que retengamos como «generalización cronológica» de la evidencia disponible en esa región. En términos arqueológicos, pues, diremos que un asentamiento de la región en cuestión podrá ser datado en el Mesolítico, cuando se registre una deposición diferencial de los restos de fauna en las distintas unidades de habitación, esto es, cuando en una de las cabañas, habiendo un único silo, falten los restos de fauna; los restantes atributos son, cronológicamente, irrelevantes. Aquellos yacimientos que no se ajusten a esa fórmula tanto pueden fecharse en el mesolítico como en otra fase.

No cabe duda que este ejemplo está demasiado simplificado, pero ayuda a comprender bastante bien el mecanismo de búsqueda secuencial. Contribuye, además, a poner de manifiesto lo que pudiera ser una dificultad mayor del algoritmo de Mitchell: la potencial explosión combinatoria cuando el número de características seleccionadas es demasiado grande. Hemos visto que el algoritmo requiere la enumeración de todas las combinaciones de rasgos descriptivos posibles, eliminando a continuación todas aquellas que no coincidan con los ejemplos positivos. Las combinaciones remanentes después de aplicar el procedimiento de eliminación han de ser generalizaciones correctas del conjunto de ejemplos positivos. El problema radica en que el conjunto de generalizaciones posibles es demasiado grande como para poder ser enumeradas una por una.

Mitchell resolvió esa dificultad explotando a fondo la relación de orden parcial que existe entre hipótesis generales e hipótesis específicas. Diremos que si una hipótesis *Ha* abarca más ejemplos positivos que otra *Hb*, entonces *Ha* es *más general* que *Hb*, en tanto que *Hb* es *más específica* que *Ha*. En nuestro ejemplo anterior, *H2d* es más específica que *H2c*. El operador que hemos utilizado para calificar de este modo a las dos hipótesis es el número de rasgos descriptivos que contienen: *H2d* contiene cinco características necesarias, en tanto que *H2c* precisa tan sólo de una. Resulta fácil observar que las hipótesis más específicas imaginables coinciden con las evidencias, los ejemplos de partida, pues todas las características aparecen especificadas; por el contrario, la hipótesis más general sería aquella en la que ninguna de las características estuviese especificada.

El algoritmo explota a fondo esta noción de generalidad/especificidad para evitar incurrir en la explosión combinatoria del espacio de hipótesis. Dos son los esquemas básicos de búsqueda heurística:

- (a) empieza con la hipótesis más general posible (aquella que sea compatible con *todos* los ejemplos positivos y negativos a nuestra disposición), la cual será modificada paulatinamente, especificándola con ayuda de los ejemplos positivos.
- (b) empieza con la hipótesis más específica (uno de los ejemplos positivos), generalizando a partir de ella a medida que se procesan nuevos ejemplos, esto es, se eliminan progresivamente los atributos que individualizan los ejemplos positivos y se retienen aquellos que aparecen en todos los ejemplos positivos y en ninguno de los negativos. En el caso antes comentado de los yacimientos mesolíticos se ha utilizado un esquema específico-general como éste.

Aunque la dirección de la búsqueda es distinta, ambos métodos son bastante similares. En la implementación del algoritmo publicada por Mitchell (1982) ambos mecanismos están combinados. El programa retiene dos conjuntos de hipótesis -el conjunto más específico de descripciones compatibles con los datos (**E**), y el conjunto más general de tales descripciones (**G**). Cuando se procesa un ejemplo positivo que no coincide con las hipótesis en **E**, el programa opera siguiendo el primer mecanismo para aumentar la generalidad de los elementos de **E**. Igualmente, cuando el ejemplo es negativo, el segundo mecanismo entra en acción, permitiendo una especificación de los elementos en **G**. La función de estos dos conjuntos es controlar la explosión combinatoria que pudiera darse en cada uno de ellos: el conjunto **E** resume todos los ejemplos positivos y se usa para eliminar los miembros de **G** que sean demasiado específicos; del mismo modo, el conjunto **G** resume los datos negativos, y lo usaremos para controlar los miembros excesivamente generales de **S** (la base de datos). En esta versión particular del algoritmo, la operación de descubrimiento habrá concluido cuando los conjuntos **E** y **G** coincidan.

### Generalización de Descripciones

La más conocida de las variantes del algoritmo de Mitchell es el programa AQ de Michalski (cf. Michalski 1980, 1983). AQ<sup>n</sup> (en donde *n* indica las sucesivas versiones del programa, la última publicada es AQ<sup>15</sup>) es un programa de descubrimiento o inducción multipropósito, cuyo objetivo consiste en la formulación de reglas generales que describan diversas clases de ejemplos. Los elementos de entrada (*input*) consisten en vectores

de valores de atributos; incluye también conocimiento de fondo acerca del ámbito o tema en el que se va aplicar el programa. El resultado del programa son expresiones del tipo

CONDICIÓN → CLASE

donde CONDICIÓN puede ser una conjunción o disyunción de conjunciones, de manera que describa todas las entidades asignadas a la CLASE. Una versión simplificada del algoritmo es la siguiente:

1. Seleccionar al azar un ejemplo positivo (llamado GERMEN)
2. Comparando el GERMEN con el primer ejemplo negativo, generar todas las hipótesis máximamente generales que cubren el GERMEN y excluyen el ejemplo negativo.
3. Especializar la hipótesis para excluir todos los ejemplos negativos. Esto puede llevarse a cabo considerando un ejemplo negativo después de otro, y añadiendo, cuando sea necesario, requisitos adicionales a la hipótesis. Después de cada uno de los pasos de especialización, las hipótesis recién generadas se ordenan según su potencia clasificadora y según los otros aspectos definidos en el criterio de preferencia. Sólo se conserva la hipótesis más prometedora. Las hipótesis obtenidas al final del procedimiento de especialización configuran un conjunto denominado ESTRELLA («star», en la terminología de Michalski).
4. Elegir de entre todas las hipótesis en la ESTRELLA aquella que esté mejor ordenada. Si esa hipótesis cubre todos los ejemplos positivos, acabar la sesión (se ha obtenido una solución al problema). De otro modo, encontrar los ejemplos positivos que siguen sin ser compatibles con la hipótesis.
5. Repetir los pasos 1-4 para el conjunto remanente. Continuar hasta que todos los ejemplos positivos sean compatibles con la hipótesis. La disyunción de hipótesis elegida al final de cada ciclo es una descripción consistente y completa de todos los ejemplos positivos y maximiza el criterio de preferencia.

Por consiguiente, el programa construye una descripción disyuntiva de un concepto, ley o expresión general cuando la descripción conjuntiva no es posible. Los conjuntos individuales pueden diferir significativamente según sea el tamaño del conjunto de ejemplos. Este resultado da pie a una interesante interpretación: el conjunto compatible con la mayoría de ejemplos puede considerarse como una caracterización de los miembros típicos o «ideales», mientras que los conjuntos compatibles con pocos ejemplos caracterizarán los casos excepcionales.

AQ<sup>n</sup> empieza con reglas máximamente generales, especificándolas paulatinamente; añade a las reglas nuevos términos para excluir ejemplos negativos, manteniendo la compatibilidad con los ejemplos positivos. El sucesor de AQ<sup>n</sup>, el programa INDUCE 1.2 (Dietterich y Michalski 1981), es su opuesto. Empieza con una descripción muy específica y la mantiene durante la generalización. Los elementos de entrada (el *input*) ya no está limitado a pares (ATRIBUTO, VALOR), sino que incluye cuantificadores, funciones y predicados relativos con más de un argumento. El programa explora el espacio de las descripciones siguiendo el método llamado Búsqueda Radial («beam search»). El mecanismo general puede esquematizarse en los siguientes pasos:

1. Crear el conjunto  $H$  de manera que contenga un subconjunto aleatorio de elementos de entrada con tamaño  $W$
2. Generalizar lo menos posible cada una de las descripciones en  $H$
3. Eliminar las descripciones implausibles, reteniendo tan sólo las mejores, es decir, aquellas que sean simples y compatibles con muchos ejemplos positivos; las peores son las descripciones complejas y compatibles con pocos ejemplos
4. Considerar como resultado aquella descripción en  $H$  compatible con un número suficiente de ejemplos. Si  $H$  está vacío, o bien se ha obtenido ya un número suficiente de reglas, acabar la sesión; de otro modo, continuar desde el paso 2.

El más avanzado de los programas de la familia AQ es, posiblemente, CLUSTER/2 (Michalski y Stepp 1983a, 1983b, Stepp y Michalski 1986). Este programa construye un conjunto de descripciones abstractas que organizan los datos iniciales en particiones disyuntivas. La estructura del algoritmo es la siguiente:

1. Dado: a) una serie de datos iniciales (*individuos*)
  - b) un número  $k$  de grupos hipotéticos
2. Elegir al azar  $k$  individuos

3. Construir distintos grupos basándose en las diferencias observadas entre los individuos elegidos.
4. Disponer los grupos obtenidos en el paso anterior en grupos disyuntivos.
5. Evaluar la agrupación usando el criterio LEF
6. Si la agrupación no es satisfactoria, elegir de nuevo  $k$  individuos, pero ahora eligiéndolos entre aquellos situados en los márgenes de los grupos anteriores, esto es, no relacionados con la agrupación que no ha pasado el criterio de evaluación. Volver al paso 3.
7. Si la agrupación es satisfactoria, FIN de los cálculos.

El criterio LEF, utilizado para evaluar la «calidad» de la agrupación, y consiguientemente, la naturaleza de los términos inducidos, es una función parametrizable que tiene en cuenta las siguientes propiedades:

- el ajuste entre los grupos y los datos
- la simplicidad de la descripción de los grupos
- el grado de diferencia entre los distintos grupos
- la cantidad de parámetros en cada grupo
- la dimensionalidad de la agrupación, esto es, la cantidad de información necesaria (nº de variables) para situar cada individuo en cada uno de los grupos.

El programa utiliza estos parámetros para producir un valor numérico que refleje la «calidad» de una agrupación final, de modo que las distintas agrupaciones posibles puedan compararse y elegir la «mejor» de entre ellas.

Ninguno de los programas de la familia AQ ha alcanzado difusión comercial, probablemente por ese motivo no han sido nunca aplicados en arqueología. Aunque basado en un algoritmo diferente, podemos citar a título de ejemplo el proyecto COCLUSH, actualmente en fase de experimentación, debido a O. Grey en colaboración con el Museo del Louvre (Grey 1991). Aunque se trata de un experimento realizado con hachas de bronce, sus líneas maestras pueden reproducirse en cualquier ensayo tipológico. La primera etapa del proyecto consiste en comprobar las tipologías existentes acerca de las hachas de la Edad del Bronce (1800-700 a.C.), para validarlas o rechazarlas. Se ha analizado un centenar de hachas atribuibles a todos los períodos de la Edad del Bronce. Las informaciones asociadas a cada objeto se refieren a su decoración, morfología, composición metálica, etc. Esas informaciones están representadas por medio de una red semántica como la que aparece en la Figura 6.2. Los distintos nodos que aparecen en el grafo (términos en el interior de rectángulos) representan los conceptos; los arcos (términos en el interior de una elipse) configuran las conexiones o enlaces posibles entre conceptos. Este grafo presenta la propiedad adicional de que cada secuencia <concepto><relación><concepto> sólo aparece una vez, de modo que podemos disecarlo sin ambigüedades en triplets:

HACHA N° 1	POSEE	CUERPO
cuerpo	posee	superficie_plana
extremo	misma_altura	bordes
.../...		

Los objetos a clasificar no son los únicos conocimientos contenidos en COCLUSH. Por medio de axiomas y taxonomías podemos precisar lo que sabemos acerca del dominio de aplicación («Hachas de la Edad del Bronce»). Las taxonomías sirven para ordenar series de conceptos, especificando, por ejemplo, que todo borde, superficie o extremo de un hacha son elementos constitutivos de esa pieza. Los axiomas expresan relaciones conocidas entre conceptos, y están representados por medio de las reglas de producción que ya nos son familiares:

SI (bordes altura ?x) Y (extremo altura ?x)  
 ENTONCES (bordes misma\_altura extremo)

SI (filo estado usado)  
 ENTONCES (objeto es\_un útil)

SI (filo estado nuevo) Y (perforación funcional no)  
 ENTONCES (objeto es\_un sistema\_premonetario)

COCLUSH contiene un centenar de grafos como el anterior. El objetivo es calcular agrupaciones de ob-

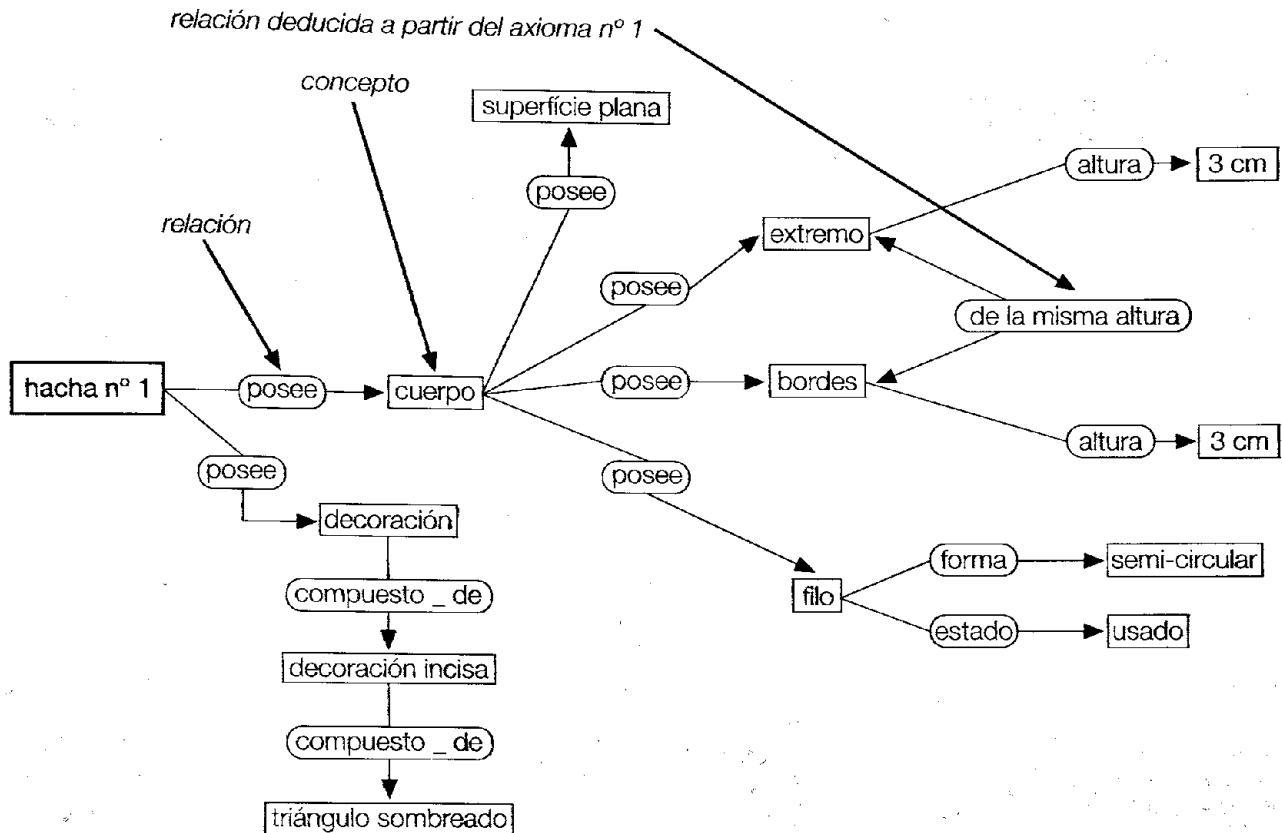


Fig. 5.2. Red Semántica utilizada por el programa COCLUSH para representar el conocimiento inicial (Grey 1991).

jetos similares y generalizar expresiones a partir de esas agrupaciones. Las agrupaciones de objetos se representan por medio de expresiones como la siguiente:

(hacha<sub>x</sub> RELACIÓN hacha<sub>y</sub>)

en donde el término RELACIÓN hace referencia a tres criterios de clasificación distintos:

- Relación «Espacio-Temporal»
- Relación «Tecnológica»
- Relación «Funcional»

con lo que obtenemos tres clasificaciones distintas: una en la que los agregados de objetos contienen hachas con la misma cronología, otra con hachas fabricadas del mismo modo, y una tercera con hachas utilizadas para la misma actividad. En cada uno de los tres casos, el objetivo es calcular la intersección y/o co-ocurrencia de descripciones entre hachas relacionadas. En otras palabras, partiendo de diversas informaciones acerca de la morfología de los útiles y de las relaciones entre objetos distintos, el programa debe calcular una definición general que sea adecuada a todos los objetos que estén relacionados cronológicamente, tecnológicamente o funcionalmente. El programa utiliza para ello los siguientes operadores de generalización :

- transformación de constantes en variables. Por ejemplo, la expresión (BORDES PRESENCIA ?) es más general que (BORDES PRESENCIA ANILLOS). El signo «?» representa una variable libre.

- *eliminación de condiciones.* La afirmación:  
(CUERPO PRESENCIA BORDES) & (CUERPO PRESENCIA SUPERFICIE)  
puede convertirse en la más general:  
(CUERPO PRESENCIA BORDES)
- *búsqueda de conexiones taxonómicas entre conceptos.* Utilizando las taxonomías que agrupan conceptos y que representan el conocimiento que tenemos sobre el dominio de aplicación podemos generalizar (CUERPO PRESENCIA SUPERFICIE) en (CUERPO PRESENCIA ELEMENTO\_ESTRUCTURAL).

Como resultado de la acción del programa, el usuario obtiene una definición compacta de todas las hachas relacionadas. Por ejemplo, todas las hachas con la cronología «Siglo IX a.C.» se caracterizan por:

- poseer un filo con forma semicircular
- poseer una decoración incisa compuesta de triángulos sombreados
- constituir un sistema premonetario.

Estas características comunes han sido calculadas por el programa a partir de una comparación de todos los grafos correspondientes a las hachas de un mismo tipo cronológico. El grafo resultante de esa comparación ha sido simplificado utilizando alguno de los operadores de generalización anteriores, y al mismo tiempo se han ejecutado las reglas «axiomáticas», que añaden nuevos caracteres no morfométricos a la generalización resultante. Llamamos *generalización* al resultado porque, obviamente, ha sido obtenida por agregación de definiciones particulares, y simplificación a posteriori del resultado de esa agregación.

### **Inducción de árboles de decisión**

Un método automático de descubrimiento algo diferente a los anteriores es el conocido como ID3 (*Interactive Dichotomizer 3*) (Quinlan 1986). Su continuador se denomina C.4.5). Aunque limitado, es bastante más rápido que los que acabamos de ver, y se adapta muy bien a las limitaciones de memoria y rapidez de proceso impuestas por los microordenadores actuales, por lo que es, de largo, el sistema más popular y utilizado.

Para empezar necesitamos un conjunto de ejemplos positivos y negativos de algún concepto, y los representaremos por medio de una lista de pares atributo-valor. Por ejemplo, consideremos los datos de la tabla siguiente, referidos a ciertas características de unas cerámicas:

Tamaño	Decoración Pintada	Decoración Incisa	Cronología
grande	presente	presente	tardía
grande	presente	ausente	ardía
grande	ausente	presente	tardía
grande	ausente	ausente	media
medio	presente	presente	tardía
medio	ausente	presente	media
media	ausente	ausente	media
pequeño	presente	presente	antigua
pequeño	ausente	presente	antigua
pequeño	ausente	ausente	antigua

El objetivo es obtener una fórmula distintiva de cada periodo cronológico, de manera que podamos usarla para distinguir la cronología de los objetos a partir de su tamaño y decoración.

El programa funciona del modo siguiente:

1. elegir al azar un subconjunto de tamaño  $V$  en el conjunto de ejemplos a nuestra disposición (ese subconjunto recibe el nombre de «ventana»)
2. aplicar el algoritmo CLS (explicado más adelante) para generar una generalización a partir de la ventana elegida

3. analizar el resto de la base de datos para encontrar excepciones a esa regla
4. si se encuentran excepciones, se incluyen en la ventana (sustituyendo expresiones anteriores), repitiéndose el paso 2; si no se han identificado excepciones, el programa muestra la generalización obtenida.

El método de selección de una «ventana» suele denominarse «filtrado por medio de excepciones», y su importancia es grande a la hora de procesar grandes bases de datos, con miles de datos, tal y como suele ser el caso con los datos procedentes de excavaciones arqueológicas. De esta manera, el procesamiento de la información tiene lugar en un lapso de tiempo aceptable.

El algoritmo CLS (Hunt et al. 1966) es una subrutina del programa principal. CLS es un acrónimo de *Concept Learning System* y deriva de ciertos trabajos en psicología experimental realizados entre 1950 y 1960; empezó como una teoría psicológica acerca de la formación de conceptos en la mente humana, y sólo posteriormente se convirtió en un programa de ordenador. La función del programa es encontrar el atributo más discriminante, dividiendo a continuación los datos con respecto a ese atributo. Según las implementaciones, existen varias formas de calcular el valor discriminante de los atributos; en ID3 Quinlan utilizó la estadística de la información, esto es, el grado de entropía:

$$H = \sum H P_i = - \sum P_i \log P_i$$

en donde  $P_i$  se calcula mediante el cociente entre la cantidad de ejemplos con el mismo valor en el atributo  $i$ , y la cantidad total de ejemplos en la base de datos. De lo que se trata es de comparar la entropía inicial a la entropía parcial de cada atributo, que calcularemos mediante la fórmula:

$$H(A) = \sum \text{prob}(A_i) \cdot H P_i$$

el incremento de información aportado por la clasificación según  $A$ , se evalúa por la disminución de la entropía producida respecto de la inicial:

$$\Delta I = H - H(A)$$

El criterio de discriminación utilizado en ID3 es ordenar los atributos en el árbol de decisión según el valor del estadístico  $\Delta I$ .

El lector no tiene que ser un matemático avanzado para poder usar cualquiera de los programas de inducción basados en el ID3, pues es el programa el que se encarga de hacer los cálculos. No obstante, puede tener interés la manera en que calcularíamos a mano el árbol de decisión asociado a los datos anteriores acerca de la cronología de las cerámica decoradas. La variable experimental es, obviamente la CRONOLOGÍA. Resulta sencillo calcular el contenido de información inicial: la probabilidad de que una cerámica decorada sea antigua, media o tardía es:

cronología tardía:  $4/10 = 0,4$

cronología media:  $3/10 = 0,3$

cronología antigua:  $3/10 = 0,3$

Aplicando la primera fórmula averiguaremos la entropía inicial:

$$H = -0,4 \log_2 0,4 - 0,3 \log_2 0,3 - 0,3 \log_2 0,3 = 0,529 + 0,021 + 0,021 = 1,571$$

Si consideramos ahora el primer atributo *tamaño*, obtenemos:

- el subconjunto *tamaño grande*, cuya probabilidad es 0,4. La entropía se calcula del modo siguiente:

Consideramos en primer lugar cuántas cerámicas grandes tienen una cronología tardía ( $3/4 = 0,75$ ); a continuación las cerámicas grandes con una cronología media ( $1/4 = 0,25$ ). Seguidamente aplicamos la fórmula anterior, de lo que resulta:

$$- 0,75 \log_2 0,75 - 0,25 \log_2 0,25 = 0,311 + 0,50 = 0,811$$

- en el subconjunto de *tamaño medio/cronología tardía*, la probabilidad es  $1/3$ , mientras que en el subconjunto *tamaño medio / cronología media* la probabilidad es de  $2/3$ . La entropía resultante es:  

$$- \frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} = 0,528 + 0,390 = 0,918$$
- en el subconjunto de cerámicas de *tamaño pequeño*, todos los objetos tienen una misma cronología (antigua), por lo que la probabilidad del subconjunto *tamaño pequeño/cronología antigua* es de  $3/3 = 1$ . La entropía resulta:  

$$-1 \cdot \log_2 1 = 0$$

- el promedio del atributo *tamaño* es:

$$0,4 * 0,811 + 0,3 * 0,918 = 0,6$$

Por lo tanto, si utilizásemos este atributo como primer nodo en el árbol de decisión, el incremento de información alcanzado sería:

$$1,571 - 0,6 = 0,971$$

Considerando el atributo *Decoración Pintada*:

- El subconjunto *Decoración Pintada Presente*, cuya entropía es:  

$$-0,75 \log_2 0,75 - 0,25 \log_2 0,25 = 0,811$$
- El subconjunto *Decoración Pintada Ausente*, cuya entropía es:  

$$-1/6 \log_2 1/6 - 1/2 \log_2 1/2 - 1/3 \log_2 1/3 = 0,431 + 0,5 + 0,528 = 1,459$$
- El promedio es:  $0,4 * 0,811 + 0,6 * 1,459 = 1,20$   
 El incremento de información asociado a este atributo es:  

$$1,571 - 1,2 = 0,371$$

Finalmente, en el caso del atributo *Decoración Incisa*:

- *Decoración Incisa Presente*:  

$$1,459$$
- *Decoración Incisa Ausente*  

$$-2 * 0,25 \log_2 0,25 - 0,5 \log_2 0,5 = 1,5$$
- Su promedio es:  

$$0,6 * 1,459 + 0,4 * 1,5 = 1,475$$
  
 y el incremento de información:  

$$1,571 - 1,475 = 0,096$$

Por consiguiente, construiremos el árbol de decisión partiendo del atributo *tamaño* (Fig. 5.3.). A partir de este árbol podemos inducir una asociación entre los objetos de tamaño pequeño y las de cronología antigua, si bien no aparece clara la relación entre los distintos tamaños y las cronologías media y tardía. El análisis continúa, ahora, limitándose a los datos que aparecen en cada uno de los conjuntos al pie de las ramas «grande» y «medio». En el primer caso, la entropía inicial del subconjunto es:

$$0,75 \log_2 0,75 - 0,25 \log_2 0,25 = 0,811$$

Tomando en consideración el atributo *Decoración Pintada* en este primer subconjunto:

- el subconjunto *Decoración Pintada Presente*, cuya entropía es nula
- el subconjunto *Decoración Pintada ausente*, cuya entropía es:  

$$-2 * 0,5 \cdot \log_2 0,5 = 1$$

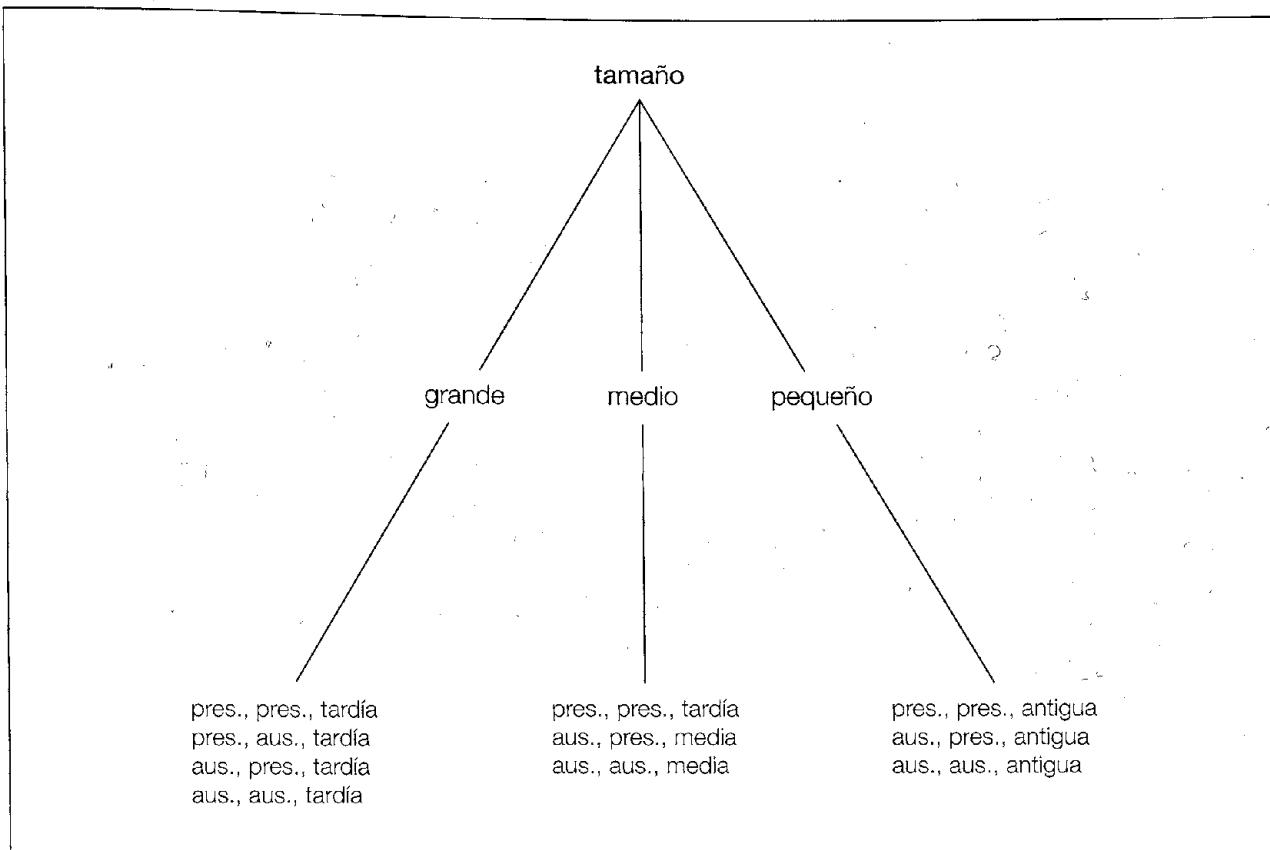


Fig. 5.3. Árbol de Decisión (Primer paso).

- El promedio resulta:  $0,5 * 1 = 0,5$   
Lo que representa un incremento de información de:  $0,811 - 0,5 = 0,311$

Utilizando el atributo *Decoración Incisa*:

- el subconjunto *Decoración Incisa Presente*, cuya entropía es nula.
- el subconjunto *Decoración Incisa Ausente*, cuya entropía es:  

$$- 2 * 0,5 \log_2 0,5 = 1$$
- El incremento de información es igual al obtenido para el atributo anterior:  

$$0,811 - 0,5 = 0,311$$

Resulta por tanto indiferente usar un atributo u otro como siguiente nodo del árbol de decisión.

Hechos los mismos cálculos en el subconjunto de datos asociados a la rama «tamaño medio» (no se detallan aquí), resulta que el incremento de información del atributo *Decoración Pintada* es mayor que en el atributo *Decoración Incisa*. El dendrograma resultante aparece en la Figura 5.4

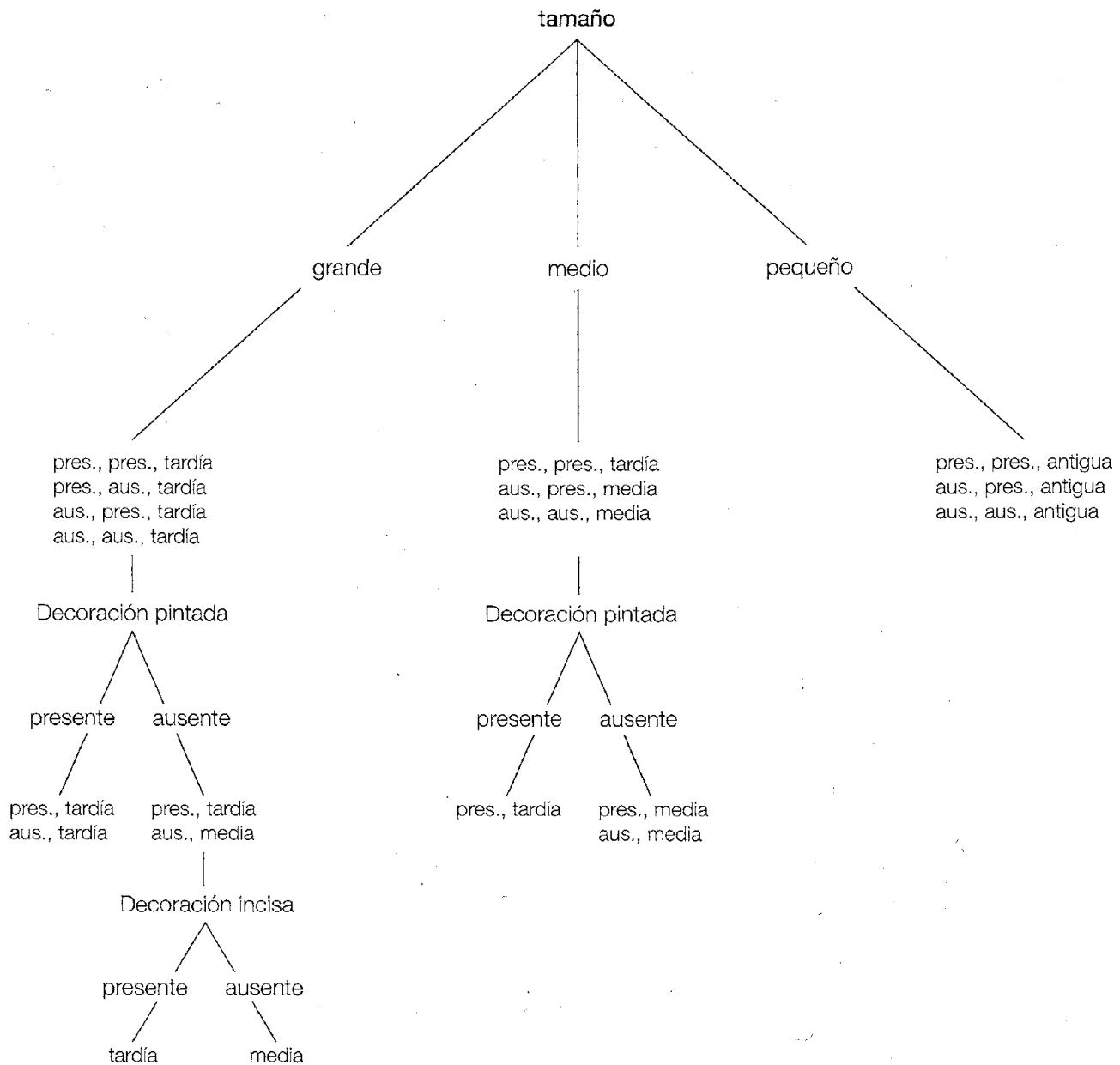


Fig. 5.4. Arbol de Decisión Completo (Segundo y Tercer paso).

### El programa I.X.L.

No abundan programas de ordenador de difusión comercial capaces de realizar inducciones automáticas. Los pocos publicados (cf. anexo) son bastante distintos entre sí, por lo que no podremos hacer una descripción de conjunto de los mismos. En su lugar, esta sección estará dedicada a la presentación detallada de uno de los programas existentes: I.X.L. (Parsaye y Hanson 1987, Parsaye et al. 1989, IntelligenceWare 1990). La descripción está basada en la versión 2.0 del programa, si bien IntelligenceWare acaba de anunciar la disponibili-

dad de la versión 3. La principal diferencia entre ambos es la velocidad de ejecución: la versión 3 es algo así como 20 veces más rápida (según anuncia el fabricante) y es capaz de gestionar la memoria extendida.

I.X.L. es un programa de «descubrimiento» o aprendizaje que funciona en un microordenador PC/AT ó IBM PS/2. El propósito del programa es definir las asociaciones entre casos y variables existentes en grandes bases de datos, presentando los resultados bajo la forma de reglas:

```

Actividad = Caza Mayor
SI
fauna = reno
útil = punta de flecha
7,5 ← Long. Max. Útil tipo 5 < 8,3

```

Es decir, «los yacimientos en los que se ha identificado actividades relacionadas con la caza mayor son aquellos en los que los restos de fauna más abundantes son los de reno, el útil lítico más frecuente es la punta de flecha, y la longitud máxima de estas oscila entre 7,5 cm. y 8,3 cm. En cierto sentido, la inducción automática en I.X.L. tiene la apariencia de un análisis de regresión «cuantitativo», en el que el resultado no es una ecuación general, sino la enumeración de las asociaciones encontradas.

Al empezar una sesión de trabajo con I.X.L. es preciso identificar los ficheros de datos que se han de analizar. Por ejemplo, imaginemos que nos interesa definir las asociaciones espaciales de una serie de artefactos líticos; la base de datos contiene, por tanto, el nº de inventario de cada artefacto, sus coordenadas  $x$  e  $y$ , su tipología (buril, bifaz, punta, etc.), diversos rasgos métricos (longitud, anchura), e índices de asociación.

Para que el programa proporcione resultados útiles para el arqueólogo, es preciso que esta base de datos sea lo suficientemente completa. La gran ventaja del programa es la posibilidad de investigar ficheros con miles de individuos y más de un centenar de variables. Obviamente, cuanto mayor y más compleja la base de datos, más tiempo de ejecución será necesario. Los ficheros de datos pueden presentarse en tres formatos distintos:

- Dbase III (\*.DBF)
- Lotus (\*.DIF)
- Ascii (\*.ASC)

El usuario puede además modificarlos a voluntad dentro del mismo programa, eliminando registros y/o campos, combinando distintos ficheros, restringiendo la búsqueda a unos pocos casos, etc.

Una vez seleccionada y modificada la base de datos, el usuario introduce los parámetros del módulo de descubrimiento. El primero de ellos consiste en especificar el objetivo, es decir, el esquema de dependencia entre las variables:

coordenada  $x \rightarrow$  tipología, dimensiones  
coordenada  $Y \rightarrow$  tipología, dimensiones

En resumidas cuentas, «seleccionar el objetivo» consiste en la identificación de la variable dependiente; seleccionaremos un objetivo distinto para cada uno de sus valores. El programa es mucho más efectivo con variables cualitativas que con variables cuantitativas; aún así, es capaz de encontrar asociaciones entre los atributos «independientes» y un rango determinado de la variable o variables dependientes. En este caso, vamos a definir rangos significativos en las coordenadas  $x$  e  $y$ , utilizando para ello información contextual: acumulaciones no aleatorias de material, presencia de hogares u otras estructuras. Por ejemplo, si definimos

$13,5 < \text{coordenada } x \leq 25,6$   
 $9,4 < \text{coordenada } y \leq 15,3$

El programa buscará todos los artefactos situados entre esas coordenadas y calculará una o varias reglas que definen las características tipométricas de todos los objetos encontrados en esa zona. El programa hubiese permitido también una búsqueda «ciega», es decir, una enumeración de todas las asociaciones posibles entre todas las variables y atributos; esa acción exigiría, no obstante, una gran cantidad de tiempo de computación (¡varias horas!).

Además de los objetivos, el usuario ha de fijar otros parámetros antes de lanzar el programa. Uno de ellos lo constituye el «nivel de interés» de la asociación. Alguna de las variables en la base de datos nos interesan más que otras. Por ejemplo, la longitud máxima de las piezas puede producir resultados engañosos si muchas de

las piezas están fragmentadas. En consecuencia, para cada objetivo  $O$  y cada variable  $v$  debe fijarse un número entre 0 y 100 que refleje el interés que tenemos (o la utilidad práctica) en el estudio del efecto de  $v$  sobre  $O$ . A primera vista puede parecer un tanto fuera de lugar un parámetro de «interés» como éste, no obstante su importancia se pone de manifiesto cuando lo que analizamos es la base de datos que contiene el registro completo de una excavación, en el cual se ha incluido una enorme cantidad de variables que probablemente no tengan relación alguna con el objetivo que hemos definido previamente. Si cambiamos el objetivo, introduciremos modificaciones en esos parámetros.

Los parámetros restantes que el usuario debe especificar para guiar y controlar el proceso de inducción son:

- *numero máximo de cláusulas en las reglas*: este parámetro especifica un límite máximo para la longitud de una regla. Aquellas que superen ese límite no serán inducidas.
- *número mínimo de casos*: fija un límite inferior para la cantidad de casos implicados en la inducción de una regla. Por ejemplo, el programa ignorará aquellas reglas aplicables (que sean verdaderas) a menos de 50 casos.
- *confianza mínima en las reglas*: define un límite inferior para la confianza que tenemos en una regla.
- *máximo margen de error*: define un límite superior para el error producido al estimar la confianza en una regla. Altos márgenes de error pueden haberse producido al utilizar muy pocos casos para inducir una regla.
- *porcentaje mínimo de la base de datos*: define un límite inferior para la fracción de la base de datos que se usará para inducir una regla. Así por ejemplo, las reglas aplicables a menos del 10% de la base de datos serán eliminadas.
- *significación mínima*: mide la «calidad» de un intervalo, según si la distribución de valores en ese intervalo es distinta en el resto de la base de datos. 0 significa que casi todos los rangos están siendo considerados. 100 significa que sólo se utilizan los más significativos.

Otros tres parámetros afectan la determinación de intervalos en las variables y objetivos con valores cuantitativos continuos:

- *generalidad mínima*: fija el límite superior de intervalo definido por el programa. Si es 0 sólo se tendrán en consideración intervalos muy pequeños; si es 100 tan sólo se producirán intervalos muy grandes.
- *generalidad máxima*: fija el límite inferior del intervalo definido por el programa.
- *incremento de la generalidad*: determina el número de intervalos entre los parámetros de generalidad máxima y mínima.

Supongamos que la base de datos contiene 1500 artefactos y que I.X.L. ha inducido la siguiente regla:

Confianza=80  
 $13,5 < \text{coordenada } X < 25,6$

Si

$15 < \text{longitud} < 18$   
 $5 < \text{anchura} < 26$

Margen de error = 2  
 Porcentaje de la base de datos aplicable = 15  
 Número de casos a los que es aplicable = 100

La regla afirma que los objetos encontrados en la zona del yacimiento determinada por esas coordenadas tienen unas dimensiones específicas. La longitud de la regla es 2, ya que se han asociado dos variables al objetivo: la regla contiene, por tanto, dos cláusulas. El *número de casos a los que es aplicable* explica que hay 100 artefactos en la base de datos con una longitud entre 15 y 18 cm, y una anchura entre 5 y 26 cm. El *porcentaje de la base de datos aplicable* afirma que la expresión  $15 < \text{longitud} < 18; 5 < \text{anchura} < 26$  es aplicable al 15% de la base de datos (100 es el 15% de 1500). La *confianza=80* otorgada a esta regla nos indica que 80 de los 100 artefactos con una longitud entre 15 y 18 cm y una anchura entre 5 y 26 cm. han aparecido entre la coordenada (x) 13,5 y 25,6. En otras palabras, en el 80% de los casos la relación inducida situación/ta-

maño es correcta. El *margen de error* = 2 indica que el factor de confianza puede ser redondeado en un 2%, con lo que el valor final se situará en el intervalo 82-78%.

El margen de error puede estar afectado por la cantidad de artefactos que se han utilizado para inducir la asociación. Por ejemplo, si tan sólo se hubiesen utilizado 10 registros, y hubiésemos impuesto el mismo nivel de confianza, el margen de error alcanzaría el 50%, debido a que no se disponía del número suficiente de datos para llevar a cabo la inducción. I.X.L. demuestra cómo puede controlarse el proceso de inducción:

- especificando un margen de error lo suficientemente bajo
- requiriendo un alto porcentaje de la base de datos para inducir reglas
- fijando un número mínimo de casos para los cuales la regla sea aplicable

El algoritmo usado por I.X.L. para inducir reglas a partir de grandes bases de datos está basado en *topologías difusas*, las cuales permiten definir distancias inexactas entre elementos. Este método encuentra su fundamento en ciertos principios matemáticos definidos por Postin (1974, cf. también Menger 1953); la idea fundamental estriba en la construcción de grupos difusos, y no en la conversión de los datos en puntos con coordenadas espaciales precisas. I.X.L. construye grupos topológicamente difusos, y a continuación generaliza esos grupos para detectar las reglas que caracterizan a sus miembros. Desgraciadamente, las características del algoritmo que utiliza I.X.L. para inducir reglas está protegido por una patente comercial, por lo que no es puplicable más allá de los pocos principios generales que acabo de reseñar.

### **Algoritmos genéticos**

En esta sección abordaremos una manera distinta de enfocar el descubrimiento de la regularidad en una base de datos. Dada la novedad del enfoque, no existen todavía aplicaciones arqueológicas del mismo, aunque Reynolds (1987) ha utilizado estas técnicas con unos propósitos un tanto distintos. Los trabajos de Reynolds se describen en el capítulo 6.

Los algoritmos genéticos están basados en ciertos principios extraídos de la teoría de la selección natural. Dicho así puede parecer un auténtico despropósito, sin embargo, la idea además de sencilla es realmente inteligente: el objetivo del algoritmo genético es *transformar* el resultado de un mecanismo automático de inducción simulando las leyes de la evolución biológica, fundamentalmente la *supervivencia de los individuos mejor adaptados*. En este contexto, el término «adaptación» significa adecuación a una tarea específica, definida de manera no ambigua al inicio del algoritmo; para ello, las expresiones resultantes son evaluadas de acuerdo con un criterio específico (velocidad de procesamiento, sencillez, claridad, valor predictivo, etc.). En otras palabras, las generalizaciones obtenidas mediante este método compiten entre sí, subsistiendo tan sólo aquellas que demuestran ser más apropiadas para resolver el problema en cuestión. Esta competencia simula también el comportamiento de los seres vivos, especialmente en lo que se refiere a la reproducción: el algoritmo genético produce diversas *generaciones* de expresiones, en cada una de ellas tan sólo aparecen las expresiones que han superado con éxito la lucha por la supervivencia en una generación anterior; obviamente, las reglas mejor adaptadas, aquellas con una mejor puntuación de acuerdo con el criterio inicial, son las que se *reproducirán* con más frecuencia en las generaciones sucesivas. La principal diferencia con la Evolución Biológica es que el entorno al que hay que adaptarse no cambia a lo largo del proceso: el propósito final es seleccionar mediante un procedimiento automático las «mejores» generalizaciones.

En definitiva, un algoritmo genético es un tipo peculiar de sistema estocástico en el que una cantidad de individuos se combinan y recombinan para constituir una generación completa. El algoritmo opera del siguiente modo:

1. Construye una población inicial de reglas. En la mayoría de los casos, esa población se configura por medio de una selección al azar entre todas las reglas posibles (espacio del problema).
2. Evalúa las reglas, y si el grado de efectividad global de las mismas es lo suficientemente bueno, detiene el programa y muestra la mejor de ellas
3. Si el grado de efectividad no es lo suficientemente bueno, ordena las reglas según su probabilidad de activación:

$$p = e / E$$

donde  $e$  es su puntuación y  $E$  la suma de las puntuaciones para todas las reglas

4. Calcula una nueva población («generación») de acuerdo con la probabilidad de activación. Para ello utiliza distintos operadores genéticos.
5. Repetir (3) y (4) hasta que se haya obtenido un candidato lo suficientemente bueno. Cada uno de los pasos en este bucle corresponde a una generación de individuos.

Los operadores genéticos utilizados en el 4º paso son:

- **reproducción:** su función primordial es transferir una expresión de una generación a la siguiente. Con ese fin, determina el grado de adaptación de cada una de las reglas existentes en la población de partida; las mejor «adaptadas» tienen una posibilidad mayor de aparecer también en las generaciones sucesivas. Una forma de implementar ese operador es programando una función que calcule el ajuste de cada una de las reglas, y sume los ajustes de toda la población. Ese ajuste total se divide entre el ajuste particular de cada una de las reglas, con lo que se obtiene una medida del «ajuste relativo» de cada una de las reglas. A continuación, se asigna a cada una de las reglas un rango de valores entre 1 y 100. Por ejemplo, supongamos que la población inicial contiene tres reglas, cuyos ajustes relativos son: 0.5, 0.25, 0.25. Los rangos asignados son: 1-50, 51-75, 76-100. Seguidamente la función genera un número aleatorio entre 1 y 100: si ese número se sitúa en el rango 1-50, entonces el programa hace una copia de la primera regla y la sitúa en la segunda generación; si el número aleatorio se sitúa en el rango 51-75, entonces la segunda regla se copiará en la siguiente generación, y así sucesivamente. Obsérvese que la primera regla tiene un rango mayor que las otras dos, puesto que su ajuste relativo es mayor; la probabilidad de ser seleccionada para reproducirla será, por consiguiente, también mayor.
- **mutación:** induce pequeños cambios y modificaciones aleatorias en las reglas durante su reproducción, de modo que en la generación siguiente aparecen reglas aparentemente «nuevas».
- **transmisión («crossover»):** permite la combinación al azar de dos expresiones que se intercambian parte del código en el que han sido implementadas. Una vez que se ha aplicado el operador de reproducción, las reglas que aparecen en la generación siguiente se combinan a pares, con lo que producen «descendencia», esto es, nuevas reglas con características mixtas extraídas de las reglas que se han combinado.

Veamos un algoritmo genético en acción; el programa BEAGLE -«Biologic Evolutionary Algorithm Generating Logical Expressions»- (Forsyth 1989, Forsyth y Rada 1986) es uno de los más sencillos y útiles para la inducción automática de hipótesis arqueológicas. El programa contiene seis componentes principales que suelen actuar sucesivamente unos después de otros. **SEED** («Selectively Extracts Example Data») transforma datos externos (por ejemplo, la descripción morfométrica de unos bifaces) en un formato apropiado. **ROOT** («Rule Oriented Optimization Tester») comprueba una serie inicial de reglas sugeridas por el usuario. **HERB** («Heuristic Evolutionary Rule Breeder») genera reglas de decisión de acuerdo con la selección natural. **STEM** («Signature Table Evaluation Module») ordena y estructura las reglas producidas por el módulo anterior. **LEAF** («Logical Evaluator And Forecaster») utiliza el resultado de **STEM** para producir predicciones o clasificaciones a partir de los datos introducidos en el primer componente.

El componente denominado **HERB** es el corazón de BEAGLE, pues implementa el algoritmo genético propiamente dicho. Su funcionamiento puede representarse por medio del diagrama de flujo que aparece en la Figura 5.5.

Supongamos que deseamos obtener una regla discriminatoria para distinguir la actividad económica que tuvo lugar en un conjunto de asentamientos estacionales. La base de datos contiene 194 yacimientos descritos por medio de los siguientes atributos:

<b>FLORA</b>	<b>1</b> = Pino, <b>2</b> = Encina, <b>3</b> = Abedul
<b>FAUNA</b>	<b>1</b> = Roedores, <b>2</b> = Ovícapridos, <b>3</b> = Bóvidos, <b>4</b> = Cérvidos.
<b>VOLUMEN</b>	Cantidad total de material lítico (en gr.)
<b>Nº DE HOGARES</b>	Cantidad de hogares distintos en cada asentamiento
<b>ACTIVIDAD</b>	10 tipos distintos de actividad, establecidos por analogía etnoarqueológica
<b>TIPOS</b>	Tipología de útiles líticos: <b>1</b> = buril, <b>2</b> = raspador, <b>3</b> = punta, <b>4</b> = pico, <b>5</b> = bifaz, <b>6</b> = núcleo, <b>7</b> = resto de talla.
<b>YACIMIENTO</b>	Tipo del yacimiento, <b>1</b> = en llano, <b>0</b> = en lugar de difícil acceso.

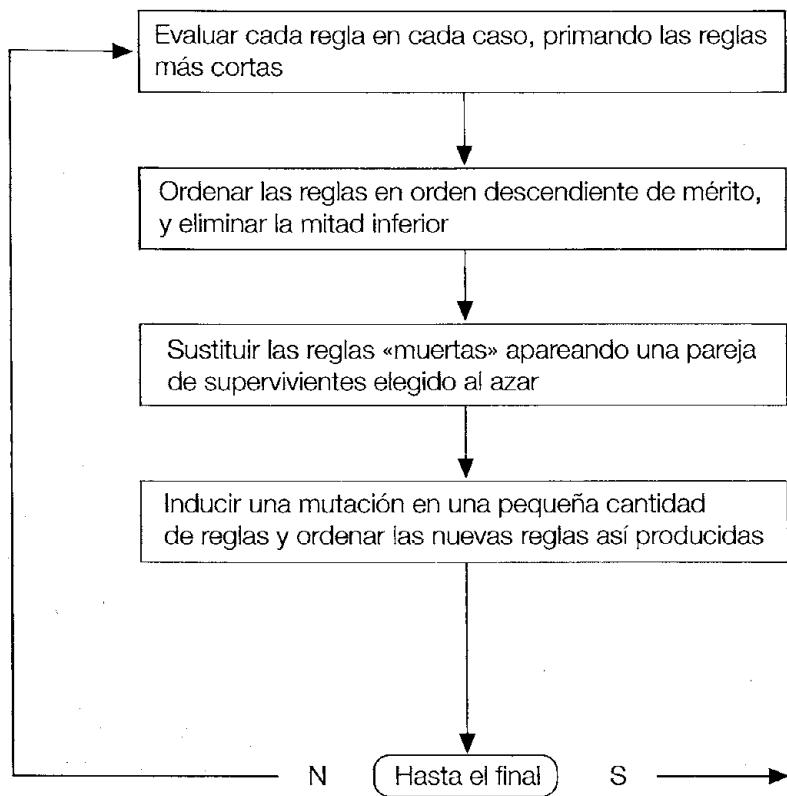


Fig. 5.5. Diagrama de Flujo correspondiente a un algoritmo genético.

El algoritmo genético crea al azar y modifica sucesivamente varias reglas, hasta que alcanza un conjunto de soluciones estables, de acuerdo con un coeficiente estadístico predeterminado. En nuestro caso, las reglas resultantes podrían ser:

(ACTIVIDAD = 2)				
\$				
((YACIMIENTO≤Nº HOGARES) — TIPOS)				
\$ 71,29	20	13	1	160
(FLORA≤2.5)				
\$ 56,17	20	28	1	145
((FAUNA≤2.6)<((FAUNA≥2)≥TIPOS)<VOLUMEN)				
\$ 54,08	21	32	0	141

Esta expresión empieza con el predicado (ACTIVIDAD = 2) que define el concepto que se pretende discriminar. En otras palabras, el objetivo es construir una regla que permita distinguir el tipo de actividad nº 2 (Caza Mayor, por ejemplo). El programa ha inducido cuatro reglas posibles. La más simple de ellas es:

(FLORA≤2.5)	.			
\$ 56,17	20	28	1	145

que afirma que la Actividad de Caza Mayor está asociada con aquellos yacimientos en los que se ha identificado polen de pino y encina. La expresión FLORA≤2.5 es verdadera para los valores 1 = Pino, 2 = Encina, y fal-

sa en el caso del valor 3= Abedul. El signo \$ lo utiliza BEAGLE como separador de reglas y no tiene ningún significado especial. El número 56,17 es la puntuación adquirida por esta regla si la utilizamos como discriminante; se basa en el coeficiente estadístico *phi* (una variante del Chi-cuadrado). Los otros números se leen así:

- 20 n° de yacimientos con polen de pino y encina en los que se han identificado actividades relacionadas con la caza mayor
- 28 n° de yacimientos con polen de pino y encina en los que no se han identificado actividades relacionadas con la caza mayor
- 1 n° de yacimientos con polen de abedul en los que se han identificado actividades relacionadas con la caza mayor
- 145 n° de yacimientos con polen de abedul en los que no se han identificado actividades relacionadas con la caza mayor

Como ninguna regla proporciona resultados perfectos, el programa ha inducido otras tres. El enunciado de la regla

((FAUNA≤2.6)<(((FAUNA≥2)≥TIPOS)<VOLUMEN)

es un ejemplo claro que BEAGLE no siempre proporciona discriminantes claros y evidentes por sí mismos. Una de las peculiaridades del programa, que complica la lectura de las inducciones por él generadas, es que las expresiones lógicas del tipo (FAUNA≥2) proporcionan un valor de 1.0 cuando son verdaderas y 0.0, lo cual transforma una expresión booleana en un coeficiente numérico cuyos valores pueden compararse con otros valores numéricos, tales como los que proporciona la variable VOLUMEN.

Analicemos con un poco más de detalle cómo ha calculado el programa esas expresiones inductivas: basándose en la teoría de los algoritmos genéticos parece haber sustituido una serie de reglas que no se ajustaban lo suficiente a los datos iniciales por nuevas reglas generadas a partir de las anteriores. Por ejemplo, imaginemos que en el segundo paso el programa hubiese producido aleatoriamente dos reglas:

(FLORA>2.5)  
(YACIMIENTO+Nº HOGARES)≤2)

Como resultado de la aplicación de los operadores de transmisión y reproducción, el programa habría producido una nueva generación de expresiones intercambiando variables, valores y relatores; por ejemplo:

(FLORA≤2)  
(FLORA≤YACIMIENTO+Nº HOGARES)

A continuación el programa obliga a esas dos expresiones a luchar por su supervivencia; el grado de «adaptación» de cada una de ellas, y por tanto la probabilidad de ser seleccionadas para formar parte de la siguiente generación se calcula mediante las estadísticas asociadas a la nueva regla: el coeficiente *phi* o puntuación de esa regla, n° de casos a favor y en contra, etc.

La dificultad con los algoritmos genéticos es que el usuario no tiene control alguno del mecanismo de descubrimiento; los programas basados en algoritmos genéticos producen asociaciones al azar basándose en juicios probabilísticos acerca de la idoneidad de las reglas así generadas y despreciando el conocimiento previo que tiene el usuario acerca de las mismas. No es de extrañar, pues, que los resultados sean impredecibles. ¿Qué sentido tiene decir que la suma de los valores correspondientes a las variables YACIMIENTO y N° DE HOGARES es mayor o igual que el valor de la variable FLORA? Esa relación existe, efectivamente, en la base de datos, por lo que es el resultado de una inducción «correcta», sin embargo no es «interpretable» en términos arqueológicos.

Quizás la dificultad estriba en el operador genético usado: si la transmisión de material «genético» no se produjese al azar, y la hibridación de reglas en generaciones sucesivas estuviese en todo momento controlada por criterios de evaluación heurísticos, esto es, específicos al problema que se quiere investigar, el mecanismo de los algoritmos genéticos sería útil para resolver el problema de la excesiva cantidad de expresiones generales sin interés para el usuario que resultan de la acción de los programas de inducción automática. En cualquier caso, el uso de algoritmos genéticos para representar inferencias científicas es, probablemente, uno de los aspectos más

interesantes de la moderna Filosofía Computacional de la Ciencia, si bien también el que necesita mayor trabajo teórico y de implementación. Algunos intentos iniciales en ese sentido han sido realizados por R.A. Young (1990a).

### Limitaciones de los programas de descubrimiento automático

El algoritmo ID3 ha sido aplicado en Arqueología por Fernández Martínez y García de la Fuente (1991), con resultados no excesivamente brillantes. Estos autores pretendían analizar las posibles asociaciones entre 25 variables que describen 389 tumbas excavadas en una necrópolis meroítica (Egipto, siglos III-I a.C.). Esas variables hacían referencia al momento cronológico, estado de la tumba (intacta o violada), tipo general, tipo de fosa, de suelo en fosa y cámara, de cierre de la cámara, presencia o ausencia de superestructura sobre la tumba, dimensión de la tumba, posición del cadáver, posición del cráneo, manos y piernas, tipo de acomodamiento del cadáver), presencia/ausencia de cuentas de collar, faldellín de cuero y sudario de tela, posición del ajuar dentro de la cámara, tipo cultural del ajuar y número de unidades del mismo,...

En la aplicación del ID3 a este caso se tomó sucesivamente como variable a discriminar cada una de las anteriores, haciendo que el algoritmo agrupara los estados del resto de las variables, de forma que se discriminaran unívocamente en función de dichas agrupaciones los estados de la variable que en ese momento se tomaba como diagnóstico. En el caso del valor FASE1 en la variable FASE, el programa obtuvo las siguientes asociaciones:

```
((TIPO C_OESTE)(ATAUD LECHO)(EDAD NO))
FASE 1 2/52
((TIPO C_OESTE)(ATAUD LECHO)(EDAD ADU)(MANOS PEL))
FASE 1 6/52
((TIPO C_OESTE)(ATAUD LECHO)(EDAD ADU)(MANOS PAR))
FASE 1 1/52
```

Es importante destacar la poca potencia de discriminación (el porcentaje de casos que son clasificados por la misma): en 52 tumbas adscritas a la fase 1, las tres reglas tan sólo son aplicables en 9 casos(2+6+1). La dificultad no estriba en la imposibilidad de descubrir regularidades en la base de datos, sino en la aparición simultánea de un número excesivo regularidades aparentes, cada una de las cuales tiene una relevancia muy escasa. Los programas de descubrimiento parecen ser incapaces de descubrir un esquema de regularidades o co-ocurrencias *general* a toda la base de datos, produciendo en su lugar una enorme lista de expresiones muy específicas adscritas a un pequeño número de casos dentro de la base datos. En otras palabras, los métodos y algoritmos que estamos analizando tienen tendencia generar árboles de decisión de tamaño monstruoso, que no proporcionen información alguna.

Un problema semejante ha sido registrado por M.S. Lagrange y M. Renaud (1987a) utilizando una variante del método de Michalski (Sallantin y Quinqueton 1984, Forsyth y Rada 1986) para analizar la planta arquitectónica de diversas iglesias góticas. El objetivo del análisis era proponer la definición intensiva correspondiente a una clasificación empírica de iglesias cistercienses, basándose en 47 variables binarias, que incluían características morfométricas de las plantas arquitectónicas, así como información geográfica y cronológica. El programa (TRINITA), desarrollado por Sallantin y Quinqueton, utilizaba la adscripción a las distintas clases empíricas como variable dependiente, calculando la conjunción de variables independientes asociada a cada uno de los grupos de iglesias. Las generalizaciones adoptan el formato siguiente:

(NAPABA) & ( $\neg$ SICLFR) STAT : 67 7 ; DIV : 197.7

Que significa: «La expresión «descriptivo NAvé\_con\_PAredes\_BAjas está presente y el rasgo Santuario\_Individualizado\_con\_Capillas\_Laterales\_de\_Forma\_Redondeada no está presente», es verdadera en el 67% de los casos del grupo A y en el 7% de los casos del grupo B». Expresando lo mismo mediante una regla de producción:

Si	NAPABA es verdadero
y	SICLFR es falso
Entonces,	la pertenencia al grupo 1 es verdadera.

La primera dificultad estriba en la longitud de la regla. ¿Qué es más conveniente, una regla de dos, de tres, de cuatro, o incluso de quince cláusulas? TRINITA, al igual que otros programas (por ejemplo el caso ya conocido de I.X.I.) permiten al usuario que defina ese parámetro, no obstante, el criterio usado es totalmente arbitrario. Aún con esta limitación, se obtuvieron 440 reglas. Aunque TRINITA dispone de un módulo específico para reducir el número de reglas, basándose en su porcentaje de validez, debiéramos plantearnos entonces si un número tan grande de regularidades es una descripción adecuada de la complejidad del esquema de asociaciones y co-ocurrencias subyacente en la base de datos, o bien si es una consecuencia de la arbitrariedad y globalidad de las estructuras de control.

Otras limitaciones o aspectos que afectan negativamente los resultados calculados por un programa de inducción o descubrimiento son los siguientes:

- La naturaleza cambiante de los datos iniciales. La calidad de una generalización depende de la calidad de los datos iniciales (ejemplos), y resulta imposible construir un conjunto completo y definitivo de ejemplos positivos y negativos, sea cual sea el concepto a generalizar.
- Variables irrelevantes: no existe modo alguno de saber si una variable es relevante antes de emplearla en una fórmula de generalización. Por ejemplo, ¿cómo saber qué es más importante, si el diámetro máximo o el diámetro de la base? La relevancia de los atributos no puede establecerse de manera absoluta, sino dependiente de los objetivos: para definir la cronología será más importante el diámetro máximo, en tanto que el diámetro de la base sirve para discriminar entre funciones posibles de una vasija.
- Importancia excesiva de los datos incompletos, que pueden llegar a falsear totalmente una generalización.
- «Ruido blanco» e incertidumbre. La Influencia del Azar (problemas de muestreo y errores de medida) afectan también la calidad de los resultados obtenidos por inducción.

Los programas que estamos analizando no buscan «a ciegas» regularidades, sino que intentan «generalizar» los ejemplos positivos y negativos de un concepto inicial que pudieran existir en esa base de dato. Dada esa condición previa, si la base de datos inicial contiene información heterogénea o redundante, será imposible definir una regularidad interpretable entre las regularidades descubiertas. El mecanismo automático (algoritmo) debiera estar dirigido en todo momento por el conocimiento previo que tiene el usuario acerca de la distribución de las propiedades en el conjunto de ejemplos de partida. Ese conocimiento es, precisamente, el que nos ha permitido distinguir entre «ejemplos positivos» y «negativos» del concepto que vamos a inducir; resulta evidente que ese conocimiento se expresará en términos de las propiedades que describen los ejemplos. En definitiva, los programas presentados en este capítulo son efectivos tan sólo en la medida en que hayamos definido previamente aquello que deseamos descubrir. La pregunta que vamos a resolver con ayuda del programa no es *«¿Qué regularidades pueden definirse en esta base de datos?»* sino, *«¿están asociadas las variables X, Y, Z en esta base de datos?»*

El conocimiento previo que necesitamos para «inducir» o generalizar conceptos arqueológicos adopta la forma de:

- expresiones observacionales (IECHIOS), que representan el conocimiento específico acerca de determinados objetos, situaciones o procesos
- una expresión general hipotética (que puede ser nula)
- conocimiento general que define los supuestos y limitaciones impuestos a las expresiones observacionales y a la expresión general hipotética. Suele adoptar el aspecto de un *criterio de preferencia* o criterio de validez.

La necesidad de conocimiento previo también es característica de la Inducción Estadística: en muchas ocasiones los arqueólogos nos olvidamos de la importante distinción estadística entre *variables experimentales* y *variables observadas*, aplicando indiscriminadamente la fórmula de la correlación a simples variables observables, cuando lo realmente importante es *transferir* el conocimiento que ya disponemos acerca de la variable experimental a un conjunto de variables observadas correlacionadas. En términos formales, tanto en el caso del descubrimiento de expresiones simbólicas como en el de expresiones estadísticas lo que se pretende es lo siguiente:

dado un conjunto de artefactos arqueológicos adecuadamente descritos por medio de diversos atributos ( $d_1, d_2, d_n \dots$ ), y una (o varias) variables experimentales o controladas ( $A$ ), hay que encontrar

los rasgos descriptivos comunes a los ejemplos de  $A$ , es decir, identificar (y resumir) un cierto grado de coherencia en los objetos que verifican  $A$ .

A diferencia de lo que pudiera crecerse, el objetivo no es *calcular los valores de la variable A a partir de los atributos que describen los artefactos arqueológicos*, sino cómo expresar de manera compatible a esos atributos la expresión general ( $A$ ) cuya idea básica se conoce.

### **Aprendizaje Automático en sistemas a base de Redes Neuronales**

Vimos en el capítulo 2 que, en una red neuronal, cada unidad de procesamiento (neurona) recibe estímulos a lo largo de sus conexiones de entrada, y los traduce en una respuesta de salida, que será transmitida, a su vez a lo largo de la conexión de salida que une esa unidad de procesamiento con las demás (cf. de nuevo la Fig. 2.5). Esa respuesta de salida es una simple función matemática, que suma todos los valores de entrada, y envía como respuesta un número que depende de esa suma. En ocasiones, el valor transmitido es el resultado de la suma (función lineal); en otras ocasiones, por el contrario, la neurona o unidad de procesamiento envía siempre el mismo valor (por ejemplo 1.0), siempre y cuando la suma de los inputs alcance determinado umbral (funciones sigmoidales o booleanas). La forma concreta en que la unidad relaciona la señal de salida con la suma de las señales de entrada se denomina *función de activación*.

En una red neuronal, las unidades de procesamiento o neuronas están conectadas por una gran cantidad de enlaces ponderados, por los cuales pasa la señal procedente de las unidades de procesamiento. Cada unidad recibe muchas señales a través de las conexiones de entrada; algunas de ellas proceden de otras unidades, otras proceden del mundo exterior, a través de un sistema de entrada de datos. Cada unidad tiene muchas de conexiones; sin embargo, sólo produce una única señal de salida. Esa señal de salida se transmite por las conexiones de salida (que son análogas de las *dendritas* en la neurona humana), cada una de las cuales termina en una destinación diferente. Cada una de esas ramas o conexiones de salida transmite la misma señal, pero con intensidad diferente debido al *peso* o *ponderación* que recibe cada una de las conexiones.

Cuando la red neuronal está en funcionamiento, cada unidad recibe cierta estimulación de las restantes unidades de procesamiento. Esas señales se modifican cuando pasan a través de conexiones ponderadas. Dado que los pesos o ponderaciones son distintos entre sí, cada unidad recibirá una estimulación distinta a la de sus vecinas. Como resultado, ciertas combinaciones de unidades se activarán con grados distintos según sea el estímulo que reciban.

Supongamos una red neuronal que contenga dos conjuntos de unidades (Fig.5.6). El primer conjunto contendrá una descripción del dato arqueológico que queremos interpretar.

Dispondremos, pues, de tantas unidades de procesamiento como variables o rasgos descriptivos sean necesarios. La estimulación de esas unidades procede del exterior, es decir, que su grado de activación coincide con el valor de la variable que describe el artefacto arqueológico: la activación de la unidad «longitud» será igual a 13.5 ya que ése es el valor de la variable; en el caso de atributos presencia/ausencia, la activación será 1 si el rasgo está presente y 0 si no lo está. El segundo conjunto de unidades contendrá tantas unidades de procesamiento como soluciones alternativas tenga nuestro problema: cronologías posibles del material arqueológico. El propósito es que la descripción de un material se traduzca en un vector de activaciones de las unidades del primer grupo, activándose a continuación alguna de las unidades del segundo grupo (cronologías), como resultado de las señales de respuesta enviadas a través de las conexiones que unen ambos grupos. Obviamente, cada unidad del primer grupo dispone de conexiones con todas y cada una de las unidades del segundo grupo, de modo que cualquiera de los resultados del problema tiene una probabilidad igual de ser activado.

En palabras más sencillas, si un artefacto es semejante a otro, la activación de las unidades del primer grupo (Descripción) será idéntica (o muy parecida en ambos casos), y por consiguiente también lo será la activación de las unidades del segundo grupo (Cronología). Si un artefacto es distinto a otro, la activación de las unidades del primer grupo será diversa, como lo será la señal de respuesta que éstas envíen y la unidad del segundo grupo activada en correspondencia.

Para que este objetivo pueda cumplirse debemos fijar el peso o ponderación de *todas* las conexiones que unen el primer grupo con el segundo de modo que cumplan el requisito antes establecido: objetos distintos, distintas respuestas; objetos semejantes, semejantes respuestas. Este proceso de *aprendizaje* puede ser realizado por medio de un algoritmo determinado. El más conocido de todos los empleados actualmente es

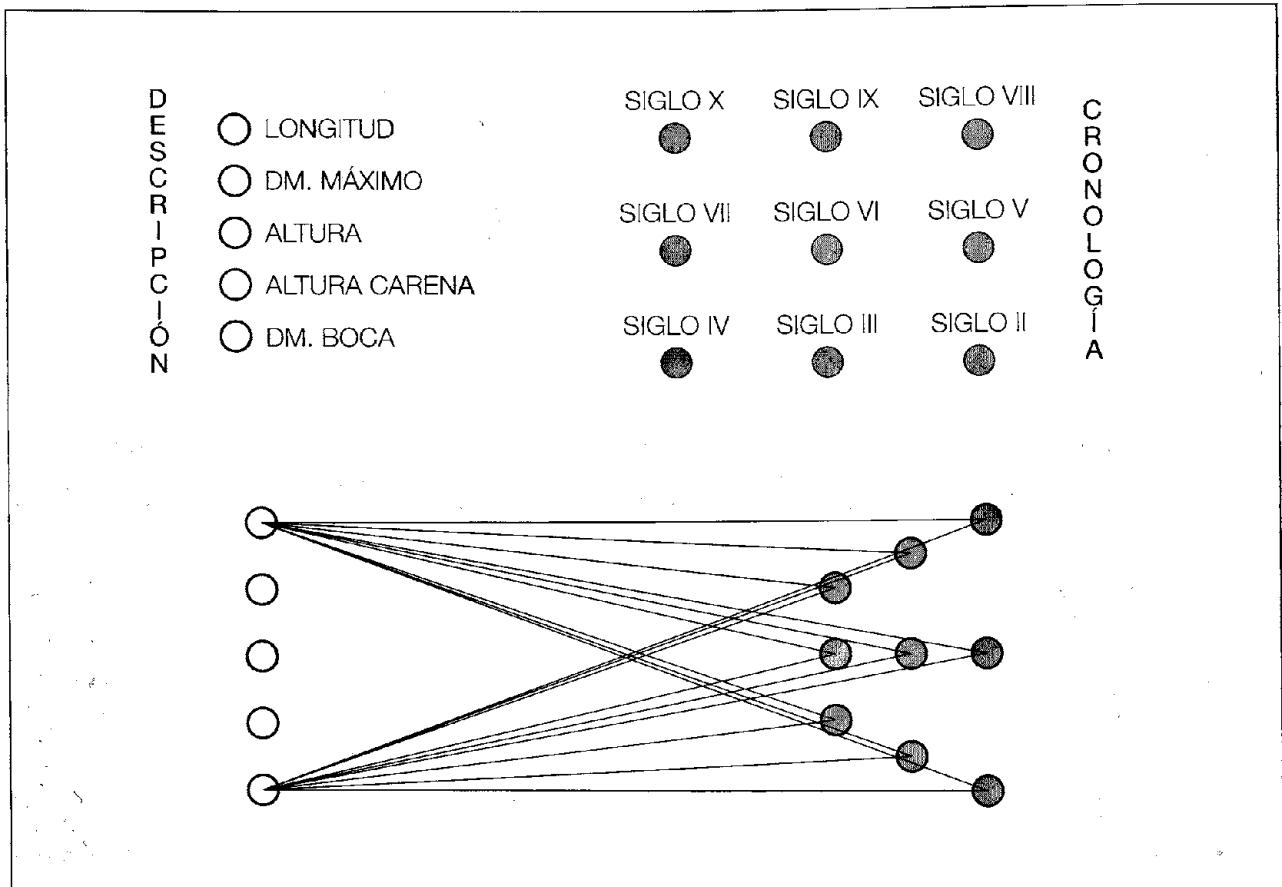


Fig. 5.6. Ejemplo de una red neuronal sencilla, con dos capas de unidades de procesamiento, una de entrada (input) y otra de salida (output). Todas las unidades de la capa de entrada están conectadas con todas las unidades de la capa de salida, si bien para no complicar en exceso el esquema sólo se han representado las conexiones de la primera y última unidad.

el denominado *algoritmo de retropropagación de los errores*, que nos permite calcular el valor de la ponderación de todos los pesos que unen las unidades de entrada con las de salida. Se trata, en realidad, de un *aprendizaje supervisado*, ya que necesitamos un conocimiento inicial (resultados asociados a descripciones) que hará las veces de «profesor» del programa, esto es, dirá en todo momento, qué aprendizaje es correcto y cuál incorrecto.

Este algoritmo de *aprendizaje* de los pesos o ponderaciones de una red fue diseñado inicialmente por Rumelhart, Hinton y Williams (1986). No es el mejor de todos los algoritmos de aprendizaje disponibles (cf. Caillard y Butler 1992), pero es muy útil en ciertas tareas específicas, siendo además el algoritmo cuyas propiedades muestrales son mejor conocidas. Existen abundantes implementaciones comerciales sobre microordenador por menos de 200 dólares, que aparecen en el anexo.

Las redes neuronales que utilizan el aprendizaje *por retropropagación de los errores* tienen dos características fundamentales. En primer lugar cuentan con un grupo intermedio de unidades de procesamiento -las *unidades ocultas*- cuyo propósito es simplemente computacional: permiten que el sistema pueda aprender a distinguir estímulos parcialmente semejantes. Las unidades en ese grupo, pues, no «significan» nada en términos arqueológicos, mientras que las unidades situadas en los otros dos grupos sí que tienen significado: «descripción» y «resultado» (Fig. 5.7). Topológicamente, todas las unidades del primer grupo están unidas a todas las unidades ocultas, mientras que todas las unidades ocultas están unidas a todas las unidades del último grupo. No existen conexiones, ni positivas ni negativas, entre las unidades de un mismo grupo.

## DESCRIPCIÓN

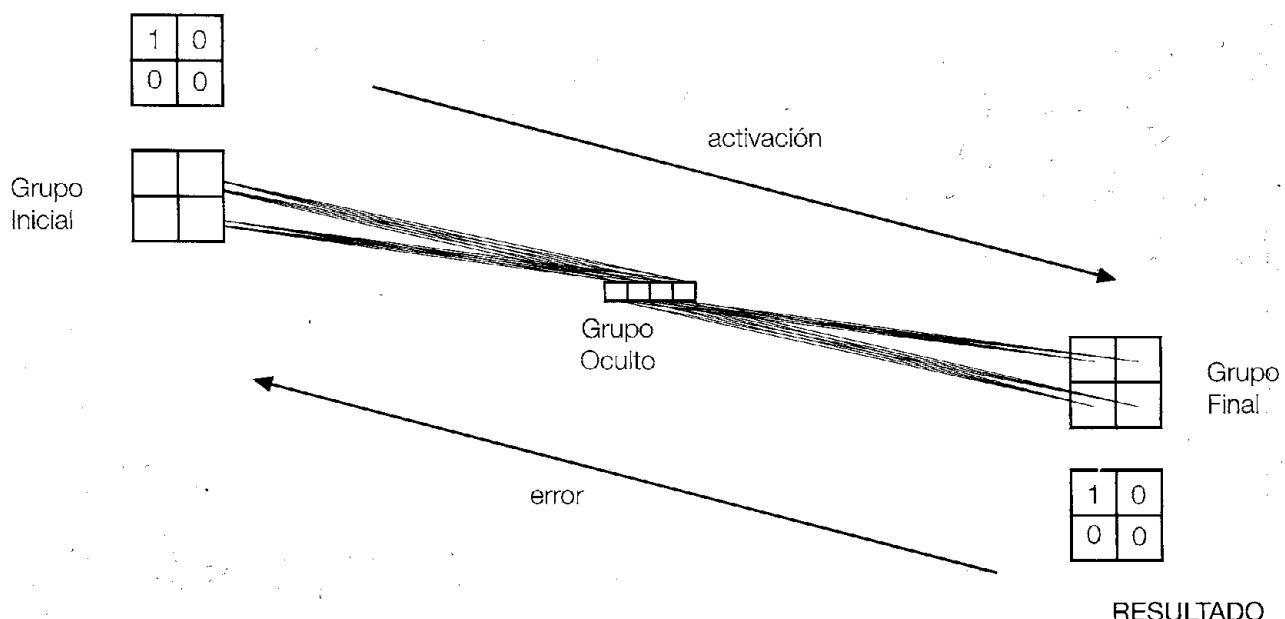


Fig. 5.7. Arquitectura básica de una red neuronal calculada por retropropagación.

Cada unidad recibe una estimulación procedente de las unidades de procesamiento en el grupo que la antecede. Así, el primer grupo recibe las estimulaciones del exterior (una base de datos que contiene una descripción del registro arqueológico), las unidades ocultas se activan cuando reciben estimulación de las unidades situadas en el primer grupo. La respuesta es activada, por tanto, por intermediación de las unidades ocultas. Todas las unidades, sin distinción del grupo al que pertenecen, usan la misma función de transferencia. Es decir, que la cantidad de estímulo que reciben es igual a la suma ponderada de las conexiones. La respuesta proporcionada por cada unidad se calcula mediante la función sigmoidal anteriormente expuesta.

Una vez definida la topología de la red, podemos iniciar el algoritmo de retropropagación. Para ello necesitamos una base de datos que contenga el valor de activación de las unidades del primer grupo y del segundo grupo. En otras palabras, precisamos de un conjunto experimental de datos en los que hayamos podido asociar previamente la descripción con el resultado. La misión del algoritmo es comparar la señal de respuesta emitida por las distintas unidades con la respuesta conocida de antemano; la ponderación de las conexiones entre todas las unidades se va corrigiendo iterativamente, cada vez que el sistema intenta calcular el estado de la activación de las unidades finales partiendo del mismo vector externo inicial, pero con una topología modificada. Como resultado del proceso iterativo de corrección, los resultados proporcionados por el sistema se van aproximando cada vez más a los conocidos. El error se reduce, pues, a medida que aumentan los ciclos iterativos de corrección. El procedimiento es, en realidad, bastante sencillo, ya que intervienen tan sólo sumas y diferencias de vectores, así como productos escalares de vectores por constantes, no obstante, todos esos cálculos se producen en paralelo, ya que cada unidad de procesamiento en el último grupo está unida a *todas* las unidades de procesamiento ocultas, por lo que es necesario corregir una gran cantidad de conexiones para cada caso. No debe extrañar, pues, que el proceso de aprendizaje, aunque se parte de una base de datos reducida, necesite varias horas de cálculo en un ordenador personal de mediana potencia.

El algoritmo de *retropropagación del error* es un proceso iterativo de corrección y refinamiento progresivo de la matriz de pesos neuronales creada por la función anterior: en cada ciclo, se utiliza el error (diferencia entre el resultado real y el calculado por el programa) para ajustar los pesos y la función de activación de cada una de las neuronas que componen el sistema, de forma que ese error se reduzca progresivamente, manteniendo constantes los estímulos iniciales. El algoritmo actúa del modo siguiente: para el estímulo *i* de la neu-

rona  $j$  en el tercer nivel (el del resultado), se asigna un valor aleatorio al peso  $w_{ij}$  ajustándolo progresivamente al valor especificado en el estímulo; para ello se suma el peso aleatorio inicial a un término determinado por el producto de la *proporción de aprendizaje*, del término de error y el valor de cada uno de los estímulos. Una vez calculados los términos de error en el último nivel, se repite el procedimiento para el nivel de neuronas ocultas y, finalmente, para el nivel inicial, variando en cada caso el término de error utilizado, ya que en cada uno de los niveles se utiliza la suma del error alcanzado en el nivel siguiente. El algoritmo recibe el nombre de la acción que lo caracteriza: la propagación hacia atrás del término de error.

La mayoría de implementaciones comerciales de este algoritmo ofrecen al usuario del programa varios resultados: en primer lugar los valores de la función de activación, algo que nos interesa bastante poco. El resultado de la inducción aparece de dos formas: en primer lugar como «aproximación» a la conceptualización del estímulo; así, si una de las matrices binarias era característica de cerámicas de baja calidad (escala = 1), la red utilizará los valores de la función para aproximarse a ese valor, 0.96 por ejemplo. Si la calidad fuese excelente (escala=4), la aproximación sería de 3.94 posiblemente. El usuario ha de estudiar estas aproximaciones para decidir si la red neuronal ha generalizado correctamente los ejemplos o si resulta preciso variar algunos parámetros para que su efectividad sea mayor. Se deja en manos del usuario la decisión final acerca de la «eficacia» del programa en esa situación concreta.

El segundo resultado útil es la predicción: junto a los estímulos hemos introducido varias matrices binarias sin especificar la calidad de cerámica correspondiente: es la red neuronal la que utilizará los valores de la función de activación para calcular esa adscripción, también en términos de aproximación, por lo que leeremos resultados como: 2.93; 1.57; 0.36; 3.75; etc. que utilizaremos para adscribir cada caso a una categoría distinta.

Veamos un ejemplo. Analizaremos las mismas representaciones rupestres que nos sirvieron para introducir el uso de los Sistemas Expertos: las Estelas Decoradas del Sudoeste (cf. Capítulo 3). Las 30 primeras unidades de procesamiento (neuronas) representan la información descriptiva, y coinciden con las siguientes variables:

- 1 Presencia de escudo
- 2 Presencia de espada
- 3 Presencia de lanza
- 4 Presencia de «espejiforme» (o maza, o «lira»)
- 5 Presencia de Figura Humana
- 6 Presencia de peine
- 7 Presencia de fibula
- 8 Presencia de carro
- 9 Presencia de casco
- 10 Presencia de más de una Figura Humana
- 11 Presencia de animales
- 12 Presencia de arco y/o flecha
- 13 Presencia de series de puntos
- 14 Presencia de diadema
- 15 Escudo y Figura Humana simétricamente dispuestos en plano horizontal.
- 16 Escudo y Figura Humana simétricamente dispuestos en plano vert.
- 17 Escudo en posición secundaria respecto a la Fig. Humana
- 18 Espada y lanza paralelas, flanqueando el elemento central
- 19 Espada y lanza, paralelas y juntas
- 20 Espada cruzada sobre la figura Humana
- 21 Espada y lanza independientes
- 22 Escudo con escotaduras en V en todos los círculos
- 23 Escudo con círculo exterior con escotadura en V, e interior sin ella
- 24 Escudo con círculo exterior sin escotadura en V, e interior con ella
- 25 Escudo con círculos concéntricos lisos
- 26 Escudo con líneas paralelas
- 27 Escudo con disposición radial
- 28 Escudo redondo, liso, sin escotaduras
- 29 Casco de cuernos
- 30 Casco de cresta

Cronológicamente, estos monumentos deben situarse entre el 1100 a.c. y el siglo VI a.c. (cronologías no calibradas). Este periodo de tiempo ha sido dividido en cuatro fases:

- BRONCE FINAL II
- BRONCE FINAL III -Horizonte de la Ría de Huelva-
- ORIENTALIZANTE
- POST ORIENTALIZANTE

Dado que suponemos que la seriación cronológica será diferente en distintas zonas geográficas, han sido añadidas nuevas variables:

- VALLE DEL TAJO
- VALLE DEL GUADIANA
- VALLE DEL ZUJAR
- VALLE DEL GUADALQUIVIR

Por consiguiente, la capa de salida de la red dispondrá de ocho unidades. Estableciendo un total de diez unidades intermedias (en la capa oculta), la red contiene un total de 48 unidades o neuronas. (Fig. 5.8).

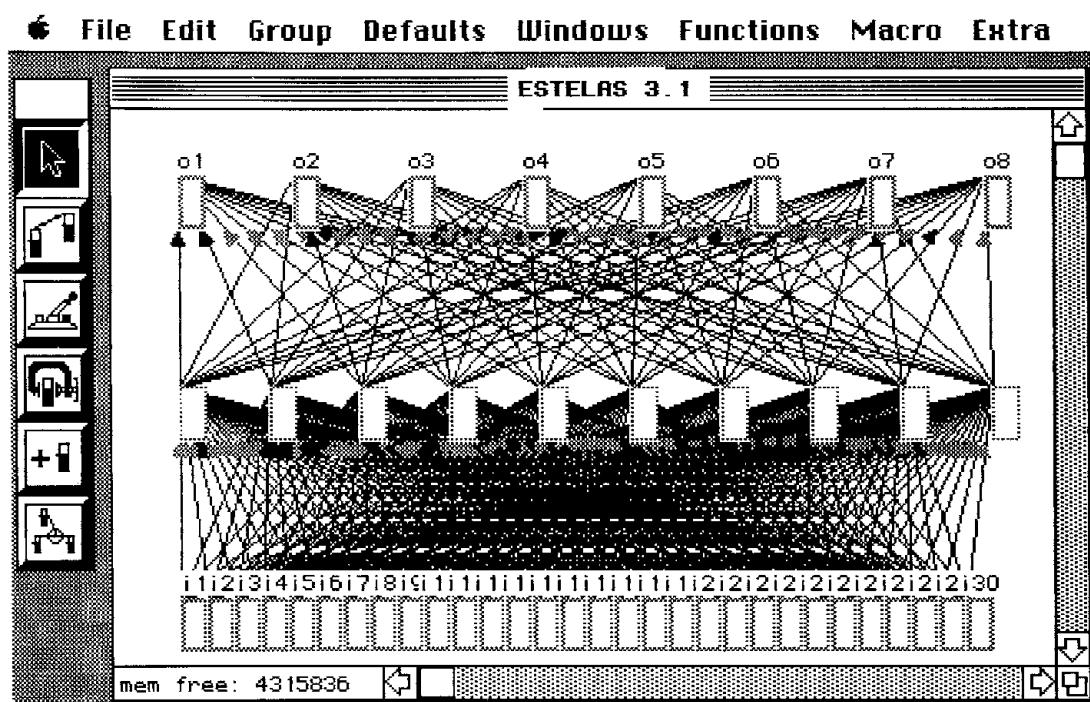


Fig. 5.8. Arquitectura del programa estelas en la que todas las unidades de procesamiento están interconectadas entre sí.  
(Usando el programa MacBRAIN 3.0, de Neurix, Inc.).

Se ha seleccionado un conjunto inicial de 33 estelas cuya cronología es conocida, debido a la aparición en ellas de algún elemento iconográfico datable por su semejanza con el registro arqueológico (espadas pistilliformes, espadas de lengua de carpa, fíbulas de codo, *ausencia* de fíbulas de codo, ... cf. Barceló 1989). Los pesos y niveles de activación de las unidades ocultas en red neuronal han sido calculados a partir de esa base de datos. En cada uno de los casos se ha consignado el área de geográfica de aparición, y la cronología más probable. Para que la red neuronal aprendiese por retropropagación a discriminar las Estelas de una Cronología y Región se han seguido los siguientes pasos:

1. Asignar al azar unos valores a las interconexiones iniciales (todos con todos)
2. Introducir un registro (descripción de *una* Estela)
3. Activar el estado actual de la pauta de interconexiones (utilización de la Función Sigmoidal de Activación)
4. Comparar el resultado obtenido con el resultado que aparece en la Base de Datos y que corresponde a la descripción de esa Estela)
5. Utilizar el algoritmo de retro-propagación del error para corregir los errores observados entre el resultado producido y el conocido.
6. Introducir un nuevo registro (otra Estela).

Este algoritmo ha sido repetido unas 2000 veces, lo que ha exigido unas cuatro horas de tiempo, hasta que se obtuvo una solución estable (los errores en el resultado entraban dentro de un intervalo de probabilidad aceptable).

Una Red Neuronal no puede utilizarse como si de los resultados de un Análisis Estadístico se tratase. El gráfico que obtenemos al final no es un dendograma comparable a los que obtenemos al final de un Análisis de Conglomerados, por ejemplo. En resumidas cuentas, una red neuronal no permite que su resultado sea *leído*, sino que una vez creada la pauta de interconexiones entre unidades de procesamiento, el usuario debe *experimentar* con ella a fin de contrastar (o no) sus hipótesis. No olvidemos que una Red Neuronal es un programa de ordenador, y no un fichero de números; como tal, el programa no contiene información válida, sino que es tan sólo un mecanismo de razonamiento que debiera permitirnos resolver ciertos problemas.

Lo primero que debemos hacer es averiguar si la topología de la red obtenida tras la aplicación del algoritmo de retropropagación tiene un mínimo grado de fiabilidad, esto es, si no se equivoca al establecer la cronología y la adscripción geográfica de las Estelas que han sido utilizadas en su creación. Los resultados obtenidos en esta primera experimentación han sido excelentes. En todos los casos el programa asigna la cronología correcta a las distintas Estelas. Se observa, no obstante, su resistencia a trabajar con «excepciones», como son ciertas Estelas con un esquema iconográfico característico del valle del Tajo, pero aparecidas en el valle del Guadiana. La Red Neuronal no proporciona resultados ideales cuando las diferencias entre los datos son muy escasas.

La siguiente tarea está directamente relacionada con el análisis de Estelas fragmentadas. De lo que se trata, ahora, es estudiar cual es la relevancia de cada una de las variables, ora juntas, ora por separado. Activaremos la Red Neuronal, no con descripciones de estelas, sino con descripciones hipotéticas, del tipo:

- ¿Qué pasaría si una Estela contuviese tan sólo un casco de cresta?
- ¿Qué pasaría si una Estela contuviese tan sólo un escudo de círculos concéntricos?
- ¿Qué pasaría si una Estela contuviese tan sólo un espejo y un carro?

Los resultados son extraordinariamente interesantes. Como era de esperar, los atributos más generalizados (los que aparecen en todos los monumentos) son muy poco discriminantes. Si aparecen aislados en una Estela difícilmente podremos conocer su cronología o lugar de aparición. Mayor significación tienen las variables que registran la distinta tipología de escudos. Así, se demuestra cómo el escudo con escotaduras en V en todos los círculos es propio del Bronce Final III, es decir, activando la unidad de entrada ESCUDO CON ESCOTADURAS EN V se obtiene una única activación en la capa de salida: la unidad BRONCE FINAL III. Lo mismo sucede el caso del escudo con escotaduras sólo en el círculo interior, mientras que el escudo con escotaduras en V en el exterior es característico del Orientalizante Pleno. Cuando en la Estela sólo aparece una figura humana, la simulación nos indica que se trata de un prototipo tardío, del siglo VI a.c.

Las variables iconográficas, esto es, aquellas que describen la manera en que están dispuestos los motivos representados en las Estelas han proporcionado poca información. Lo más interesante procede de la manera en que se asocian la espada y la lanza: cuando estos elementos flanquean la figura central (ya sea un antropomorfo o un escudo) se trata de un prototipo antiguo, datable en el Bronce Final II ó Bronce Final III. Cuando la espada y la lanza aparecen paralelas y juntas, la cronología más probable es la del Post-orientalizante, en tanto que la espada cruzada sobre la figura Humana aparece indistintamente en el orientalizante y en el Post-orientalizante, esto es, se activan dos unidades en la capa de salida para una sola activación en la capa de entrada.

Se han realizado diversas simulaciones para estudiar cómo responde la red neuronal ante estímulos contradictorios y ante estímulos que se refuerzan mútuamente. Comentaremos aquí sólo alguno de los resultados

más evidentes. Por ejemplo, al activar dos variables contradictorias en la capa de entrada, como son *fibula* y *escudo/figura Humana en posición simétrica vertical* aumenta la indeterminación cronológica, ya que se activa más de una unidad en la capa de salida; en otras palabras, una Estela que mostrase esas dos variables en su descripción no podría ser asignada, con claridad, a ninguna datación. En el caso de la activación a partir de las variables *espejo* y *casco de cuernos* observamos que al ser los resultados de cada una de las dos variables muy semejantes, el resultado conjunto no varía del obtenido al procesar por separado cada una de las variables. Lo mismo sucede en la activación de las variables *carro* y *escudo con escotaduras en V*, o bien *carro* y *figura humana*, si bien en el primer caso, la fijación cronológica de la segunda variable reduce la indeterminación de la variable *carro*.

En definitiva, cuando las variables activadas son contradictorias, en algunos casos aumenta la indeterminación, al equilibrarse la probabilidad entre cronologías alternativas. Por el contrario, siempre que la red sea activada con variables asociadas (presentes en los monumentos de la misma época), los resultados obtenidos en la activación individual se mantienen o se refuerzan (más detalles de este ejemplo aparecen en Barceló 1995a, 1995b).

Ahora bien, la validación de todos estos resultados dependerá de la verosimilitud que concedamos a la red neuronal. Ciertamente hemos basado esa verosimilitud en los resultados de la primera tanda de simulaciones, esto es, las experimentaciones con Estelas conocidas: en todos los casos, la red neuronal ha proporcionado los resultados esperados. Pero, la muestra de datos utilizada en la creación del programa puede ser parcial o incompleta, o bien, los resultados de la investigación arqueológica pueden superar rápidamente a lo que se sabía en un momento dado, y eliminar los criterios sobre los cuales se han creado los pares (DESCRIPCION-CRONOLOGIA). De ahí que resulte imprescindible insistir en el carácter fundamental que tiene la selección de los datos iniciales. La fiabilidad del análisis depende, pues, de la manera en que hayamos seleccionado esa información.

### Limitaciones de las Redes Neuronales

Aparentemente, el algoritmo de retropropagación y el aprendizaje a base de redes neuronales no parece seguir el algoritmo general de generalización como búsqueda, o algoritmo de Mitchell, con el que empezábamos el capítulo. El énfasis en el «ajuste» iterativo de los pesos neuronales y función de activación nos remite a un algoritmo de *optimización*, antes que a un algoritmo basado en la búsqueda heurística. A pesar de ello, no parecen haber demasiadas diferencias en lo que pudieramos llamar la «epistemología» del método. Tanto el «aprendizaje» o descubrimiento simbólico como el neuronal están basados en el mismo principio: encontrar el rasgo común a una serie de ejemplos positivos y/o negativos, calculando a continuación la «similaridad» de nuevos estímulos con los ya conocidos para predecir su adscripción o no al concepto al que se refieren los ejemplos.

Las diferencias en uno y otro caso estriban en la computación, esto es, en la manera de estimar los términos generales a partir de los ejemplos particulares. Si en el aprendizaje simbólico el término general a inducir es corregido paso a paso, a medida que se presentan nuevos ejemplos positivos o negativos, en el caso de las redes neuronales esto se hace por medio de una traducción del ejemplo en términos de una función matemática. En otras palabras, *cambiando* los pesos de las neuronas y no introduciendo nuevos estímulos. Los modelos de aprendizaje en una red neuronal son reglas o procedimientos que explican a cada una de las neuronas cómo modificar sus pesos y conexiones en respuesta a ciertos estímulos. De este modo, la redundancia de los estímulos iniciales no afecta al resultado final, si bien el algoritmo no funciona muy bien del todo (bajo poder predictivo) cuando las diferencias entre los estímulos iniciales son mínimas.

Otra ventaja de las redes neuronales es que algunos algoritmos de activación neuronal permiten el aprendizaje sin necesidad de definir ejemplos positivos y/o negativos; es decir, se trata de programas capaces de realizar inducciones *a ciegas* o no supervisadas. El método más conocido capaz de ello es el algoritmo de auto-organización (Kohonen 1988): su funcionamiento no se basa en la corrección progresiva del error, ya que al faltar los ejemplos no hay manera de calcular la diferencia entre el resultado esperado y el calculado, sino en la agrupación de estímulos que activan neuronas vecinas entre sí. El nivel final, aquel cuyas neuronas representan el resultado, corresponde a las agrupaciones de neuronas provocadas por los estímulos en el primer nivel. Los pesos de las neuronas se organizan de modo que las neuronas topológicamente unidas sean sensibles a estímulos semejantes entre sí. Al igual que en el algoritmo de retropropagación, los pesos neuronales son es-

tablecidos al principio de manera aleatoria, siendo mejorados progresivamente, si bien, el término de corrección no es ahora el término de error, sino la distancia euclídea entre el estímulo y el resultado.

Estas ventajas de las redes neuronales sobre el aprendizaje simbólico o heurístico, deben compensarse con las desventajas obvias del método. En primer lugar, el fenómeno de *caja negra*: el procedimiento de cálculo del algoritmo de retropropagación es mucho más complejo, matemáticamente, que las técnicas estadísticas multidimensionales. ¡Y se trata del algoritmo de aprendizaje neuronal más sencillo! En definitiva: la red neuronal es efectiva, y resulta sencillo para el usuario apercibirse de esa efectividad, no obstante, al desconocer los cálculos el usuario puede ser incapaz de crear una red neuronal. En este sentido es importante tener en cuenta que la parametrización inicial de la red es muy importante: antes de empezar una sesión de aprendizaje, el usuario debe fijar por sí mismo muchos de los valores iniciales de la función de activación, tales como la proporción de aprendizaje, nivel de ruido existente en el estímulo, tolerancia del entrenamiento y de la comprobación, cantidad de iteraciones necesarias, etc. Aunque los programas comerciales incluyen valores por defecto, estos no suelen concordar con el problema que se pretende resolver.

Otro inconveniente de las redes neuronales es más «tecnológico» que teórico: la tecnología informática utilizada hoy en día para programar redes neuronales no está lo suficientemente probada. Pongamos por caso la cuestión de la configuración de la red: parece estar claro por qué el primer y el último nivel de neuronas tienen el número de unidades que tienen: coinciden con el enunciado del problema y la solución que se busca. Pero ¿qué decir del nivel intermedio, de las «enigmáticas» neuronas ocultas, cuya función exacta no es bien conocida? Los resultados serán distintos si configuramos la red con dos, cuatro, cinco o las neuronas que sean. En otros casos podemos disponer incluso redes de cuatro o más niveles (en un microordenador el máximo suele ser 6-10 niveles), en los que las neuronas ocultas se agrupan en distintos niveles. ¿Qué configuración es la mejor y por qué motivo? El único criterio que proponen los informáticos es la fiabilidad de la predicción, o bien ciertas medidas estadísticas que hacen referencia a la distribución del error de clasificación. En cualquier caso, parecen soluciones *ad-hoc* para salir del paso, sin que revelen un conocimiento profundo de lo que está calculando el ordenador y por qué lo hace.

Quizás esto sea lo más irritante en la «inducción neuronal». Se trata de un procedimiento de generalización efectivo, pero cuya complejidad está muy por encima del nivel del arqueólogo que pretenda usarlo. La conclusión, a mi juicio, es evidente, ¡no debiéramos prescindir del método, sino empezar a estudiar álgebra!

### Comentarios bibliográficos

Acerca del aprendizaje y la inducción automáticas, el lector puede consultar los manuales de Forsyth y Rada (1986), Forsyth (1989), Weiss y Kulikowski (1990), Bergadano, Giordana y Saitta (1991), Thornton (1992), Langley (1994), Michie y Spiegelhalter (1994); en castellano, el libro de Cuena et al. (1985) y Castillo y Alvarez (1989). El libro de Kearns (1991), aunque exhaustivo, es probablemente demasiado difícil para el lector medio. Lectura fundamental sigue siendo el libro de Holland et al. (1986). En todos ellos aparecen descripciones detalladas de los métodos explicados en este capítulo y otros muchos. A ese respecto también pueden consultarse los libros de Pao (1989) y Gaines y Boose (1991).

Algoritmos genéticos: Goldberg (1989), Davis (1990).

Redes Neuronales: los libros citados en capítulos anteriores explican con detalle la manera de definir la función de activación de las neuronas y los distintos algoritmos. Los libros más sencillos son el de Caudill y Butler (1990) y el de Alexander y Morton (1990). El libro de Zeidenberg (1990), aunque completo, especialmente en lo que se refiere a ejemplos reales, es más difícil para el lector sin preparación específica. Otro ejemplo arqueológico «real» puede encontrarse en Gibson (1993) y en Claxton 1995.

## 6.+ SIMULACIÓN E INTELIGENCIA ARTIFICIAL

### Simulando la conducta del hombre prehistórico

Durante mucho tiempo los matemáticos han trabajado con Sistemas Artificiales que parecían capaces de modificar por sí solos su estado inicial. Me refiero a las técnicas de Simulación y Modelización, procedimientos de resolución de problemas en el que lo que se pretende averiguar es la situación futura de un sistema real. Esa solución aparece como la *evolución lógica* del estado inicial del sistema. Supongamos, por ejemplo, que hemos de averiguar, dado el conocimiento que tenemos de una comunidad humana en una fase cronológica determinada (Estado Inicial), si alcanzará o no un nivel de desarrollo social determinado (Estado Final). Lo fundamental, pues, radica en la formulación de la ley evolutiva capaz de esa transformación, es decir, en el *procedimiento* capaz de transformar el Estado Inicial en Estado Final.

En el fondo, esos Sistemas no son más que series de ecuaciones que *representan* ciertas entidades reales y que *simulan* su conducta. Esquematizándola al máximo, una simulación informática puede ser vista como «el arte de explorar un campo de posibilidades a partir de leyes formales definidas a priori» (Quéau 1986: 118). Esto es, una entidad artificial compleja (un sistema de ecuaciones) «que se refiere a sí misma» (*autorreferencia*) por medio de iteraciones y retroacciones. La simulación permite a los signos matemáticos (símbolos, al fin y al cabo) proliferar automáticamente, podríamos decir, casi «biológicamente», constituyendo una materia experimental susceptible de todo tipo de manipulación. El mecanismo informático del simulador analiza las distintas y sucesivas modificaciones del sistema real simulado centrándose en la «conducta individual» de sus componentes, conducta que suele representarse por medio de ecuaciones matemáticas, pero que puede programarse también mediante un Sistema a base de Reglas o un Sistema a base de Encuadres (Orientado a Objeto). El programa resultante, no obstante, se limitará a una simulación discreta (esto es, sin ecuaciones diferenciales, que son las que permiten representar mejor la evolución temporal de la entidad simulada).

La estructura básica de un Sistema Experto modificado de ese modo sería:

- una Base de hechos que contenga la descripción del Sistema en el momento inicial
- un conjunto de objetivos que especifiquen la forma que adoptará la entidad simulada al final de la simulación
- una serie de reglas, responsables de la modificación del Estado Inicial (hechos) en Estado Final (objetivos). Obviamente, esas reglas representan los procesos reales responsables de la evolución de la entidad o entidades simuladas.

La única, y fundamental, diferencia con los Sistemas descritos en capítulos anteriores es que las reglas de producción no representan un mecanismo de inferencia, sino procesos físicos, existentes en el mundo real, susceptibles de modificar el estado inicial del sistema, tal y como ha sido representado en la Base de Hechos. En términos de simulación, lo único que hemos realizado es substituir las ecuaciones matemáticas por reglas de producción, formato de representación más adecuado para aquellos fenómenos no expresables matemáticamente.

De todo ello se deduce que no cualquier sistema evolutivo es inteligente. A nadie se le ocurriría decir que una simulación es «inteligente», sino que es sofisticada, correcta, etc. Lo importante no está en la transformación del Estado Inicial en Estado Final, sino la «consciencia» que pueda tener el Sistema de que sus contenidos han evolucionado. Veamos un ejemplo (adaptado libremente de Tello 1989: 269ss).

Sea un yacimiento arqueológico en el que se asentó una comunidad humana esencialmente agrícola, conclusión a la que se ha llegado analizando la calidad de los suelos cultivables y la cantidad y diversidad de útiles relacionados con actividades agrarias. El yacimiento más cercano (y único) en varios kilómetros a la redonda, es un asentamiento dedicado esencialmente a la producción de mineral, situado en una región de suelos agrícolas pobres y con un registro arqueológico en el que faltan útiles dedicados a las actividades agrarias. En

el principio de la simulación, la situación en el yacimiento agrícola (yacimiento A) es más próspera que en el yacimiento B (minero): área de hábitat más extensa, estructuras de residencia más complejas, y arquitectónicamente elaboradas; abundancia de representaciones artísticas, etc. El yacimiento B, está al borde del abandono, ante la falta de recursos alimentarios. La finalidad de la simulación es analizar el mecanismo de intercambio entre ambos, de manera que en el Estado Final del Sistema ambos constituyan una red recíproca en la que los minerales son intercambiados por comida.

La representación del conocimiento se realiza por medio de encuadres, en el que las entidades fundamentales representan a los yacimientos. Los atributos son: recursos (agrícolas, mineros) y situación (próspera, crítica). La simulación es llevada a cabo por un conjunto de doce reglas (mecanismo de encadenamiento hacia adelante). La primera regla (esto es, aquella con un mayor nivel de prioridad) establece que todo yacimiento precisa de recursos alimentarios (agrícolas en este caso) para no ser abandonado. Una regla auxiliar establece que en ausencia de tales recursos el yacimiento estará en situación crítica.

La simulación empieza averiguando las necesidades de los yacimientos: si un yacimiento necesita algo y no lo tiene, entonces procurará obtenerlo. Esta situación está expresada mediante una regla que establece: «cuando un yacimiento intenta obtener algo ( $v$ ), si hay algo que considera que puede intercambiar ( $u$ ) y existe otro yacimiento que tiene lo que el primero busca, entonces el primero intentará establecer una red de intercambio con el segundo». El formato de esa regla sería:

Si	((x) ES UN yacimiento)
y	((x) INTENTA ((x) OBTENER (v)))
y	((y) ES UN yacimiento)
y	((y) TIENE (v))
y	((x) ≠ (y))
y	((x) CONSIDERA ((x) PUEDE INTERCAMBIAR (u)))
Entonces:	(x) ESTABLECE INTERCAMBIO CON (y)
	(u) A CAMBIO DE (v)

Otra regla afirmará que si se establece intercambio con un yacimiento y considera que tiene lo que se le solicita a cambio, hará efectivo el intercambio.

Este sistema experto representa por medio de reglas de producción ciertos fenómenos, que, aparentemente, van mucho más allá de la mera manipulación de símbolos. Fijémonos especialmente en el comportamiento de expresiones relacionales como «CONSIDERA». Es precisamente ella la que da la impresión de que la entidad simulada sea «inteligente»: el encuadre «Yacimiento», que en el fondo no es más que un sector de la memoria del ordenador que contiene cierta información, parece adquirir conciencia, ya que es capaz de «considerar» o «creer» algo.

Existen dos modos de implementar esa ilusión de «actividad cognitiva artificial», uno de ellos usa el término CONSIDERA como una mera expresión relacional, esto es, como una secuencia de caracteres alfabéticos que el ordenador registra y compara con otras; en el otro caso, el programa permite al encuadre «yacimiento» una cierta actividad propia (en forma de subprograma particular).

En el primer caso, existe una regla que asigna esa expresión concreta al encuadre. Por ejemplo:

Si	((x) ES UN yacimiento)
y	((x) PUEDE INTERCAMBIAR (y))
Entonces:	((x) CONSIDERA ((x) PUEDE INTERCAMBIAR (y)))

A su vez, la expresión «PUEDE INTERCAMBIAR» es producida por la regla:

Si	((x) ES UN yacimiento)
y	((x) PRODUCE (y))
Entonces:	((x) PUEDE INTERCAMBIAR (y))

En otras palabras, lo que parece «conciencia propia» por parte de las estructuras de datos no es más que un sugerente juego de lenguaje: lo que en el ser humano es una acción (el verbo «considerar»), para el orde-

nador es una propiedad (un adjetivo) de cierta estructura pasiva de datos. Los Sistemas Expertos suelen simular la acción verbal mediante adjetivos relacionales, que se convierten en acciones, ya que son capaces de «activar» reglas u objetos. Es decir, sólo si a esa estructura de datos que representa el concepto humano «Yacimiento» se le asigna el adjetivo «Considera», podrá ponerse en marcha un mecanismo capaz de generar ciertas acciones (asignar nuevos adjetivos). Con ello no quiero decir que esa implementación de la «consciencia artificial» sea un truco sin interés: la «Conciencia» del encuadre «Yacimiento» es algo más que una etiqueta. Su apariencia de actividad se ha logrado programando su definición (las acciones condicionales) y dándole la oportunidad de generar nuevas acciones; la expresión «CONSIDERA» es una condición para la asignación de nuevas expresiones, por lo que definiremos su significado enumerando las acciones que es capaz de producir en otras entidades. Por lo tanto, siempre y cuando un programador sea capaz de definir un concepto abstracto o una acción cognitiva exclusivamente en términos de sus condiciones y sus consecuencias, esa acción cognitiva será implementable mediante reglas de producción.

Este mecanismo no está tan alejado, como parece, del razonamiento humano. Es cierto que expresiones como «CONSIDERA» parecen meras etiquetas conceptuales, ahora bien, no se trata tan sólo de una mera palabra añadida a otra; «CONSIDERA» es en realidad un concepto ya que tiene un significado propio (acciones que contribuye a generar). En el caso cognitivo humano, la condición de activación denominada «CONSIDERAR», es sinónima de la expresión «TENER INFORMACIÓN ACERCA DE (x)», que contiene a su vez una función computacional capaz de realizar una acción (leer una Base de Datos).

Martin Biskowski (1992) ha desarrollado una simulación arqueológica bastante semejante. El Sistema Experto construido por este investigador contiene tres componentes mayores. El primero de ellos es la superficie de los recursos, que representa el área de distribución de la tierra cultivable y otros recursos naturales o artificiales en el espacio. El segundo componente es un conjunto de subprogramas de toma de decisiones, cuya misión es «generar» el razonamiento de diferentes individuos simulados. El tercer componente es un conjunto de subprogramas de arbitraje que controlan dónde se produce la decisión, qué tipo de información está disponible para cada individuo y cómo las decisiones y las no-decisiones se traducen en modificaciones de los subconjuntos en la superficie de los recursos.

La superficie de los recursos representa la distribución regional de diferentes productos importantes para las primeras sociedades complejas. El modelo utiliza información empírica del área centroamericana, como por ejemplo: tierra con diferentes potenciales para agricultura de secano u otras formas de agricultura más intensivas, fuentes de obsidiana, arcilla, madera y basalto. La superficie de recursos presenta, además, las relaciones espaciales entre los recursos.

Los programas de «toma de decisiones» constituyen, en realidad, un sistema experto que simula el conocimiento cultural y personal de un individuo acerca de las diversas maneras de interactuar con la superficie de recursos, con el fin de cumplir diversos objetivos y necesidades. Se incluyen reglas para elegir los individuos con los que se intercambiarán productos, así como el valor de diversos elementos. La función específica de cada sistema experto es simular cómo un individuo intenta satisfacer las necesidades y objetivos impuestos por un árbitro. El razonamiento simulado del individuo se traduce en tres tipos de acciones:

- interactuar con la superficie de los recursos
- hacer intercambios
- buscar información en otros individuos acerca de los procedimientos necesarios para obtener los bienes deseados.

Los subprogramas de arbitraje controlan la simulación. Se han dispuesto cuatro módulos:

- 1 Determinación de Procedimiento: El conjunto de reglas de producción usadas por los individuos serán seleccionadas de acuerdo con un modelo de la transmisión cultural de ideas. Un aspecto interesante de este subprograma es determinar cómo tratar con las fuerzas que impelen a un individuo hacia la posesión de reglas de producción contradictorias: patrones de comportamiento existentes entran en conflicto con el curso aparentemente racional de una acción. En otras palabras, la acción de un individuo no siempre es racional -en términos de nuestra propia vida acomodaticia de la Europa Occidental-, sino que está sometida a los esquemas ideológico-sociales de su comunidad.
- 2 Determinación de Objetivos/Necesidades: Los objetivos y necesidades de los individuos simulados dependen de factores tan diversos como el lapso de tiempo entre las decisiones tomadas en momentos diversos o el papel desempeñado por los individuos simulados en la sociedad (estatus). Las limitaciones

impuestas a los individuos son: obtención de los bienes subsistenciales necesarios, obtención/expansión del control sobre los recursos materiales y/o sociales, obtención de prestigio en su propio beneficio.

- 3 Resolución de intercambios: Los intercambios podrán aceptarse o rechazarse según sea el contexto social, la disponibilidad de individuos con los que se pueda intercambiar algo, la comprensión de los valores de los productos intercambiados. El subprograma de arbitraje habrá de determinar: si los productos intercambiados son obligatorios o determinados libremente, si el intercambio se produce por regalo, trueque u otro medio de interacción, quién inicia el acto de intercambio, si alguno de los participantes en el acto de intercambio excede la noción preconcebida de «coste».
- 4 Control del Mundo: Control de los diferentes factores que afectan el mundo simulado: variabilidad climática, inversión en obras de irrigación, etc.

Ejemplo de las reglas de producción del Sistema Experto ideado por Binkowski serían:

SI	1. Los productos/posesiones de algunas unidades domésticas exceden las necesidades previstas
Y	2. El exceso de bienes es intercambiable
ENTONCES	Hay un exceso de bienes intercambiables
SI	1. Conoce candidatos para establecer un intercambio
Y	2. Algunos candidatos son socialmente apropiados
ENTONCES	Hay disponibles suficientes individuos para un intercambio
SI	1. Hay un exceso de bienes intercambiables
Y	2. Hay disponibles suficientes individuos para un intercambio
Y	3. Se puede negociar el intercambio
ENTONCES	Se pueden obtener bienes utilitarios por medio del intercambio.

La propuesta de Biskowski es extraordinariamente interesante, fundamentalmente en lo que hace referencia al «objetivo» del sistema experto: simular la cognición de los individuos del pasado. El problema puede que se establezca en la implementación, debido al formato excesivamente rígido de los módulos. A excepción de los ejemplos aquí citados, Biskowski no ha publicado todavía una versión operacional del programa (cf. Biskowski 1992). En otras palabras, el programa es un conjunto de reglas, cada una de las cuales se refiere a una acción. Falta, por tanto, una representación más individual de la especificidad de cada uno de los individuos.

Eso se consigue mediante los llamados «lenguajes a base de actores», programas informáticos no muy alejados de los lenguajes orientados objeto en los que expresiones relaciones semejantes a las anteriores sirven de llamadas a subprogramas específicos que realizan una tarea determinada (recuérdese lo que decíamos sobre los lenguajes Orientados Objeto y los sistemas «distribuidos» en el capítulo 4). Así, en el caso anterior, siempre que el ordenador encontrase la palabra clave CONSIDERA, la ejecutaría como un «mensaje», con los siguientes componentes:

- el agente (o «actor») que va a llevar a cabo la acción de «considerar»
- el objeto de la consideración, esto es, la información que se va a introducir en la representación interna del actor como resultado del mensaje.

Pongamos por caso la simulación, en un Lenguaje a base de Actores, de la jerarquización social. ¿Cómo la representaríamos usando los elementos que se acaban de mencionar? Un Actor (*x*) está «subordinado» a un actor (*y*) si (*y*) es capaz de imponer tareas a (*x*). Un actor (*y*) podrá imponer tareas a un actor (*x*) si (*y*) envía un objetivo o mandato a (*x*) y (*x*) intenta ejecutarlo o cumplirlo. El mecanismo es bastante diferente del que veíamos en el caso de la simulación mediante reglas de producción. Aquí la «actividad» de los objetos u actores no se lleva a cabo mediante juegos de lenguaje, sino mediante programas específicos que «hacen» algo. Es decir, si en el caso anterior la expresión relacional «CONSIDERA» era para el ordenador una mera secuencia de caracteres alfabéticos que se asigna a ciertos objetos, en el presente caso, esa expresión contiene conocimiento procedural, pues se refiere a un programa que, efectivamente, ejecuta diversas acciones.

Cada actor dispone de su propio modelo de conducta (aquel que debe cumplir por sí sólo o en colaboración con otros actores), definido por un subprograma específico. La «consciencia artificial» de cada actor está representada por su conocimiento de algunas de las características de los actores que le rodean con los que interactuar enviando y recibiendo mensajes; en otras palabras, la «consciencia» de un actor no es más que la suma de conocimiento declarativo que tiene no sólo de los objetivos que debe cumplir (modelo de conducta), sino de los objetivos y características internas de los actores que le rodean. Esto es relativamente fácil de programar y provoca que los mensajes sólo puedan enviarse a aquellos actores que son conocidos por la unidad que emite dicho mensaje. Este principio, sin embargo, no es fijo, pues la identificación particular de un actor puede ser, a su vez, un mensaje, e integrarse en el conocimiento que tiene un actor acerca de los demás, lo cual contribuye a modificar paulatinamente la estructura de interconexiones posibles.

Una primera aplicación de los Lenguajes a base de Actores en el dominio de la simulación arqueológica es el estudio de Jim Doran acerca de las redes de intercambio entre comunidades prehistóricas (Doran 1987, Doran y Corcoran 1985). La Hipótesis que se pretende modelizar es la que propusieron en su día Wright y Zeder (1977): la proporción de bienes intercambiados está relacionada con sus niveles anuales de producción. En el programa EXCHANGE («intercambio») cada «actor» simula un asentamiento. Su modelo de conducta contiene conocimiento ‘tecnológico’ acerca de la manera de combinar materias primas -obtenidas directamente del entorno físico- en bienes de consumo. Los productos manipulados por los actores son totalmente abstractos en el sentido en que no poseen propiedades a excepción de su distribución y de las reglas de combinación a las que obedecen. Dado que la disponibilidad de materias primas es desigual, un actor individual está motivado para implicarse en una red de intercambio recíproco y obtener así los materiales necesarios para los bienes de consumo que ha de producir.

Según Doran, el esquema general de producción e intercambio deriva de la distribución de las materias primas, el conocimiento particular que cada actor acerca de los demás elementos en la red, la distribución de ese conocimiento (transmisión de mensajes), la estructuración concreta que adopta el intercambio, y las motivaciones (modelo de conducta) particulares a cada actor. Resulta evidente la incomparable mayor riqueza expresiva de esta simulación que usa técnicas de Inteligencia Artificial sobre la original de Wright y Zeder, limitada a una simulación estadística.

Más ambicioso es el proyecto actualmente en curso por Jim Doran y sus colaboradores en la Universidad de Essex (Proyecto EOS: Doran 1993, Doran et al. 1994), que pretende simular un modelo teórico de la evolución de las sociedades humanas. El modelo adoptado por Doran es el de Mellars (1985): en el sudoeste de Francia, durante el Paleolítico Superior, la riqueza y diversidad de los recursos alimenticios, la concentración de los mismos en el espacio y en épocas puntuales del año, su estabilidad a largo plazo y predictibilidad favorecieron una importante densidad de ocupación humana en grandes unidades co-residenciales, lo cual impuso un cierto grado de sedentarismo. En esas circunstancias la presión demográfica y la necesidad de coordinación sirvieron de impulso a la mayor complejidad de la organización social, que incluía la aparición incipiente de jerarquías sociales y de sistemas de privilegio. Esta tendencia a la complejidad social se reforzaría en el momento en que el sistema se viese afectado por disminución de recursos subsistenciales.

El sistema está configurado por un conjunto de actores situados en un contexto concreto que fija los objetivos «subsistenciales» de cada actor individual. De lo que se trata es de analizar con esos elementos la siguiente hipótesis: en circunstancias ideales cada miembro de la comunidad de actores satisface localmente su subsistencia sin interferir en las acciones locales de los otros actores en ese contexto. Sin embargo, cuando un proceso determinado obliga a los actores a actuar más cerca unos de otros, aparecerán interferencias en la satisfacción de los objetivos individuales.

La estructura interna del simulador es la siguiente:

- a un problema de explotación de recursos, el cual requiere cooperación entre los actores para poder ser resuelto. El sistema genera automáticamente una serie de recursos que deben ser aprovechados por los actores. Estos están distribuidos en distintas localidades y tienen que intercambiar y combinarse los recursos para producir con el fin de producir bienes de consumo. El objetivo de todos los actores es el consumo de esos bienes transformados.
- b la densidad de la población de actores. Las variables clave son: 1) la cantidad de actores que conoce cada actor y 2) el grado de competencia que existe entre los actores para conseguir los recursos existentes.
- c la respuesta cognitiva de los actores a la densidad de población (esto es, a la presión demográfica).

A medida que la densidad de población cambia, también lo hace el grado de dificultad de la tarea colectiva de adquirir y distribuir recursos. Cada actor dispone a su vez de un modelo social, bajo la forma de una jerarquía de clases. A medida que la densidad de población aumenta, cada actor simplifica el conocimiento que tiene de los demás por una descripción de la clase a la que pertenecen.

- d) la capacidad de un actor para generar e intentar la ejecución de ciertas estrategias para la consecución de su supervivencia. En general, esa estrategia se basa en la actividad coordinada de los actores más cercanos entre sí (según las especificaciones del modelo social del actor). Las ofertas de colaboración que recibe cada actor son diversas, por lo que tiene que elegir entre ellas para decidir con qué grupo iniciará una actividad concertada. En circunstancias de presión demográfica, como las que se intenta estudiar en este programa, esas decisiones convergen entorno a una organización jerarquizada de los actores.

La implementación informático de EOS está definida por:

- un entorno simulado bi-dimensional o «paisaje» con una población de agentes móviles y recursos cambiantes que proporcionan «energía» a esos agentes.
- agentes estructurados como sistemas de producciones, con reglas que «reactivamente» conectan el input sensorial (percepción que el agente tiene del «paisaje») con la acción. Hay reglas que implementan la comunicación inter-agente y que generan, mantienen y actualizan planes simples o modelos sociales de los otros agentes. En otras palabras, cada agente dispone de un modelo o representación de los demás agentes y de sus intenciones.

El programa (EOS1) funciona con los siguientes pasos:

- 1 Simulación de la recogida de información concurrente y asíncrona: cada agente adquiere información acerca de los recursos existentes en el «paisaje» y acerca de las intenciones y conocimientos de otros agentes. Fijación de los objetivos para la adquisición de recursos y propuestas de acción colectiva propuestas por los diferentes agentes.
- 2 Comunicación inter-agentes, que conduce a la adopción de un pequeño número de planes colectivos para su ejecución.
- 3 Ordenación social de los agentes según las propuestas sugeridas. Aumenta el «prestigio» individual de aquellos agentes cuyos planes de acción colectiva han sido adoptados por el grupo.
- 4 Una vez que un agente ha adquirido prestigio, se activan relaciones de tipo líder/seguidores semi-permanentes. Agrupación social de los agentes.
- 5 Formación de jerarquías (grupos de grupos de grupos) por medio de la instanciación recursiva de mismo tipo de procesos de formación.

Las próximas versiones del programa (EOS2, Doran 1993, comunicación personal) pretenden superar algunas de las limitaciones «antropológicas» de la versión inicial, en especial la excesivamente funcionalista y reduccionista descripción del proceso de formación de las jerarquías sociales.

En definitiva, los lenguajes a base de actores permiten representar parcialmente la complejidad de ese modelo social, especialmente aquello que cada actor conoce acerca de los demás (si pertenecen o no a su propio grupo), y la efectividad de la resolución de conflictos en una comunidad en la que se requiere un esquema de intercambios recíprocos para el proceso productivo. La efectividad de los intercambios varía con respecto a la variación en la base de datos que representa las «creencias» que cada actor tiene con respecto a los demás: el intercambio es más sencillo dentro de un mismo grupo.

Otro programa informático que pretende modelizar la manera en que los hombres prehistóricos tomaban decisiones económicas, esto es, la manera en que adoptaban nuevas estrategias productivas es el, dirigido por R.G. Reynolds (Reynolds 1984, 1986, 1987, 1989, 1990, 1991, Savatsky y Reynolds 1989). El mecanismo informático sobre el que se basa la simulación es, sin embargo, bastante distinto al de Doran, ya que utiliza algoritmos genéticos (cf. Capítulo 5) y no lenguajes de programación a base de actores. De entre los distintos programas particulares integrados en este proyecto (simulación de una economía de base ganadera, simulación de la conducta económica en grupos de cazadores-recolectores) he elegido el estudio de la adopción de una economía agrícola en el valle de Oaxaca, en el México pre-colombino.

El programa intenta reproducir los motivos que llevaron a un grupo humano, situado en un ecosistema específico, a decidirse por una economía de producción (de base agrícola), antes que a una economía de re-

colección. El programa desarrolla la Teoría que sobre el Origen de la Agricultura formulase Kent Flannery a fines de los años 60 (Flannery 1968), proponiendo un método que permite la operacionalización de dicha teoría, el principal punto débil de la formulación original de Flannery. El supuesto de partida es el siguiente:

«Consideremos un grupo de cazadores-recolectores en un ecosistema semiárido como el de Oaxaca (Méjico). En determinadas épocas del año, ciertos productos son más abundantes y fáciles de recoger que en otras; la estrategia económica de ese grupo humano, pues, debe adaptarse al ciclo anual de los recursos disponibles. La adopción de la agricultura provoca conflictos en la gestión y programación de la secuencia estacional de las restantes actividades subsistenciales, ya que distrae una importante cantidad de trabajo en la época del año en la que la recolección es más eficaz».

El programa informático propuesto por R.G.Reynolds (sin nombre o acrónimo que lo identifique) calcula la manera en que los miembros de ese grupo organizarían sus actividades económicas en ese supuesto, equilibrando la cantidad de trabajo necesaria para cada una de esas actividades. La idea fundamental estriba en definir un punto de equilibrio idóneo entre todas las actividades subsistenciales, de modo que la subsistencia y reproducción del grupo quede asegurada. Si el modelo fuese correcto, los resultados del programa debieran ser idénticos a los resultados de la excavación arqueológica: la cantidad de restos de comida medida por los arqueólogos en el transcurso de la excavación debe ser idéntica a la cantidad de comida calculada por el programa a partir del punto de equilibrio supuesto. Para conocer más detalles acerca de la excavación arqueológica sobre la que se basa la simulación, el lector deberá consultar la monografía original de Flannery (1986).

La simulación utiliza un grupo humano de cinco o seis individuos (tal y como se deduce de la excavación del yacimiento de Guilá Naquitz) que tiene acceso a 10 estrategias económicas (o actividades de subsistencia) alternativas durante un periodo de 4 meses; esas estrategias económicas han sido establecidas en función de los hallazgos arqueológicos de restos faunísticos y vegetales. Se supone una dedicación de 10 días para cada una de las 10 estrategias, lo que da lugar a que el grupo humano recurra a ellas en repetidas ocasiones durante el periodo fijado. Cuantas más veces se acuda a una estrategia determinada, más influirá ésta en el registro arqueológico (mayor cantidad de huesos animales o de restos vegetales).

Cada vez que un miembro del grupo acude a una de las 10 estrategias adquiere una cierta cantidad de comida. Tanto en la excavación como en el programa de simulación se han medido las cantidades de proteínas y de calorías proporcionadas por la subsistencia así obtenida; las medidas permiten estudiar el grado en que las necesidades energéticas de la población están satisfechas. Otro parámetro importante hace referencia al esfuerzo necesario para reunir esa cantidad de comida. El programa utiliza una estimación de la cantidad de proteínas y calorías por unidad de área en la que el grupo humano busca los recursos. Otro tipo de información que el programa necesita se refiere a la «competencia» entre las distintas actividades, dicho de otro modo, qué estrategias son viables en un mismo momento, dadas las circunstancias del entorno en ese instante.

Dados estos parámetros, el objetivo de la simulación es constatar las modificaciones inducidas por ese grupo humano en el orden e importancia relativa entre las distintas actividades. Transcurrido un periodo de tiempo determinado (4 meses), debe llegarse a una secuencia de actividades en equilibrio. Para cumplir este objetivo, el programa diseñado por Reynolds consiste, en realidad, en un algoritmo genético (cf. Capítulo 5). En una primera etapa, el programa asigna aleatoriamente un número de orden y una intensidad específica a cada una de las 10 actividades o estrategias. Con ayuda de la información disponible acerca del grupo y del entorno (ecosistema), ese orden e intensidad se modificarán paulatinamente, hasta definir una situación de equilibrio. Los distintos ciclos del programa pueden expresarse del modo siguiente:

- se elige una muestra de estrategias de producción
- se evalúa la efectividad de las distintas estrategias seleccionadas, reproduciendo, a partir de la información disponible, el mecanismo de toma de decisiones que, verosímilmente, produjo esa selección.
- se utiliza la regla de decisión así obtenida para producir nuevas decisiones económicas
- se utilizan las decisiones económicas anteriores para modificar la muestra inicial de estrategias de subsistencia.

No es este el lugar para desarrollar la forma en que se ha implementado el algoritmo; una introducción general a los mismos ya ha sido presentada en el capítulo 5. La cuestión fundamental a la que debiera prestar el lector una atención especial es al hecho que el programa aparenta ser capaz de alterar su conducta basán-

dose en su experiencia. En resumidas cuentas: la toma de decisiones subsistenciales es modificada según se aproxime la secuencia de estrategias a un punto de equilibrio, que se ha establecido en función de ciertas medidas de consumo proteínico y calórico por el grupo humano simulado. El lector interesado puede consultar (Reynolds 1986) para saber más detalles acerca del algoritmo genético en cuestión.

Para poder estudiar la relevancia de la agricultura en una secuencia estacional de estrategias subsistenciales, Reynolds ha trabajado con distintos casos:

- un modelo pre-agrícola (ninguna de las 10 estrategias posibles hace referencia a la agricultura).
- un modelo de agricultura incipiente (en el que la agricultura es una más de las 10 estrategias posibles).

La comparación de los puntos de equilibrio a los que se llega en cada uno de estos dos modelos, y las estimaciones de consumo proteínico y calórico, permiten analizar la relevancia de la agricultura entre todas las actividades, así como los efectos que la adopción de la agricultura produjo en pequeñas comunidades asentadas en ecosistemas semiáridos, en los que la secuenciación de actividades subsistenciales es crítica.

Los proyectos de J. Doran y de R. Reynolds expuestos en esta sección debieran permitirnos examinar más a fondo una de las cuestiones más importantes abordadas en este libro: el mecanismo informático no es tan sólo un vehículo de representación, sino una representación en sí. En ambos casos se afirma explícitamente que los seres humanos *actúan* del mismo modo que lo hace el ordenador. Los trabajos de Doran y de Reynolds, no obstante, demuestran la no universalidad de los mecanismos sociales de toma de decisiones, al menos en lo que hace referencia a la adopción de estrategias económicas. En ambos programas las decisiones del grupo se llevan a cabo gracias a la *integración* de las decisiones individuales; se parte del principio, demostrable formalmente, que los límites en la comunicación entre los individuos imponen una restricción en la forma e idoneidad de las decisiones del grupo. Ahora bien, lo que diferencia el programa de Doran del de Reynolds es la función matemática utilizada para calcular esa integración. En el caso de EOS se trata, aparentemente, de una acumulación simple de decisiones individuales (mensajes cruzados por los distintos actores); no obstante, la existencia de numerosas estructuras de control, ya a nivel individual, ya a nivel colectivo (información sobre el entorno), permiten la representación de decisiones extremadamente complejas. En el caso del programa de Reynolds, la acumulación de decisiones individuales está calculada por medio de un algoritmo genético.

La diferencia entre ambos sistemas es, en realidad, la diferencia entre un programa declarativo y un programa algorítmico. ¿Pueden lograrse resultados comparables acerca de la conducta humana en el pasado utilizando diferentes métodos informáticos? Tendremos que esperar hasta la publicación definitiva de ambos proyectos para poder responder a esta pregunta fundamental. Provisionalmente, podemos decir que en ambos casos parece haberse llegado a un resultado semejante: la debilidad estructural de los mecanismos de toma de decisiones en grupos humanos con una estructura social igualitaria; la toma de decisiones en estos casos no está afectada por la cantidad de individuos en el grupo, sino por la cantidad de información recogida por los miembros individuales y por la forma en que esa información circula entre el grupo.

### **Simulando la “inteligencia” de los arqueólogos**

Los programas presentados en el capítulo anterior son en realidad mecanismos «universales» de generalización: exploran la posibilidad de que en una base de datos cualquiera puedan -o no- existir regularidades. Lo que hacen esos programas no es más que traducir o re-escribir un concepto inicial, de manera que la nueva expresión, siendo formalmente equivalente a la de partida, tenga ciertos atractivos adicionales. Por lo tanto, las regularidades así descubiertas no son «interpretaciones» de una población inicial de datos o resultados observacionales, sino expresiones equivalentes que expresan de manera más simple y comprensible el esquema de co-ocurrencias existente en la población inicial. Los informáticos agrupan estos programas bajo la etiqueta «Descubrimiento del Conocimiento», y definen su funcionamiento como «la extracción no trivial de la información implícita, previamente no conocida y potencialmente útil» (Frawley et al. 1991: 3). La definición de conocimiento que se deduce es la siguiente: «una expresión que tiene interés (de acuerdo con cierto criterio de evaluación establecido por el usuario) y que es lo suficientemente cierta (también de acuerdo con un criterio determinado)».

Ahora bien, al final del capítulo anterior, se afirmaba que la calidad del esquema de asociaciones y co-ocurrencias descubierto en los datos dependía de la homogeneidad y adecuación de la base de datos al pro-

blema en cuestión. Según esto, ¿hasta qué punto las regularidades encontradas son significativas? En otras palabras, ¿son esos programas realmente capaces de descubrir *unidades de conocimiento*? La respuesta a esa pregunta debe ser negativa: los programas de descubrimiento automático, al menos tal y como los hemos descrito en el capítulo anterior, son incapaces de «descubrir» unidades de conocimiento, puesto que son incapaces de evaluar la utilidad o relevancia de sus resultados.

Tanto la «valididad» como la «utilidad» son cuestiones relativas, que no dependen de los datos ni son generadas por el algoritmo. Ambas forman parte de lo que se denomina conocimiento del entorno del problema, y hacen referencia al contexto específico en el que tiene lugar la inducción y en el que se utilizarán los resultados. Precisamente porque son «universales» los programas descritos en el capítulo anterior no incluyen ninguna referencia al contexto de la inducción, por lo que sus resultados no pueden conceptualizarse como «unidades de conocimiento». Debido a esta incapacidad o inadecuación, la función de programas de descubrimiento al estilo de I.X.L. o los basados en el algoritmo ID3 y semejantes debiera reducirse a la búsqueda de la representación interna de la clase o concepto al que pertenece una instancia del mismo. Para que sus resultados, expresiones del tipo

DATO → INSTANCIA → CLASE

adquieran el rango de «conocimiento» es preciso que las clases resultantes estén organizadas en un sistema coherente, esto es, debe existir una relación no trivial entre los resultados y los objetivos.

Por otra parte, los programas de descubrimiento no reproducen adecuadamente el razonamiento científico. Para ello, debieran ser capaces de *aprender conceptos* y no tan sólo señalar la existencia de co-ocurrencias. En este caso, la noción de «aprendizaje» estaría definida como «aquel fenómeno exhibido cuando un sistema (agente humano o máquina) mejora en la ejecución de una tarea concreta, sin necesidad de haber sido reprogramado». Ahora bien, los conceptos científicos no pueden ser generados algorítmicamente; un científico no analiza la realidad aplicando fórmulas mágicas a sus datos y usando el resultado de esas fórmulas como conceptos o leyes absolutas, sino que *modifica* progresivamente la formulación de un concepto inicial (hipótesis), acercándose paulatinamente a un concepto «mejor» que el de partida. «Mejor» puede significar cosas distintas: un mayor porcentaje de soluciones correctas, una respuesta más rápida, soluciones más prácticas o eficaces, o con un mayor abanico de posibilidades de aplicación, etc. Es precisamente en este punto donde radica la necesidad de que el concepto o ley inducida sea «útil».

El aprendizaje de una categoría suele definirse en Psicología Cognitiva en tanto que *transferencia* de conocimiento de un conjunto de ejemplos a una expresión lógica. Sin embargo, no toda transferencia de conocimiento permite aprender; es preciso que el conocimiento de la segunda unidad *aumente* como consecuencia de esa transferencia. Pero no hay aprendizaje, no hay «incremento» del contenido, si no se definen previamente los parámetros que permitan establecer la «mejora» del sistema. Aquí radica el *problema de la inducción*, que durante siglos se han planteado los filósofos: a diferencia de lo que sucede con la deducción, no puede probarse si las leyes o entidades generales resultantes de una inferencia inductiva (p.e. una generalización) son correctas. Mientras que la deducción es un tipo de razonamiento finito, es decir, concluye tras un número determinado de pasos, la inducción no concluye nunca, porque no puede establecerse con seguridad cual de sus resultados es el definitivo. De este modo, la creación inductiva de conceptos científicos es en realidad un proceso continuo, que no puede reducirse a la aplicación de un único operador. Lo realmente importante en una inferencia inductiva no es la naturaleza de los operadores (estadísticos o lógicos), sino las heurísticas de procesamiento que asegurarán que las conclusiones inductivas son plausibles y relevantes para el objetivo que se está procesando.

En definitiva, el control de la inducción tiene más importancia que la inducción en sí misma. En Inteligencia Artificial suelen utilizarse para ello dos estrategias distintas: introducir estructuras de control en el algoritmo, o bien controlar la base de datos. El «error» en la aplicación de Fernández y García (1991) descrita en el capítulo anterior, estriba en no haber controlado ni lo uno ni lo otro: el algoritmo ID3 no incorpora estructuras de control efectivas, ese es su mayor defecto, reconocido incluso por los autores de la aplicación. En esas circunstancias, el tamaño y complejidad del árbol de decisiones resultante puede reducirse controlando el tamaño y la diversidad de la base de datos. A diferencia de los procedimientos estadísticos, en los que priman muestras grandes que preserven la Ley Normal, en la inducción automática lo importante es la «calidad» de los datos: la cantidad de ejemplos positivos y negativos ha de estar equilibrada, y al mismo tiempo esos ejemplos han de ser exhaustivos; un mayor número de ejemplos no mejorará la interpretabilidad de los resultados, sino que aña-

dirá redundancias y aumentará peligrosamente la complejidad del árbol de decisiones al incorporar rasgos peculiares a cada uno de los ejemplos particulares. Por consiguiente, cuanto mejor («más específico») y más apropiados sean los datos iniciales, más útil y aprovechable será el árbol de decisiones. Obviamente, el «control de los datos iniciales» nos introduce de lleno en el tema de la experimentación científica, que suele definirse en tanto que *creación controlada de resultados observables*. (cf. por ejemplo Bunge 1983).

La segunda opción para controlar la cantidad, complejidad y utilidad práctica de las inducciones automáticas consiste en la programación de rutinas especializadas en el control de las inferencias. Parcialmente, ése es el enfoque adoptado en todos los programas que derivan de una manera u otra del método de Michalski. En COCLUSI<sup>H</sup> (Grey 1991), la inducción se controla por medio de la evaluación post-hoc de los resultados del programa. Los criterios de evaluación son los siguientes:

- estructurar al máximo la información contenida en los objetos
- tener un poder de predicción aceptable
- reflejar una relación: la pertinencia de una propiedad varía según la relación que se haya establecido entre las hachas que constituyen un grupo.

En el primer caso, el programa detendrá los operadores de generalización-especialización cuando en la clase ya no sean posibles más operaciones, o bien cuando la cantidad de objetos con los que la descripción resultante es compatible sea inferior al límite definido por el usuario del programa. El segundo criterio hace referencia a la maximización de las correlaciones en el interior de una clase y a la minimización de las correlaciones entre clases. La calidad de una clasificación se mide entonces usando la fórmula:

$$\text{Calidad_Clasi} = \Sigma \text{ cohesión (sc)} * \text{contraste (sc)}$$

La cohesión de una clase es mayor cuanto mayores sean las similaridades entre los objetos que pertenecen a la clase. El contraste evalúa las disimilaridades entre los objetos de una clase y los de otra. La «calidad» de una clasificación será mayor cuanto mayor sea el resultado de esa ecuación.

Hay que limitar la búsqueda de generalizaciones y/o asociaciones, aun con el riesgo de omitir informaciones importantes. No puede haber aprendizaje si no se controla de un modo u otro el mecanismo de inducción, ya se trate de la generalización o de la correlación de propiedades. Cuanto más estrictas sean las condiciones que impongamos al mecanismo, mayor riesgo de eliminar generalizaciones interesantes; si esas condiciones fuesen demasiado débiles, las dimensiones del espacio de búsqueda serán excesivas para encontrar en él hipótesis satisfactorias.

De este modo, en el dominio de la Inteligencia Artificial, las inferencias inductivas se caracterizan de un modo bastante distinto al de la inferencia estadística clásica: el objetivo no es distinguir qué atributos o variables están correlacionados con qué objetos en una base de datos, sino en modificar progresivamente la definición de un objeto paradigmático de modo que la definición final sea utilizable para todas las instancias posibles del mismo objeto. Se trata de un tipo de inferencia que ejecutaremos en respuesta a ciertos objetivos específicos, tales como la búsqueda de una explicación plausible para un resultado inesperado. *El estudio de la inducción es, pues, el estudio de cómo se modifica el conocimiento a través de su uso.* Por consiguiente, debemos programar sistemas informáticos capaces de generar y revisar las unidades de conocimiento que sirven para construir las Teorías Científicas. Esos sistemas deben llevar a cabo tres tareas interrelacionadas:

- 1 evaluar las reglas del sistema como instrumentos para alcanzar los objetivos, mejorándolas cuando sea posible y favoreciendo la aplicación de las mejores.
- 2 generar plausiblemente nuevas reglas útiles que sean capaces de extraer y explotar las regularidades que pudiera haber en la experiencia
- 3 proporcionar asociaciones y agrupaciones entre reglas para crear estructuras de conocimiento progresivamente mayores que conduzcan a una modelización efectiva del entorno.

La ejecución de ese programa estará dirigida por el mecanismo automático de resolución del problema, basado a su vez en la retro-alimentación (*feedback*) concerniente al éxito o fracaso de las predicciones generadas por el sistema. R. Forsyth (1989) ha propuesto un esquema general para explicar la estructura de ese programa. Se compone de los siguientes módulos:

El **Crítico** compara el resultado obtenido con el resultado que se esperaba obtener. Para ello debe existir algo así como un 'sistema ideal' que sirva para medir la conducta del sistema. En la práctica, se trata de la

figura del experto humano o del profesor, que identifica los errores cometidos por el agente que está resolviendo el problema. De su actuación se deriva la importancia de la *retro-alimentación evaluativa*, cuya función consiste, precisamente en identificar errores y relanzar el mecanismo de solución partiendo de nuevas premisas (definidas, precisamente, por el error que se ha cometido).

El **mecanismo de aprendizaje** constituye el núcleo básico del sistema. Es la parte del sistema que tiene la responsabilidad de corregir la base de conocimientos para impedir la obtención de resultados erróneos.

El **módulo ejecutivo** es el que lleva a cabo la acción (validar los objetivos). No se limita a hacer las veces del «motor de inferencias», sino que funciona como «intérprete» de las instrucciones procedurales almacenadas en la base de reglas. Cuando las reglas experimentan modificaciones, la conducta general del sistema cambia, debido a la conexión entre la base de reglas y el módulo ejecutivo.

En las páginas siguientes, cuando hablemos de inducción automática nos referiremos siempre a la modificación de las reglas existentes en una base de conocimiento y a la generación de nuevas reglas. La **modificación** adoptará la forma de revisión de parámetros tales como el factor de certidumbre de una regla, la enumeración de sus condiciones de activación,... La **generación** de nuevas reglas se caracterizará por la generalización o especificación de las ya existentes. Ambos procedimientos conducen a la **recombinación** del conocimiento previo. Precisamente esa recombinación será la que nos permita representar informáticamente el aprendizaje inductivo. Obviamente, cualquier procedimiento de recombinación de los elementos constitutivos del conocimiento depende de las técnicas empleadas para identificar esos elementos constitutivos. También se usa conocimiento heurístico para guiar la modificación y recombinación de reglas; por ejemplo: los rasgos presentes en las reglas que demuestran un alto poder predictivo son elegidos como candidatos para incluirlos en la condición o acción de nuevas reglas.

En las páginas siguientes se presentarán distintos programas de ordenador que intentan responder a las dificultades derivadas del uso indiscriminado de los algoritmos universales de generalización automática. En otras palabras, programas que pretenden acomodar de una manera u otra el principio de Occam: *entia non sunt multiplicanda praeter necessitatem*. El lector puede pensar que no son programas útiles, en el sentido en que el arqueólogo no puede emplearlos tal cual para facilitarle la labor de investigación. Antes al contrario, se trata simular, reproducir por medio del ordenador, la actividad de un investigador en distintas disciplinas; de ahí que esos programas no superen el nivel de la «experimentación». Su función es, pura y simplemente, poner de manifiesto cómo debe utilizarse el razonamiento inductivo estándar en la investigación científica. Como la Ciencia no puede reducirse a la Inducción, esos programas son limitados, pero tan sólo porque son específicos. No pretenden reducir el método científico a una mecánica simple, sino que ponen de manifiesto uno de los muchos mecanismos de razonamiento posibles.

Los primeros intentos en esta línea (simulación informática del razonamiento científico) fueron los de Lenat (1979), si bien puede decirse que la Filosofía Computacional de la Ciencia empieza con las sucesivas versiones del programa BACON (Langley 1979, Langley et al. 1987), sobre la base de los trabajos teóricos de H.A. Simon (1973, 1977). Estos programas fueron diseñados específicamente para estudiar la lógica del descubrimiento científico, reproduciendo experimentos clásicos de la Historia de la Astronomía, de la Física y de la Química. Dado que nunca se ha intentado algo semejante en Arqueología, la explicación que sigue quizás pueda parecer un tanto ajena al lector, aunque no por ello pierde interés.

Todos los programas de la familia BACON consideran que el descubrimiento de leyes científicas en una población de datos es formalmente idéntica a la resolución de problemas por medio de algoritmos de búsqueda heurística. La misión de BACON y sus versiones sucesivas (de BACON.1 a BACON.6) es analizar la inducción de leyes cuantitativas que describan una población de datos numéricos, distinguiendo propiedades intrínsecas y prestando atención al ruido o error aleatorio que pudiese existir. Las leyes inducidas o «descubiertas» ponen en relación diversos términos. Precisamente, el objetivo de los programas es crear esos términos y encontrar alguna relación matemática que describa adecuadamente su relación mutua. Por ejemplo, supongamos que los datos se refieren a una secuencia de signos:

ABCMIDFMGHIJKLMNNM...

El objetivo del programa es definir la ley o proceso que ha dado lugar a esa secuencia: una lista alfabetica normal interrumpida cada tres signos por la letra M. Arqueológicamente hablando, supongamos una secuencia de cerámicas, de buriles, o de cualquier otro artefacto, descritos por medio de variables morfométricas; el objetivo del programa sería encontrar el «sentido» de la secuencia: una seriación cronológica, por ejemplo, ex-

sándola por medio de una ecuación matemática. Ni que decir tiene que el programa ha de ser lo suficientemente sofisticado para saber si la ley o principio identificado se ajusta a esa secuencia, o bien, la secuencia misma es meramente aleatoria y no tiene sentido en ninguna Teoría Científica dada. Por ese motivo, el criterio de validez de la ley descubierta es, simplemente, su ajuste a los datos iniciales, ajuste definido por medio de ciertos criterios heurísticos de aproximación incorporados al programa como parámetros.

Desde el punto de vista de su implementación informática, BACON y sus derivados son *programas declarativos*, esto es, se componen exclusivamente de reglas de producción. Las condiciones de esas reglas describen los objetivos del sistema o los esquemas que pudieran aparecer en los datos; sus acciones respectivas son las que contienen los operadores de creación y modificación de nuevos términos.

La primera versión del programa, BACON.1, es capaz de descubrir una ley que describa la relación entre dos términos cuantitativos. Una ley arqueológica de este tipo es la ley del decrecimiento monótono (Renfrew 1977) que relaciona dos términos cuantitativos, la DISTANCIA, y la CANTIDAD DE MATERIA PRIMA por medio de una función monótona decreciente, cuya ecuación es:

$$I = a - b D + e$$

Que la formulación de la ley del decrecimiento monótono en arqueología presente innumerables dificultades y haya dado pie a bien fundadas críticas (cf. Djindjian 1991) es otra cuestión, que no nos interesa en este momento. Consideremos tan sólo que se trata de un ejemplo arqueológico de ecuación simple.

El programa contiene diecisésis reglas de producción agrupadas en cuatro bloques. El primero de ellos se encarga de disponer los datos de una forma específica: inicializa los términos que se emplearán en la definición de la ley. El segundo bloque («Generador de Regularidades») examina los pares de términos creados por el bloque de reglas inicial y comprueba la presencia o ausencia de ciertos tipos de regularidades:

- *linealidad* : si los términos están relacionados proporcionalmente
- *incremento* : si las magnitudes de ambos términos experimentan un crecimiento monótono
- *decrecimiento* : si las magnitudes de uno de los términos experimenta un crecimiento monótono, mientras que los valores del otro término decrecen
- *constancia* : si un término resulta (aproximadamente) constante.

Según sea el tipo de regularidad encontrado, el tercer bloque de reglas define nuevos términos teóricos (nuevos conceptos) y calcula sus valores a partir de los términos iniciales. Por ejemplo, por medio de reglas de producción como éstas:

SI	se buscan leyes
y	se dispone de un conjunto de valores para el término X
y	se dispone de un conjunto de valores para el término Y
y	los valores absolutos de X experimentan un incremento cuando los valores absolutos de Y también lo experimentan
y	esos valores no están relacionados linealmente
ENTONCES	considerar el cociente de X e Y.
SI	se buscan leyes
y	se dispone de un conjunto de valores para el término X
y	se dispone de un conjunto de valores para el término Y
y	los valores absolutos de X experimentan un incremento cuando los valores absolutos de Y experimentan un descenso
y	esos valores no están relacionados linealmente
ENTONCES	considerar el producto de X e Y.

Según esto, el resultado de los operadores COCIENTE y PRODUCTO es un nuevo término, susceptible de aparecer en la ley cuantitativa que se desea inducir. La regla correspondiente se aplica inmediatamente después de una de las anteriores:

SI	se ha de <i>considerar</i> el producto (o el cociente) de $X$ e $Y$
y	la definición de $X$ es $D_X$
y	la definición de $Y$ es $D_Y$
y	no existe ningún término cuya definición equivalga a $XY$ (ó $X/Y$ )
ENTONCES	<i>define</i> el término $T$ como $D_X D_Y$ (ó $D_X / D_Y$ ).
y	<i>calcula</i> los valores de $T$ (iniciándolo desde 1).
SI	se han de <i>calcular</i> los valores del término $T$
y	la definición de $T$ es $D_T$
y	$A^p$ es uno de los componentes de $D_T$
y	los valores de $A$ son $a_1, \dots, a_k$ ,
ENTONCES	<i>multiplicar</i> los valores actuales de $T$ por $a_1^p, \dots, a_k^p$

Cuando como resultado de la aplicación de las reglas anteriores se obtenga un término nuevo que tenga valores constantes o casi constantes, se detendrá el proceso de inducción. Precisamente esta búsqueda de constancia en los términos o conceptos inducidos es lo que permite reducir la influencia del ruido o redundancia que pudieran existir en los datos iniciales.

En definitiva, BACON.1 no busca el conjunto de asociaciones que pudieran existir entre los individuos analizados, sino que su función es generar un concepto nuevo a partir de un número limitado de regularidades hipotéticas predefinidas, siempre y cuando los individuos y las variables iniciales cumplan ciertas condiciones, a las que denominamos heurísticas. El concepto a generar tiene una forma predefinida, lo que obliga a que su inducción sea el resultado de una serie de operaciones sucesivas, que incluyen, además, la creación de sub-términos o sub-conceptos necesarios para la formación del concepto. De este modo, la creación de conceptos en BACON adopta la forma de una progresiva combinación de propiedades, entre las que cabe distinguir entre propiedades independientes y propiedades dependientes. Las primeras son aquellas definidas al principio y que caracterizan la Base de datos (por ejemplo, DISTANCIA, INTERACCIÓN); las propiedades dependientes son aquellas que implican una combinación de las anteriores (por ejemplo  $D/I$ ,  $D^2/I$ ,  $D^3/I^2$ ).

El mecanismo es exactamente el mismo que el presentado en el primer capítulo y que hacía referencia a la descomposición de los problemas científicos: el objetivo (estado final del problema) es resolver la ecuación:

$$X_i^m = Y_i^n$$

El estado inicial del problema está constituido por el listado de valores de las variables **X** e **Y**. Los estados intermedios del mismo problema están configurados por los distintos cálculos que hay que realizar para despejar las incógnitas en la ecuación. En muchos casos, esos estados intermedios adoptan la apariencia de conceptos, por ejemplo: el *cociente*  $X_i / Y_i$ , el *producto*  $X_i * Y_i$ . El problema de la inducción, esto es, el criterio utilizado para detener la ejecución de los algoritmos, es también heurístico: la existencia de términos constantes.

La estructura de control en BACON es, por tanto, un sistema de producciones o programa declarativo, en donde todas las reglas aplicables a una situación dada son activadas, pero sólo una es ejecutada. La acción de esta regla induce ciertos cambios en la memoria de trabajo, esto es, en la configuración inicial de variables (por ejemplo, calcular los valores de la propiedad dependiente  $X_i * Y_i$ ), o bien crear una nueva propiedad dependiente ( $X_i / Y_i$  a partir de  $X$  y de  $Y$ ). Una vez transformado el estado anterior del problema, el proceso de activación empieza otra vez, ejecutándose nuevas reglas que se ajusten a la situación actual.

La mayoría de versiones restantes de BACON son extensiones de este esquema. Así, BACON.2 es capaz de definir leyes polinómicas, más complejas que las encontradas por BACON.1, al incorporar las regularidades secuenciales a los tipos de regularidad que debe buscar. BACON.3 acepta datos múltiples y no tan sólo pares de valores, incorporando a su vez diversas heurísticas que controlan la búsqueda de forma sistemática. BACON.4 es capaz de identificar propiedades intrínsecas asociadas a términos nominales. BACON.5 detecta la simetría de las leyes producidas, lo que reduce drásticamente la búsqueda de soluciones idóneas. Otros programas estrechamente relacionados son STAHL, GLAUBER y DALTON, capaces de analizar datos cualitativos y de inducir expresiones lógicas (a diferencia de BACON, que sólo genera ecuaciones a partir de datos cuantitativos).

BACON y sus sucesores no están basados en una concepción inocente del método inductivo. De su funcionamiento se deriva una concepción muy distinta de la filosofía inductiva y de su relevancia en la investigación.

ción científica, muy en línea con la revisión de la filosofía de Francis Bacon (1561-1626) emprendida por Peter Urbach (Urbach 1982, 1987, Gibbins 1990). Según este autor, las críticas vertidas a Bacon (el filósofo y no el programa) son infundadas. El Francis Bacon descrito por Urbach se parece curiosamente a Popper: aunque Bacon oponía la *interpretatio naturae* a la *anticipatio mentis*, nunca se opuso al razonamiento por medio de hipótesis, sino que se oponía al predominio de las hipótesis sobre la investigación empírica. En otras palabras: el razonamiento por hipótesis no debía sustituir al análisis empírico, lo que nosotros podemos caracterizar como búsqueda de regularidades o definición del esquema de co-ocurrencias existente en una base de datos.

BACON (el programa) pretende integrar la «interpretación de la naturaleza» con la «anticipación de la mente» que Bacon (el filósofo) diferenciaba. De lo que se trata es de implementar un razonamiento *hipotético-inductivo*, capaz de plantear conjeturas y refutaciones. Sus conceptos -expresados bajo la forma de ecuaciones- son generados gracias a un mecanismo explícito (conjetura) y a continuación refutados con ayuda de los datos iniciales. Lo que hace que el programa parezca un mecanismo inductivo simple es la manera en que se conjecturan los conceptos: no como hipótesis deductivas, sino por medio de heurísticas específicas. La característica anti-baconiana presente en esos programas es la prioridad dada a la «anticipatio». BACON ha sido programado junto con una teoría específica -que una ecuación de la forma  $X^n Y^n = constante$  es válida-. ¿Por qué este formato es más adecuado que otro? El programa no puede dar una razón, ya que esa «anticipatio» es totalmente externa a la «interpretatio» buscada. Ahora bien, «externa» no quiere decir «ajena»: esas «anticipaciones» proceden del conocimiento que tiene el programador acerca del dominio al que se refieren los datos, y son utilizadas por el programa como estructuras de control y asistencia para poder desarrollar una investigación «empírica». El programa es lo suficientemente fiable como para tener razones para sostener que el concepto así generado es «real», y no fruto de un capricho de la máquina.

La implementación de los programas de la familia BACON presenta bastantes limitaciones, muchas de ellas mencionadas por sus propios diseñadores (Langley et al. 1987) y por otros autores (Walker 1987, Haase 1989, Cheng 1990). Algunos de estos problemas han sido abordados en uno de los sucesores de BACON, el programa FAHRENHEIT (Kochin y Zytkow 1986, Zytkow 1987). Este programa utiliza BACON.3 como subprograma, de manera que es capaz de definir las mismas leyes que BACON, pero con una particularidad: es capaz de definir el rango de valores en los que la ley es válida (en términos de las variables que se refieren a la ley, claro está). Así, en el caso de un ejemplo arqueológico, una ley o modelo general no tiene por qué ser aplicable a todos los artefactos hallados en un yacimiento o en un sector del yacimiento, sino que su ámbito de aplicabilidad se reduce a un tipo determinado. La limitación del programa es que ese tipo de artefactos para los que es aplicable ha de definirse en términos de las variables que definen la ley (por ejemplo, longitud, diámetro máximo, peso,...) lo cual no siempre es evidente. FAHRENHEIT constituye un avance con respecto a la familia de programas BACON ya que no sólo es capaz de encontrar leyes, sino que también puede especificar dónde es válida esa ley. No obstante, esta capacidad no debiera confundirse con la identificación de los «valores de verdad» de la ley. La descripción del espacio de aplicabilidad de una ley caracteriza tan sólo otro de los aspectos de ese modelo, y no la justifica.

Muchos de los límites y deficiencias de BACON, sin embargo, no son imputables a su implementación, sino al tipo de razonamiento que pretenden reproducir, lo que Gibbins (1990) ha denominado Baconianismo Computacional: cómo asegurarse la relevancia de las variables independientes que se utilizan en la definición del término o concepto a inferir.

## 7. ¿DEBEMOS AUTOMATIZAR EL RAZONAMIENTO ARQUEOLÓGICO?

### Inteligencia Artificial y Práctica Arqueológica

Aunque la utilización de las técnicas de Inteligencia Artificial no está muy difundida en la Arqueología actual, podemos mencionar algunos ámbitos de nuestra disciplina que han de beneficiarse, necesariamente, con los actuales avances en tecnologías de programación de ordenadores:

- Los análisis estadísticos (uni- y multidimensionales) no tardarán en ser sustituidos por las técnicas de Inducción Automática, utilizadas en la actualidad como útil exploratorio en grandes bases de datos. Ventajas: facilidad de uso, sencillez para interpretar el resultado del análisis (Reglas vs. Ecuaciones), ausencia de parametrización y no dependencia con la Curva Normal
- Tipologías Automáticas: El uso de Sistemas Expertos constituye la alternativa más evidente a las listas tipológicas, ya que permiten acumular más datos, su lectura no es ambigua, y gestionan incluso la posibilidad de excepciones
- Tratamiento de imágenes. En Arqueología todo es imagen, desde la microfotografía del desgrasante de una cerámica hasta la planimetría de una región, pasando por la forma y decoración de los artefactos, la disposición micro-espacial del registro espacial y las secciones estratigráficas. Hasta ahora esas imágenes tenían que traducirse en un lenguaje descriptivo que en la mayoría de los casos demostraba sus limitaciones debido a su dependencia con respecto a la subjetividad del que hacía la descripción. El uso de Redes Neuronales y Sistemas Distribuidos permitirá, a partir de ahora, el análisis directo de las imágenes, sin tener que pasar por la fase de la codificación descriptiva en variables y atributos.

No obstante estos avances «reales» en el ámbito *práctico*, en este libro se han presentado algunos proyectos recientes que intentan ir más allá, pretendiendo *simular* la manera en que piensan los arqueólogos. El propósito de estos programas es estudiar los mecanismos cognitivos subyacentes en el razonamiento arqueológico, *visualizando* el razonamiento en sí. Cada uno de estos proyectos se inscribe de lleno en lo que podríamos denominar *Arqueología Automática*: una subdisciplina encargada de poner de manifiesto la manera en que los arqueólogos interpretan sus datos. El razonamiento científico es un mecanismo inobservable, del cual sólo conocemos sus efectos (las interpretaciones). Por consiguiente, un método de *visualizar* el mecanismo de producción de las interpretaciones debiera ser bienvenido. La Arqueología Automática utiliza medios computacionales para cumplir ese objetivo. Su supuesto básico es el siguiente:

si somos capaces de programar un ordenador *como si* fuese una persona, empezaremos a comprender cómo actúa esa persona.

Algunos pueden pensar, sin embargo, que un programa «simulador» funciona de manera sensiblemente distinta a la del arqueólogo. La diferencia, sin embargo, ha de entenderse en los términos de una Teoría de la Representación: el programa informático hace las veces de *modelo* o *representación* del mecanismo de producción de interpretaciones arqueológicas. Ese modelo no tiene por qué ser *idéntico* a su objeto de estudio, sino que existe una determinada relación de correspondencia entre lo representado y la representación. Las características y propiedades de esa correspondencia dependen de los propósitos para los que se haya creado dicho modelo.

A continuación, y como conclusión a este libro, recapitularemos la manera de representar el mecanismo de inferencia arqueológica por medio de técnicas de programación de ordenadores.

## La Naturaleza de los Problemas Arqueológicos

A lo largo de este libro se ha mantenido como axioma fundamental que *toda interpretación, todo significado, no son más que la solución a un determinado problema*. Sólo hay un modo de resolver este tipo de problemas: disponiendo de un conjunto de soluciones posibles alternativas (p.e. cronologías que puede tener un objeto arqueológico) y decidiendo cuál de ellas es la más apropiada en el caso en cuestión, según ciertos criterios bien especificados. Este conjunto de soluciones suele denominarse *generador* o *espacio* del problema y representa una Teoría Científica.

El *estado inicial* del problema coincide con lo que los lógicos denominan *explanans*, esto es, aquello que queremos interpretar. El estado inicial de un problema arqueológico es obvio: el registro arqueológico, el artefacto o la disposición de artefactos que queremos interpretar, y lo representaremos por medio de una lista de rasgos descriptivos. La *interpretación* arqueológica coincide, obviamente, con la *solución* al problema arqueológico que hemos planteado. En este caso, hacemos referencia al *explanandum*, que denominamos *estado final* del problema. Cuanto más generales sean los términos que describen esa solución, tanto más amplio será su ámbito de aplicación, y responderá a más problemas.

Ahora bien, para que una unidad de conocimiento así caracterizada sea realmente una solución precisamos de un conjunto de criterios de validación o requisitos para aceptarla como válida; ese conjunto de requisitos constituye el *objetivo*, que puede ser definido en términos sencillos como: «aquel que desea obtenerse». Necesitamos, además, de un conjunto de *operadores*, definidos, a su vez, como «unidades de conocimiento necesarias para alcanzar un fin». Nos referiremos, por tanto, a un objetivo cuando detectemos una «falta de información» para obtener algo; por su parte, los operadores determinarán las maneras de «usar» cierta información con el fin de llenar ese vacío de conocimiento.

Aceptando la identidad del mecanismo de producción de interpretaciones con el mecanismo de resolución de problemas, afirmaremos que los datos arqueológicos se interpretan por medio de una operación de *búsqueda*: tanto el arqueólogo como el programa de ordenador que resuelve automáticamente problemas, son capaces de *buscar la interpretación o estado final que corresponde a la descripción empírica o estado inicial facilitado por el usuario*. Según el tipo de problema, la operación de búsqueda será distinta:

- *Problemas de Diagnóstico* (cf. Torasso y Console 1989): se parte de un conjunto de interpretaciones alternativas totalmente conocido y se trata de seleccionar una de ellas, utilizando para ello diversos criterios que reduzcan la búsqueda a unos límites computables.
- *Problemas de Diseño* (cf. Brown y Chandrasekaran 1989): se trata de construir la interpretación de ciertos datos a partir de una serie de restricciones que deben ser satisfechas. El conjunto de interpretaciones posibles es, en este caso, desconocido, y hay que generarla basándose en las restricciones. La operación de búsqueda, por tanto, no se realiza entre las interpretaciones candidatas, sino entre los operadores -cuál es el más conveniente para transformar el estado actual en un estado intermedio que cumpla los requisitos-.

Así pues, computacionalmente hablando, la interpretación de un artefacto arqueológico empezará siempre con la identificación de la dificultad que debe resolverse (el *objetivo*). Una vez definido el objetivo recurrimos a la información disponible (conocimiento previo) y construimos el estado inicial sobre el cual aplicaremos el operador o los operadores necesarios. Recuérdese que la *descripción* no es una operación neutra, sino que depende de la teoría a nuestra disposición, además de los objetivos que hayamos formulado.

A continuación estimamos la diferencia existente entre nuestro punto de partida y el punto al que queremos llegar (el *objetivo*) analizando las relaciones de similaridad existentes entre el estado inicial y el estado final del problema. Dado que ambos son unidades de conocimiento representadas de la misma manera, esa estimación, aunque compleja debido a las distintas modalidades, es computacionalmente factible. El procedimiento u operador que seleccionemos para *generar* la interpretación tendrá que reducir esa diferencia, permitiéndonos ir del punto de partida hasta el punto de llegada interpretación.

Tanto el estado inicial como el estado final de ese proceso han sido *representados* gracias al conocimiento previo que disponemos. Sin ese conocimiento, la representación no hubiera sido posible y la interpretación no habría tenido lugar. Este principio equivale a decir que *sin una Teoría Científica no puede existir interpretación arqueológica*. Por consiguiente, necesitamos conocimiento para definir la situación en la que se ha detectado una dificultad, y necesitamos conocimiento para resolver esa dificultad. El estado inicial está de un modo

u otro implícito en el enunciado del problema, es decir, aparece determinado por el objetivo a resolver y por el conocimiento previo que se dispone; «representar» la situación inicial, por tanto, equivale a *describirla*. Representar el resultado, sin embargo, exige construir un conjunto de soluciones posibles a partir de un conjunto de conocimiento previo. Gran parte de la eficiencia del procedimiento de resolución estará basada en la «construcción» de ese conjunto, es decir, en el isomorfismo entre las soluciones posibles y el conocimiento del cual proceden.

Dado que «interpretamos» por medio de la aplicación de ciertos cálculos lógicos (los *operadores*) concluirímos que la interpretación de un artefacto arqueológico debiera ser una *función* de su descripción empírica. La demostración de este principio es sencilla, ya que aparece como una consecuencia de la caracterización de los mecanismos de resolución de problemas en términos de búsqueda y selección de la mejor solución posible: dada una dificultad concreta (objetivo), la solución *dependerá* de la situación en la que se haya identificado esa dificultad. Por tanto, la solución concreta que obtengamos habrá de estar determinada por el estado inicial que hayamos empleado como punto de arranque del procedimiento de solución.

Concluirímos, entonces, que la relación existente entre estado inicial y estado final o solución coincide con la aplicación de un operador. En realidad, ese operador se limita a *representarla* función entre ambos, traduciéndola a un formato ejecutable, esto es, como una secuencia ordenada de reglas, acciones e instrucciones. Esta relación (y el operador que la representa) es un elemento de conocimiento científico, que forma parte del conjunto de conocimiento en el que se ha enunciado el problema, el estado inicial y el conjunto de soluciones posibles.

Ahora bien, la relación entre un dato arqueológico y su interpretación, no tiene por qué ser una relación formal o matemática. Existe una gran diversidad de funciones o relaciones posibles entre el estado inicial y el estado final de un problema. No olvidemos que los problemas surgen de la identificación de cierta dificultad; por lo tanto, la relación entre un estado inicial y un estado final no es nunca obvia, sino no se llamaría dificultad. Precisamente el punto clave en la resolución de un problema radica en determinar si existe o no alguna relación entre los datos iniciales y la solución propuesta.

En la vida cotidiana, pocas veces nos molestamos en definir esa relación, pues actuamos por ensayo y error o bien recurrimos a nuestra experiencia: si la situación es semejante a alguna antes experimentada, actuaremos de igual forma que lo hicimos en aquel momento. La interpretación arqueológica también está basada muchas veces en el ensayo y error, aunque lo más aconsejable sería establecer la relación entre datos iniciales y concepto interpretativo experimentalmente. Una vez que se ha obtenido una lista de posibles operadores, se procede de nuevo por búsqueda y selección del más idóneo entre ellos, esto es, de aquel que mejor reproduzca la relación existente entre la descripción de unos artefactos y la interpretación más verosímil de los mismos. En definitiva, para poder buscar y seleccionar una interpretación, es preciso que antes busquemos y seleccionemos un operador, operación para la que necesitaremos, también, algún tipo de conocimiento previo que nos permita realizar esa selección con ciertas garantías de éxito. No hay manera de construir un operador que busque en un conjunto infinito de soluciones o interpretaciones posibles, porque esa búsqueda no acabaría nunca. En Inteligencia Artificial denominamos a esa dificultad *explosión combinatoria*, y suele resolverse recurriendo a una serie de condiciones o *heurísticas* que progresivamente limitarán la cantidad de operadores posibles y que, finalmente, permitirán elegir el más adecuado. El procedimiento es aparentemente simple: en cada una de las etapas de la resolución se compara el estado actual del problema con el estado final del mismo (si es conocido) o con ciertas condiciones previamente enunciadas que debe cumplir ese estado para poder ser considerado la solución; a continuación se lleva a cabo una operación que permite reducir la diferencia entre el estado actual y las características del estado final. El uso de este procedimiento heurístico permite al agente encargado de resolver el problema avanzar reduciendo progresivamente la distancia a la solución y evitando el uso de estados del problema que aumenten esa diferencia.

Una búsqueda heurística es un procedimiento que aplica pasos plausibles para navegar en un espacio de alternativas inmenso, con el fin de conocer la mejor (o una colección de las mejores) alternativas para cierto propósito. Lo que convierte a esta búsqueda en *heurística* es que el procedimiento no garantiza que el resultado sea la mejor de todas las alternativas, o una colección que incluya la mejor alternativa, si bien el procedimiento se aproximará bastante a ella, de acuerdo con cierto criterio. Ese criterio puede que no sea muy riguroso y que no siempre proporcione una solución óptima, si bien ésta será la mejor que se puede obtener, muy superior a la que se habría logrado sin una búsqueda heurística. Se insiste en aquello que es alcanzable y suficientemente bueno, prescindiendo de lo que es óptimo, pero inalcanzable.

La idea de *heurística* aplicada al razonamiento científico no tiene nada de insólito: ciertas HIPÓTESIS VEROSÍMILES organizan el proceso de resolución de un problema científico, reduciendo el espacio de dicho problema

El operador abductivo por excelencia es la *analogía*, aunque los lógicos señalan también el interés de los siguientes: igualdad, simetría, homología, desigualdad y referencia. Así, dos estados de un problema estarán asociados:

- si puede establecerse alguna correspondencia heurística entre ambos,
- si son semejantes,
- si un estado intermedio hace referencia a otro en su representación.

Más de un lector preguntará por qué son tan débiles, formalmente hablando, las asociaciones que configuran una cadena de inferencias. La razón estriba en las características del mecanismo *heurístico* de producción de interpretaciones desarrollado en la sección anterior. La operación de búsqueda de la «mejor solución posible» no puede implementarse mediante un algoritmo de tipo polinomial, es decir, *no existe ningún procedimiento que nos permita averiguar, en un número finito de pasos, la idoneidad de una interpretación dada cierta información empírica inicial*.

Tanto los especialistas en lógica como en informática le han dado muchas vueltas a este corolario de los Teoremas de Gödel y de Church, y han llegado a la conclusión que no hay una respuesta *formal* al mismo, aunque sí la hay *heurística*: el mecanismo de resolución de un problema debe implementarse en términos de un conjunto finito de operaciones independientes, cada una de ellas con sus propios criterios heurísticos de validez. Ese mecanismo será muy diferente según utilicemos un modelo proposicional de representación o un modelo topológico.

En el primer caso, el modelo proposicional de representación, el mecanismo de razonamiento que utilizaremos para generar una interpretación arqueológica adopta el aspecto de una secuencia de operaciones lógicas muy sencillas, ninguna de las cuales por sí misma, es lo suficientemente sofisticada para generar por sí sola la interpretación, si bien, el esfuerzo combinado de todas esas operaciones permite construir un estado final aceptable. Es decir, un gran número de operadores abductivos muy simples «activa» las distintas unidades de conocimiento que configuran la cadena de inferencias. Dado que en esa cadena las unidades son muy distintas entre sí, las «condiciones de activación» diferirán de unas a otras, teniendo en cuenta, además, el orden o situación de cada unidad en la cadena de inferencias. Se necesita, por tanto, una «reacción en cadena» para poder «activar» la interpretación, ya que estamos obligados a usar como conocimiento previo las unidades de conocimiento que han sido activadas previamente. En términos computacionales, esa reacción en cadena es el resultado de la ejecución de un mecanismo de *propagación de la función de activación*. Ese mecanismo puede llegar a ser bastante complejo, computacionalmente hablando, ya que depende del *orden* de las unidades en la cadena de inferencia –el cual depende, a su vez, de la naturaleza de las diferencias entre las distintas unidades–, así como de la naturaleza de la función de activación de cada una de ellas; por lo general se trata de una función cualitativa (acumulación simple de unidades de conocimiento de menor nivel) y, por lo tanto, discontinua, que enumera los estados intermedios del problema «activados» entre el estado inicial y el estado final.

En el caso de la *representación topológica de las inferencias*, las asociaciones entre unidades adoptan otro aspecto; lo que condiciona el tipo de cadena de inferencias obtenido. La excitación (equivalente a la «activación» proposicional) de uno de los elementos de procesamiento («neurona») que configuran la red neuronal, es el resultado de la excitación de un número  $x$  de neuronas anteriores, así como de la existencia de conexiones excitatorias (conexiones con un peso elevado) entre unas y otras. Ahora bien, esa función -continua y no lineal, a diferencia de la «activación» proposicional- no transfiere información declarativa: el valor excitatorio de las neuronas y de sus conexiones no tiene sentido fuera del entorno de programación. Por consiguiente, las asociaciones abductivas no tendrán que buscarse entre las neuronas, sino entre los *vectores de excitación*. Por ejemplo, para interpretar neuronalmente la función de un útil lítico hemos de traducir previamente la asociación proposicional –más intuitiva–

(Rasgos descriptivos (x), (y), (z)) → Raspador Carenado

en un vector de elementos excitados y pesos de sus conexiones respectivas. Ello se realiza mediante ciertas ecuaciones que tienen en cuenta el número de ejemplos y de rasgos que describen los raspadores carenados conocidos. Siempre que introduzcamos la misma lista de rasgos descriptivos, «excitaremos» el mismo *vector de excitaciones* que representa aquella interpretación «aprendida» por el programa a partir del análisis de un determinado número de ejemplos de raspadores carenados. Obsérvese que se ha establecido una relación de analogía entre el conjunto de rasgos descriptivos (estímulo) y la respuesta generada por el sistema. Que esa «res-

puesta» esté representada *topológicamente* (en términos de un vector de excitaciones) no afecta al contenido de la analogía. La cadena de inferencias, en este caso, tiene tan sólo un estado inicial y un estado final, sin estados intermedios. El inconveniente es que la interpretación ha tenido lugar en una *caja negra*: a no ser que el investigador sea un matemático avezado, no podrá *visualizar* el mecanismo que ha establecido las asociaciones.

Los enlaces o asociaciones que definen una cadena de inferencia son, en realidad, unidades de conocimiento en sí mismas. Su significado suele definirse en términos «explicativos». Es decir, dos unidades de conocimiento están asociadas cuando una de ellas *explica* a la otra. Tradicionalmente la definición rigurosa del término *explicación* se ha resistido a los intentos de la mayoría de filósofos de la ciencia. Si seguimos a un autor como Paul Thagard (1989) tendremos que admitir que lo que nos ha conducido a establecer una «conexión» explicativa son motivos meramente prácticos: *A* explica *B* porque *A* cumple ciertos requisitos. Esos requisitos pueden ser muy diversos, formales o casi formales o bien derivados del objetivo formulado al enunciar el problema interpretativo (en ese caso lo denominaremos requisito heurístico). Lo cierto es que no existe una definición formal (y, por tanto, computable) del término *explicación*. En ausencia de una definición operativa debemos trabajar bajo el supuesto -probablemente erróneo- que una explicación equivale a una asociación objetiva, en donde el adjetivo «objetivo» no hace referencia a ninguna verdad universal, sino tan sólo al ajuste entre las condiciones de activación de una unidad de conocimiento y el objetivo explícito con el que se formula el problema.

El punto de partida de la Arqueología Automática es que el «razonamiento científico» no es más que un conjunto artificial de mecanismos y operadores asociativos muy diversos. Ni podemos referirnos a él como la sublimación del sentido común, ni tampoco podemos reducirlo a un conjunto de operaciones formales sin significado. Parafrascando a Paul Feyerabend, en el razonamiento científico «todo vale», cualquier operador que nos permita asociar distintos estados de un problema será bienvenido, siempre y cuando la relación entre información empírica y la interpretación se haya puesto de manifiesto. No hay bases cognitivas ni formales para justificar un modo de conocimiento, sino tan sólo motivos prácticos.

### Deducción e Inducción en el razonamiento arqueológico

Hasta no hace mucho se creía que una cadena de inferencias estaba constituida, necesariamente, por unidades de conocimiento asociadas *deductivamente* (Watson et al., 1971). Uno de los resultados de la arqueología automática radica, precisamente, en señalar la naturaleza *heurística*, y por tanto, no obligatoriamente deductiva, de las asociaciones que configuran la cadena de inferencias. Ello no significa, sin embargo, que las deducciones no tengan utilidad en el razonamiento arqueológico.

Los investigadores en Inteligencia Artificial y Psicología Cognitiva han contribuido a redefinir lo que la Lógica Clásica entendía por deducción (cf. Blasius y Burckert 1989, Johnson-Laird y Byrne 1991). Un razonamiento deductivo correcto es aquel cuya conclusión es verdadera en cualquiera de las situaciones en las que la premisa es verdadera; por tanto, denominaremos *deducción* a un tipo de asociación entre dos unidades de conocimiento caracterizada por establecer una relación de consecuencia lógica (*implicación*) entre ellas. En otras palabras, entre ambas unidades existe una relación *necesaria* que nace en la propia definición de las unidades y no de una condición heurística externa impuesta por el investigador.

Para poder establecer esa relación de consecuencia entre ambas unidades, es preciso que la segunda de ellas (la unidad *deducida*) contenga explícitamente cierta información que ya existía, aunque de forma implícita, en la primera de ellas (la premisa). A diferencia del razonamiento abductivo, el razonamiento deductivo es incapaz de *crear* nueva información, tan sólo vuelve explícito lo que antes se conocía de forma implícita.

Es fácil reproducir este tipo de razonamiento en un ordenador; basta con que implementemos el mecanismo lógico denominado *modus ponens*:

- |           |   |
|-----------|---|
| Si        | El Estado inicial del problema que se caracteriza por la presencia de los rasgos descriptivos (x) (y) (z) está <i>siempre</i> asociado a un Estado Final caracterizado por los rasgos (a) (b) (c) |
| Entonces, | Dado cualquier estado inicial que contenga los rasgos (x)(y)(z)<br><i>Deduciremos</i> ,   |
|           | La interpretación de (x)(y)(z) es (a)(b)(c).  |

Obsérvese que para establecer una asociación deductiva es preciso que esa asociación haya sido validada previamente. No basta con que conjecturemos heurísticamente la asociación, ni que creamos que es la «mejor» de todas las posibles; hemos de estar seguros que en ningún caso encontraremos una interpretación mejor. Para estar seguros de una afirmación es necesario que demostramos la verdad de cada una de las instancias de esa información. ¡Pero puede haber infinitas instancias! No hay ordenador ni cerebro humano capaz de analizar uno por uno todos los contextos posibles en los que determinada información deba ser, necesariamente, verdadera. Por tal motivo, tanto lógicos como informáticos recurren a ciertos principios que reduzcan de algún modo la evaluación de la «verdad» de una asociación deductiva:

- recurriendo a reglas formales que validen sintácticamente la validez de la asociación. Siempre que el razonamiento reproduzca esa sintaxis lo consideraremos «verdadero». El lenguaje de programación PROLOG, por ejemplo, implementa el llamado Teorema de la Resolución de Robinson (una de esas reglas formales).
- recurriendo a reglas semánticas fijas, esto es, reglas cuyo contenido sabemos válido. Se trata de una variante del *modus ponens*.
- recurriendo a una simulación del contenido de la asociación (Johnson-Laird 1983):
  - se construye un modelo explícito del conocimiento contenido en las premisas
  - se construye un modelo explícito del conocimiento en el consecuente.
  - se intenta unir ambos modelos, analizando su compatibilidad (ausencia de contradicciones).

La aplicación de alguno de estos mecanismos de asociación nos permite establecer, automáticamente y sin necesidad de pruebas ulteriores, la validez e idoneidad de la interpretación generada. Ahora bien, si este mecanismo de asociación fuese el único modo de combinar información, la mayor parte de problemas arqueológicos quedaría sin resolver. Si los conceptos asociados en una cadena de inferencias no cumplen las condiciones siguientes:

- la aplicabilidad de la conclusión ha de ser mayor o igual que la aplicabilidad de las premisas de las que ha sido deducida, nunca menor.
- la conclusión no puede afirmar algo que se haya afirmado con anterioridad
- la conclusión ha de afirmar algo que no esté explícito (aunque, necesariamente, debe estar implícito) en el enunciado de las premisas.

diremos que de la información empírica inicial no puede *deducirse* ninguna interpretación por falta de conocimiento fiable (experimentalmente válido). Ello no quiere decir que ese estado inicial no sea interpretable, sino que la validez de la interpretación (obtenida gracias a los operadores abductivos) no se deriva *exclusivamente* de la descripción inicial. Tal y como hemos visto en la sección anterior, las inferencias no-deductivas (abductivas) son mucho más frecuentes.

En ausencia de enlaces deductivos que garanticen formalmente que cada una de las unidades de conocimiento que aparecen en la cadena de inferencias ocupa el lugar que le corresponde, tendremos que utilizar un criterio heurístico de validación: la interpretación generada tiene que ajustarse a las especificaciones del objetivo del problema. Los *objetivos* de un problema suelen representarse por medio de proposiciones que hay que validar, es decir, por medio de preguntas:

¿(x) se fecha en el siglo IX a.C.?

Ahora bien, tan sólo en el caso en que dispongamos de una gran cantidad de conocimiento bien estructurado podremos resolver el problema respondiendo a una sola pregunta. En la mayoría de las ocasiones, es preciso que hagamos una gran cantidad de preguntas para poder encontrar la solución. Cuando se le plantea una pregunta al sistema, éste activa una de las unidades de conocimiento utilizando alguno de los operadores asociados específicamente a esa unidad. Si el sistema no puede decidir qué operador es el que hay que utilizar, deberemos plantear otra pregunta de tal modo que la respuesta a la misma proporcione la información necesaria para activar la solución a la primera que hemos formulado.

En otras palabras, los «fallos» en la resolución dan lugar a que nos planteemos *subproblemas* cuyo objetivo es, precisamente, resolver la nueva dificultad aparecida reduciendo la distancia entre el estado inicial y al-

guna de las soluciones. Por consiguiente, para poder generar una interpretación precisamos de un *plan* o secuencia de subproblemas. La apariencia externa de ese plan es la de una compleja red en la que están integrados todos los subobjetivos, unidos entre sí por medio de relaciones de «cooperación», pues cada subproblema, con ayuda de los demás contribuye a encontrar la solución idónea al estado inicial.

Los subobjetivos son unidades de conocimiento preespecificadas cuyo objetivo es, precisamente, imponer un orden determinado en el flujo de asociaciones, esto es, en la propagación de la función de activación. Se trata de un conocimiento acerca del procedimiento para resolver el problema, antes que acerca de la solución correcta del problema; nos referiremos a él como *meta-conocimiento*.

La secuencia de subobjetivos ha de estar deductivamente ordenada, de tal modo que un subobjetivo debe deducirse del anterior. Es precisamente esa secuencia *deductiva* la que nos permitirá validar la cadena de inferencias que hemos obtenido abductivamente: cuando la cadena de inferencias reproduzca la secuencia de subobjetivos, el estado final de la misma se ajustará con lo especificado en el objetivo del problema. Aquí radica la noción de *objetividad* (ajuste con los objetivos) y no en ninguna esotérica verdad universal inmanente.

Todas estas dificultades son exclusivas del modelo proposicional de representación de las cadenas de inferencia. Muchos autores piensan que este modelo está limitado por culpa de la necesaria unidimensionalidad del esquema de asociaciones. Sin embargo, hemos visto en la sección anterior que para interpretar los datos arqueológicos podemos (y en muchos casos debemos) utilizar cualquier combinación de enlaces asociativos. Es precisamente la diversidad de los operadores asociativos necesarios para configurar una interpretación la que debiera prevenirnos antes de encerrarnos en cadenas de inferencia con una estructura predefinida.

¿Por qué limitarnos entonces con el uso de estructuras de control de naturaleza deductiva, mucho más exigentes, formalmente, y que exigen grandes cantidades de conocimiento válido, del cual carecemos en la mayoría de las ocasiones? Porque deseamos estar razonablemente seguros que la interpretación es adecuada, y nadie ha dicho que el razonamiento científico tenga que ser sencillo. Ahora bien, ¿no existirá algún método de validación que no exija tanto *meta-conocimiento*? La respuesta es negativa, dentro del modelo proposicional de representación, pero afirmativa dentro del modelo topológico.

La combinación de enlaces asociativos diversos debiera dar lugar a una estructura en paralelo o reticular y no a una cadena o secuencia de unidades. En esa estructura la función de activación no se propagaría unidimensionalmente, sino que se distribuiría en infinitas direcciones. Obviamente, esas estructuras multidimensionales han de ser representadas mediante sistemas topológicos, esto es, *vectores de excitación*. Quizás el formato de las redes neuronales, tal y como ha sido expuesto en la sección anterior no sea el más conveniente, dada su poca transparencia, de ahí el modelo mixto diseñado por algunos autores, en el que las «neuronas» tienen contenido declarativo y las conexiones excitatorias representan la influencia causal de las distintas unidades de conocimiento (Feldman y Ballard 1982, Shastri 1988). Se trata de un sistema quasi-proposicional, en el que el orden de las inferencias puede establecerse matemáticamente a partir del peso de los enlaces entre las neuronas (cf. Churchland 1989).

En tales circunstancias, el mejor modo de calcular el ajuste entre la interpretación y el objetivo es por medio de operaciones de *optimización*, esto es, el cálculo de la mejor de todas las funciones continuas que relacionan el estado inicial con el final. No detallaremos aquí las ecuaciones necesarias para *optimizar* una interpretación (cf. Pao 1989, Zeidenberg 1990), si bien podemos hacer ciertas apreciaciones: teóricamente esas ecuaciones son *universalmente válidas*, esto es, no dependen del problema concreto que se pretende resolver, ya que hacen referencia a la sintaxis y no a la semántica. Algunos algoritmos de optimización (por ejemplo el algoritmo de Kohonen) utilizan criterios formales muy estrictos, según los cuales la mejor solución es aquella que adopta una estructura matemática determinada, usualmente basada en Teoría de las Probabilidades. Obviamente esa solución nos devuelve a la polémica positivista acerca del uso de reglas sintácticas para demostrar la verdad de las afirmaciones científicas. Por otro lado, los algoritmos más comúnmente usados en la programación de redes neuronales (el algoritmo de retro-propagación) exigen también meta-conocimiento. Ese «conocimiento acerca del problema» es muy distinto del que necesitábamos en el caso proposicional: en lugar de desarrollar el plan del problema (secuencia de sub-objetivos), se precisa de un conjunto adecuadamente grande de ejemplos a partir de los cuales puedan calcularse las propiedades topológicas de la solución óptima.

Hasta ahora no hemos mencionado para nada el razonamiento *inductivo*, fundamental para muchos arqueólogos (Clarke 1968): los conceptos científicos, las interpretaciones, se generaban por medio de ciertos operadores estadísticos (correlación de atributos), utilizando como información inicial una descripción de los datos. Según este enfoque, las interpretaciones se expresarían mediante series de atributos estrechamente correlacionados y el mecanismo de producción de *interpretaciones* se reduciría a la aplicación de operadores

inductivos universales -por ejemplo, la correlación estadística y sus derivados- sobre la información inicial, sin que se precise ningún tipo de hipótesis previa cuya relación con la evidencia fuese preciso establecer.

A lo largo de este libro se ha desarrollado una concepción muy distinta de la filosofía inductiva y de su relevancia en la investigación científica. Los arqueólogos no analizan los fenómenos sociales aplicando fórmulas mágicas a sus datos y usando el resultado de esas fórmulas como conceptos o leyes absolutas. Debieran ser capaces de *aprender conceptos* y no tan sólo señalar la existencia de co-ocurrencias. La noción de «aprendizaje» suele definirse como «aquel fenómeno exhibido cuando un sistema (agente humano o máquina) mejora en la ejecución de una tarea concreta, sin necesidad de haber sido reprogramado» (Anderson 1983, Newell 1990). Por consiguiente, si la inducción ha de facilitar el «aprendizaje» de nuevos conceptos, interpretaciones y soluciones a problemas diferentes, la inferencia inductiva se caracterizará por la *modificación* progresiva de la formulación de un concepto inicial (hipótesis), acercándose paulatinamente a un concepto «mejor» que el de partida. «Mejor» puede significar cosas distintas: un mayor porcentaje de soluciones correctas, una respuesta más rápida, soluciones más prácticas o eficaces, o con un mayor abanico de posibilidades de aplicación, un ajuste más apropiado con el objetivo, etc.

La modificación inductiva de una categoría puede definirse, también, en tanto que *transferencia* de conocimiento de un conjunto de unidades a otra que, por el mero hecho de recibir esa transferencia de conocimiento, altera su naturaleza lógica volviéndose más general. Sin embargo, no toda transferencia de conocimiento es inductiva -es decir, permite aprender-: el conocimiento de la segunda unidad tiene que *aumentar* como consecuencia de esa transferencia. Pero no hay aprendizaje, no hay «incremento» del contenido, si no se definen previamente los parámetros que permitan establecer la «mejora» del sistema. Aquí radica el *problema de la inducción*, que durante siglos se han planteado los filósofos: a diferencia de lo que sucede con la deducción, no puede probarse si las leyes o entidades generales aprendidas inductivamente son correctas. Mientras que la deducción es un tipo de razonamiento finito, es decir, concluye tras un número determinado de pasos, la inducción no concluye nunca, porque no puede establecerse con seguridad cuál de las modificaciones de un concepto o interpretación es la definitiva. De este modo, la creación de conocimiento científico constituye en realidad un proceso continuo, que no puede reducirse a la aplicación de un único operador.

Por consiguiente, lo realmente importante en una inferencia inductiva no es la naturaleza de los operadores (estadísticos o lógicos), sino las heurísticas de procesamiento que asegurarán que las conclusiones inductivas son plausibles y relevantes para el objetivo que se está procesando. El control de la inducción tiene más importancia que la inducción en sí misma. Hay que limitar la búsqueda de generalizaciones y/o asociaciones, aun con el riesgo de omitir informaciones importantes. No puede haber aprendizaje si no se controla de un modo u otro el mecanismo de inducción, ya se trate de la generalización o de la correlación de propiedades. Cuanto más estrictas sean las condiciones que impongamos al mecanismo, mayor riesgo de eliminar generalizaciones interesantes; si esas condiciones fuesen demasiado débiles, las dimensiones del espacio de búsqueda serían excesivas para encontrar en él hipótesis satisfactorias.

De este modo, en el dominio de la Inteligencia Artificial, las inferencias inductivas se caracterizan de un modo bastante distinto al de la inferencia estadística clásica: el objetivo no es distinguir qué atributos o variables están correlacionados con qué objetos en una base de datos, sino en modificar progresivamente la definición de un objeto paradigmático de modo que la definición final sea utilizable para todas las instancias posibles del mismo objeto. *El razonamiento inductivo, pues, puede estudiarse a través de la modificación del conocimiento a través de su uso.*

A diferencia de lo que creían los primeros partidarios de la nueva Arqueología no necesitamos mecanismos formales estrictos (deducción lógica) para interpretar los datos arqueológicos, aunque nada nos impida utilizarlos en la resolución de determinados problemas. Ahora bien, no debemos prescindir del carácter «racional» de la arqueología, aun cuando ni la deducción ni la inducción sean mecanismos de inferencia fundamentales. Existen otros mecanismos mucho más útiles e igualmente efectivos (abducción, analogía), que nos permiten definir el «razonamiento arqueológico» en términos de un conjunto de mecanismos de *combinación conceptual*, hayan sido santificados o no por la Lógica de Predicados (Thagard 1988, 1990). En definitiva, el investigador debe combinar la información que dispone para poder producir interpretaciones (conceptos). Brian Falkenhainer propone la siguiente *conjetura de similaridad* al respecto:

Todo proceso de construcción de interpretaciones puede caracterizarse como la búsqueda de la similaridad máximamente explicativa entre la situación que se desea explicar y cierto fenómeno explicado con anterioridad. La situación previa puede haberse extraído de una experiencia actual, una ex-

periencia prototípica o una situación imaginaria derivable del conocimiento general» (Falkenhainer 1990: 164).

Ni la combinación de conceptos ni la búsqueda de una «similaridad máximamente explicativa», sin embargo, son operaciones formales; existen numerosísimos operadores combinatorios y muchos tipos de correspondencia (similaridad) entre los estados intermedios de un problema, cada uno de los cuales será útil en determinadas circunstancias. El único requisito «objetivo» para que una combinación conceptual produzca una inferencia válida es que la secuencia de combinaciones se haya producido siguiendo un plan específico y no al azar. En otras palabras, ha de resultar posible, en todo momento, saber *cómo* (secuencia de combinaciones) y *por qué* (plan que hay que seguir para resolver el problema) se ha generado esa interpretación y no otra.

Esta caracterización de la naturaleza del razonamiento científico debiera permitirnos distinguir las funciones de las tres modalidades clásicas: abducción, deducción, inducción:

- La *abducción* permite resolver problemas, una vez que el estado del problema ha sido configurado. Su característica fundamental es la existencia de una correspondencia heurística (no válida sintácticamente, sino pragmáticamente) entre el estado inicial y el estado final del problema que se desea resolver.
- La *deducción* permite controlar la «objetividad» de un mecanismo automático de resolución de problemas. Estableciendo relaciones de similaridad literal entre los distintos subobjetivos que configuran el árbol de descomposición de un problema, impondremos al algoritmo de resolución un *ajuste máximo al objetivo inicial* (esa es la definición de «objetividad»). Atención, una solución «objetiva» puede ser falsa y válida a la vez. El término validez hace referencia tan sólo a la aceptabilidad individual de un resultado, no a su relación con el mundo real.
- La *inducción* debe entenderse como una «propiedad» del razonamiento científico, antes que como un tipo de inferencia. Consiste en la modificación constante de conceptos y soluciones previas, si bien, para ser auténticamente inductiva, esa modificación *controlada* ha de conducir a conceptos «mejores», heurísticamente, que los de partida.

Sólo podremos «razonar» cuando dispongamos del espacio de un problema y alguna solución inicial. Ahora bien, si las «interpretaciones» no pueden construirse sin conocimiento previo, ¿cómo crearemos esos conceptos iniciales? Aquí radica, a mi juicio, la cuestión fundamental en Filosofía de la Ciencia. Ningún arqueólogo trabaja en un vacío conceptual, sino que es partícipe de un conocimiento, parte del cual ha aprendido en la universidad, parte del cual es constitutivo de la ideología social que él, como miembro de una comunidad, ha contribuido a ampliar. Ese Conocimiento Inicial coincide con lo que Kuhn denominaba «paradigma»; en cierto sentido es ajeno al investigador, que muchas veces es inconsciente que está utilizando un espacio de problema plenamente configurado (el malhadado *sentido común*). Algunos trabajos recientes (Stoczwoski 1991) han demostrado la aparente «vida propia» que tiene ese conocimiento inicial, influyendo al investigador en la mayor parte de sus decisiones.

Por consiguiente, siempre existe un espacio de problema que nos permitirá interpretar, mal o bien, nuestras evidencias. Obviamente, ese espacio será distinto según el grado de formación del arqueólogo. Nuestros conocimientos iniciales son mucho mejores -al menos eso es lo que quiero suponer- que los que disponían los arqueólogos de principios de siglo. Lo realmente importante es que hemos de ser *conscientes* de ese lastre que llevamos encima; más allá del método empleado para resolver nuestros problemas y del control de esa decisión, debemos ser capaces de modificar paulatinamente nuestros resultados de manera que aumente su «calidad», esto es, que tengan una mayor validez, una mayor sencillez, sean aplicables a un mayor número de casos,...

## La Naturaleza de las Teorías Arqueológicas

En capítulos anteriores hemos podido apreciar la necesidad imperativa de conocimientos *estructurados* para poder resolver problemas arqueológicos. Los arqueólogos no debieran limitarse a la mera acumulación de datos empíricos simples, sino que deben estructurar y relacionar toda la información que consiguen para poder llegar a resolver nuevos problemas interpretativos. La tarea del arqueólogo es *asociar* paulatinamente uni-

dades de conocimiento de naturaleza muy diversa, estableciendo una jerarquía de entidades teóricas cada vez más alejadas del nivel observacional y cuya generación requerirá cadenas de inferencia extraordinariamente largas, así como una enorme variedad de operadores heurísticos. Una vez verificadas empíricamente, esas asociaciones se convertirán en unidades de conocimiento de pleno derecho y pasará a formar parte del conocimiento preexistente, a la espera de ser asociadas a otras unidades. En definitiva, una Teoría contiene interpretaciones que han sido generadas, en algún momento anterior como solución a cierto problema.

Las Teorías Arqueológicas, por tanto, no son más que bloques de conocimiento, que forman estructuras extraordinariamente complejas. Ese conocimiento aparece bajo la forma de asociaciones cuya validez ha sido establecida heurísticamente, esto es, por su ajuste a algún objetivo específico. Ahora bien, una Teoría debe contribuir a resolver una gran cantidad de problemas distintos: ninguno de los bloques de conocimiento puede ser validado, simultáneamente, en todos los dominios de aplicación de la Teoría. De ahí que las entidades teóricas sean tan sólo válidas particularmente, y no actúen en ningún momento como axiomas o leyes universales. La validez de todas las unidades es revisable; cualquier asociación es automáticamente eliminada si se demuestra su inutilidad.

Si queremos «programar» en el ordenador una Teoría Arqueológica, habremos de tener en cuenta todas esas consideraciones. Una teoría no es una base de datos, sino un programa aparentemente «inteligente» capaz de operar con nuestros datos, permitiendo inferencias automáticas, esto es, la combinación de datos e hipótesis para generar nuevos datos que resolvieron los problemas que tenemos planteados. En cualquier caso, una *Teoría Computacional* puede describirse en tanto que base de datos con la propiedad de «modificarse» a sí misma para realizar un fin concreto, pues, sin finalidad, no habría «inteligencia».

La utilización de una teoría computacional se diferencia, pues, del empleo de bases de datos clásicas (relacionales). En estas últimas de lo que se trata es de «recuperar» ciertas unidades de conocimiento (los datos) que se han introducido previamente: el usuario *consulta* la información. Para llevar a cabo esta tarea, el programa gestor de la base de datos dispone de un lenguaje especializado en el que se programarán las consultas, las cuales estarán representadas por medio de una enumeración simple de las características que ha de cumplir el dato buscado. El usuario de la teoría computacional, por el contrario, no *consulta* información, sino que «activa» aquellas unidades de conocimiento que el programa considera son las soluciones adecuadas al problema planteado. Puede pensarse que esa «activación» no deja de ser una consulta maquillada; sin embargo hay una diferencia fundamental entre la interrogación de una base de datos clásica y la interrogación de una teoría computacional: se necesita *información contextual* para activar las posibles soluciones al problema. Es decir, los contenidos de la teoría computacional serán activados si y sólo si son relevantes en el contexto marcado por el problema que hay que resolver.

En términos informáticos, una teoría computacional es una *memoria asociativa*, es decir, un sistema de almacenamiento de la información capaz de acceder a su contenido por medio de asociaciones y correlaciones. Existen diversos tipos de memorias asociativas, si bien la mayoría de ellas comparten las mismas características: 1) pueden detectar similaridades o analogías entre las unidades de conocimiento que contienen y nuevas unidades de conocimiento introducidas por el usuario; 2) la ejecución del programa no se deteriora por el uso de información incompleta.

Otra propiedad importante de las memorias asociativas, y que es la que más nos interesa para programar teorías computacionales es la *direccionalidad por el contenido*. Tradicionalmente los ordenadores sitúan unidades individuales en lugares concretos de sus bancos de memoria llamados «direcciones». Esas unidades pueden recuperarse sólo en el caso que se conozca la dirección correcta. En una memoria asociativa dotada de la propiedad anterior no existe una única ubicación en la memoria que contenga cada uno de los distintos ítems de información. Por el contrario, la responsabilidad de representar el conocimiento científico recae en el esquema de asociaciones entre los distintos elementos de construcción, ya sean estos predicados simples o nodos en un grafo. Una memoria direccional por el contenido utiliza los valores específicos de numerosos elementos de programación para acceder a la ubicación de los conceptos.

Esta importancia de lo *asociativo* en las teorías computacionales afecta también al significado o identidad semántica de las unidades de conocimiento: el significado de las entidades teóricas no deriva de su aplicación observacional, sino que depende de las inferencias (cadenas de asociaciones) en las que toma parte. En otras palabras, la identidad semántica de un concepto está determinada por su situación peculiar en la red de asociaciones (cf. Churchland 1989).

Aún no se ha podido programar una teoría computacional en su integridad, aunque sí se conocen los componentes informáticos que deberá tener: un mecanismo de solución de problemas capaz de tratar inferen-

cias para construir la estrategia de satisfacción de ciertos objetivos, una memoria accesible (asociativa), y un mecanismo de inducción que actualice el conocimiento que se vaya obteniendo. Es importante señalar que, en esta estructura, los objetivos, y por consiguiente la estructural de control deductivo del flujo de inferencias, no pertenecen al sistema cognitivo, sino que son introducidos por el usuario siempre que éste utilice los contenidos de la teoría. Es obvio que el investigador propone esos objetivos, en tanto que miembro de un determinado grupo social; el razonamiento arqueológico está afectado por los valores sociales. Sin embargo, en tanto que conjunto *artificial* de mecanismos que depende de la voluntad del investigador, nada impide trascender esos valores. Todo depende de los riesgos que el investigador deseé asumir.

En definitiva, una teoría computacional debe contener:

- un conjunto de conceptos que describan el conocimiento relevante al problema que se quiere resolver
- una representación adecuada de esos conceptos, esto es, una representación *activa* que permita que los conceptos reaccionen a los mensajes enviados por el usuario o por los otros conceptos del Sistema
- un conjunto de reglas que gobiernen la descripción de los conceptos en términos de su representación
- un conjunto de operadores que actúen sobre las representaciones.

Otra de las características básicas de cualquier teoría computacional es que su arquitectura interna sea análoga al mecanismo de razonamiento que hemos descrito en páginas anteriores, puesto que, al margen de interpretar las observaciones empíricas, la teoría debe proporcionar una explicación de las unidades de conocimiento usadas para explicar las observaciones; es decir, debe especificar el *cómo* y el *por qué* de la interpretación generada. Según este principio, la arquitectura idónea de una Teoría Computacional sería modular:

- 1 para responder al objetivo **F**, se necesita el conocimiento **C**
- 2 el objetivo **G** hace referencia cómo se ha resuelto **E** con ayuda de **C**
- 3 para responder al objetivo **G** se necesita el conocimiento **D**
- 4 el objetivo **H** hace referencia a la relación entre **C** y **D**
- 5 para responder al objetivo **H** se necesita el conocimiento **E**

y así *ad infinitum*.

Una consecuencia importante de este enfoque estriba en la necesidad de diferenciar claramente la tarea de adquirir el conocimiento necesario para resolver un problema situado más arriba en la escala de abstracción que acabamos de presentar. Un programa que carezca del conocimiento **D**, por ejemplo, será incapaz de resolver el problema **G**, pero no por ello le faltará «inteligencia»: es inteligente con respecto al problema **F**, aunque no lo sea con respecto al problema **G**. Ningún filósofo de la ciencia debiera de extrañarse; la respuesta a la Paradoja de Russell y al Teorema de Gödel pasa por construir sistemas formales que incluyan a un sistema formal incompleto. Como *todos* los sistemas formales son incompletos por definición, la tarea de construir sistemas que los incluyan no acabará nunca; tampoco el razonamiento científico tiene un final, aunque el conocimiento sea acumulable.

Los datos y conceptos científicos son, esencialmente, dinámicos, de ahí que la Teoría Computacional deba representar de un modo u otro, el cambio entre dos momentos o estados. Muchos avances científicos dependen de la introducción de nuevos conceptos que sólo al final demuestran no ser válidos. Su indeterminación suele descubrirse a la larga, porque ha aparecido una situación inesperada en la que un conocimiento, hasta entonces válido, genera conflictos. En otros casos, el investigador es consciente que está trabajando con una información disyuntiva, incompleta, negativa o meramente implícita. De ahí que el razonamiento científico precise de la facultad de elaborar inferencias plausibles en presencia de una información que a menudo es incompleta y evolutiva, esto es, cambiante con el tiempo. Este peculiar tipo de razonamiento no se deja formalizar por la lógica clásica, que trabaja con premisas verdaderas universalmente, completas y no cambiantes. En el razonamiento científico, las conclusiones obtenidas en un momento dado, ya no serán válidas cuando se haya revisado la teoría que la produjo. En otras palabras, la principal característica del razonamiento científico es la de basarse en inferencias «revisables», o no verdaderas tautológicamente.

Las teorías computacionales han de permitir inferencias revisables; para ello tendremos que implementar las dos propiedades fundamentales de ese tipo de razonamiento, en primer lugar a propiedad de *no-monotonía*, que se define del siguiente modo:

lo que es deducible en el instante **t**,  
no lo es necesariamente en el instante **t+1**.

En otras palabras, el número de conclusiones válidas (extensiones) de en una inferencia científica, puede decrecer al aumentar las premisas de la misma.

La otra propiedad del razonamiento «revisable» es la *pluri-extensionalidad*: a partir de un mismo conjunto de informaciones iniciales, pueden obtenerse distintos conjuntos incompatibles de conclusiones. Es decir, dado un mismo problema y una única Teoría, pueden inferirse distintas soluciones. El orden en el que se producen las inferencias determina la estructura final del conjunto de soluciones. Mientras que los sistemas deductivos formales obtienen un único conjunto cerrado, completo y universal de conclusiones, los sistemas no-monótonos obtienen conclusiones muy diversas, particulares y abiertas, que no se «deducen» de las premisas, sino que son meramente compatibles con ellas. La programación de sistemas no-monótonos en los que la pluri-extensionalidad no degenera en contradicciones constituye uno de los dominios más dinámicos de la Inteligencia Artificial. (cf. Besnard 1989, Grégoire 1990, Lukaszewicz 1990).

De la programación de teorías computacionales se puede derivar una nueva definición del término *teoría*, una definición que dé cuenta de las ventajas del uso del ordenador como vehículo de representación. Allen Newell (1989), propone la siguiente:

«un cuerpo de conocimiento del cual pueden obtenerse respuestas a ciertas preguntas».

Las preguntas, obviamente, pueden ser muy diversas, tanto como las respuestas que se obtengan: predicciones, explicaciones, interpretaciones. Lo importante de las Teorías Computacionales es que nos permiten apreciar que es la teoría la que proporciona las respuestas, y no el teórico. Tanto los seres humanos como las teorías son agentes «racionales» capaces de responder preguntas. Los seres humanos, sin embargo, no suelen proporcionar información adicional acerca de la corrección de su respuesta, y si lo hacen están obligados a recurrir a una teoría. En ausencia de criterios de validación externos, los seres humanos emplean la argumentación como mecanismo de crítica y reformulación, evaluando la idoneidad de las respuestas con arreglo al cumplimiento o no de sus propios objetivos individuales (subjetivos), prescindiendo de los objetivos sobre los que estaba basada la pregunta. Por el contrario, una teoría científica -computacional o no- es un cuerpo *explicito* de conocimiento, a partir del cual cualquier persona lo suficientemente entrenada podrá extraer respuestas a sus preguntas.

Obviamente, qué preguntas encontrarán respuesta y qué preguntas permanecerán sin solución dependerá de la preparación del usuario de la teoría; pero no así la respuesta en sí, la cual, una vez obtenida, depende exclusivamente de la teoría; en nuestro caso, del conocimiento implementado en el ordenador y de la manera en que ha sido implementado.

### **¿Cómo «evaluar» una Base de Conocimientos?**

Aceptar o rechazar Teorías es algo mucho más complejo que rechazar cierta unidad de información concreta, ya que una Teoría se compone de muchas unidades de información interrelacionadas. Contrariamente a lo que supondría Karl Popper, entre otros, no podemos rechazar una Teoría simplemente porque falle ocasionalmente: la «evaluación» de una teoría Computacional no puede establecerse, exclusivamente, en términos de sus resultados.

La verificación de cualquiera de los programas que hemos comentado en capítulos anteriores debiera empezar por el estudio de la concordancia entre el Programa y la Teoría que se supone representa. Dicho de otro modo, ¿produce el Sistema Informático los mismos resultados que lograría un investigador especialista en ese ámbito con ayuda de la Teoría en cuestión? Supongamos que disponemos de un Sistema Experto capaz de datar cerámica y que lo utilizamos para datar un artefacto específico. El programa ha producido los siguientes resultados:

SIGLO IX A.C. : 60 %  
 SIGLO VIII A.C.: 45 %  
 SIGLO VII A.C.: 40 %  
 SIGLO VI A.C.: 20 %

Es decir: para el programa, este artefacto arqueológico tiene un 60% de probabilidades de ser datado correctamente en el Siglo IX a.C., y sólo un 20 % de probabilidades de ser datado en el siglo VI a.C. Nótese que el ordenador *no está seguro* de la respuesta, pero «cree» que la mejor solución es: «Siglo IX a.C.». Por su parte, un arqueólogo, especialista en ese periodo, considera que las probabilidades de datación correcta de esa misma pieza de cerámica son:

SIGLO IX A.C. : 80 %  
 SIGLO VIII A.C.: 75 %  
 SIGLO VII A.C.: 30 %  
 SIGLO VI A.C.: 0 %

Para el investigador humano «lo más probable» es que esa cerámica sea datable en los siglos IX y VIII a.C.

¿Qué significan estas cifras? Básicamente, se trata de una simple representación numérica de la inseguridad o incertidumbre del Sistema Experto y del Arqueólogo a la hora de distinguir entre el siglo IX y el siglo VIII a.C. La incertidumbre es el resultado lógico de una base de conocimientos incompleta. Expresándolo en términos informáticos:

- la Base de Hechos que contiene la descripción de esa cerámica activa tan sólo un 60 % de las condiciones necesarias para que un artefacto sea datable en el siglo IX a.C., a juicio del Sistema Experto
- la Base de Hechos que contiene la descripción de ese artefacto activa tan sólo un 80 % de las condiciones necesarias para que sea datable en el siglo IX a.C., a juicio del Arqueólogo.

Para evaluar la idoneidad del Sistema Experto, lo primero que tendremos que hacer es *comparar* los resultados obtenidos por uno y otro. A tenor de las cifras, el arqueólogo parece estar más seguro que el Sistema Experto de la «antigüedad» de ese artefacto. Concretamente, sabe con certeza que resulta imposible datarla en el siglo VI a.C., cosa que el Sistema Experto tan sólo sospecha.

Por otro lado, ambas series de «probabilidades» cronológicas son *compatibles ordinalmente*. Este tipo de compatibilidad suele utilizarse bastante como índice de validación: si el orden de las predicciones del Sistema y del arqueólogo es el mismo, los resultados del Sistema serán «válidos», en tanto en cuanto su orden secuencial «coincide» con el propuesto por el arqueólogo.

En consecuencia, para medir el grado de compatibilidad entre los resultados de un Sistema Experto y los de un Experto humano no basta tan sólo con contar el porcentaje de errores cometidos por el Sistema, donde el término «errores» equivale a resultados distintos. Es necesario introducir también la «importancia» subjetiva de dichos errores, y el «coste» de un error: en nuestro primer caso cualquier arqueólogo aceptaría que el Sistema se equivocase en la atribución entre siglo IX y VIII, porque él mismo es incapaz de diferenciarlos, si bien no aceptará en ningún caso un error a la hora de distinguir entre siglo VIII y siglo VI. Esta «valoración» suele introducirse por medio de las denominadas *matrices de confusión*:

	SIGLO IX	SIGLO VIII	SIGLO VII	SIGLO VI
SIGLO IX	—	30	50	70
SIGLO VIII	50	—	50	70
SIGLO VII	70	70	—	70
SIGLO VI	100	100	100	—

Una entrada de 100 sugiere que confundir un diagnóstico con otro es muy grave. Según la información contenida en esta matriz de confusión, por tanto, sería menos importante que el Sistema Experto se equivocase entre el siglo IX y el siglo VIII que entre el siglo VIII y el siglo VI. La matriz no tiene por qué ser simétrica; por ejemplo, puede que sea más sencillo saber si una pieza data en el siglo VIII que en el IX, debido a que el

registro arqueológico del siglo VIII es mejor conocido; en la matriz, por tanto, es menor el riesgo de confundir siglo IX con siglo VIII, que el de confundir siglo VIII con siglo IX.

Los errores de diagnóstico también suelen representarse por medio de matrices de confusión. Así, por ejemplo:

	SIGLO IX	SIGLO VIII	SIGLO VII	SIGLO VI
SIGLO IX	—	20	25	0
SIGLO VIII	20	—	30	30
SIGLO VII	10	25	—	20
SIGLO VI	0	10	10	—

que se leería del siguiente modo: el Sistema Experto confunde los objetos del siglo VIII y los del siglo VII en el 30 % de los casos; los del siglo IX y los del siglo VI en ningún caso, los del siglo VIII y los del siglo VI en el 30 %, y así sucesivamente.

En resumen, disponemos de tres índices de verificación de la concordancia Teoría/Programa:

- Orden de los Diagnósticos
- Grado de Confusión
- Exactitud

Estos índices no son independientes unos de otros. Por ejemplo, supongamos que hemos obtenido:

Coincidencia en el Orden de los Diagnósticos: 74 %

Grado de Confusión: 11 %

Falta de Exactitud: 48 %

Estos porcentajes no son otra cosa que el porcentaje medio que aparece en las distintas matrices de confusión. Nos fijaremos, en primer lugar en el último de ellos: en el 48 % de los casos, los resultados del Sistema Experto no coinciden con los resultados que un arqueólogo considera «verdaderos». Este resultado podría llevarnos a rechazar el Sistema por poco fiable: nadie necesita un programa que se equivoque la mitad de las veces. Ahora bien, fíjémonos en el resultado del Grado de Confusión (11 %): el «coste» de confundir un diagnóstico con otro es bastante bajo, quizás porque el arqueólogo no está muy seguro que sus propias interpretaciones sean todo lo «verdaderas» que debieran. Por consiguiente, sólo queda fijarnos en la compatibilidad ordinal: en el 74 % de los casos los resultados del Sistema Experto y los del arqueólogo serán, sino los mismos, sí al menos «comparables».

A este modo de «evaluar» la ejecución de un Sistema Experto o de un programa de Inducción Automática podríamos denominarlo «análisis de la coherencia externa», pues lo que se pretende es averiguar si los resultados del programa son *coherentes* con los resultados que se derivan de la formulación clásica de la Teoría. Otro modo de evaluación vendría definida, entonces, por el análisis de la *coherencia interna*. Si pretendemos utilizar una Teoría Computacional para realizar tareas explicativas resulta imprescindible garantizar que durante la ejecución de la base de Conocimiento, la solución esperada se obtendrá sin incoherencias. «Probar» la coherencia de los resultados del Sistema supone verificar la coherencia estática de cada uno de sus componentes:

- *coherencia estática de un hecho* .- Es preciso verificar que los elementos constitutivos de un Hecho (atributos y/o Relaciones) no contradigan las asociaciones implementadas previamente en la Base de Reglas.
- *coherencia estática de un conjunto de hechos*.- Obviamente, las referencias cruzadas entre los Hechos deben ser consistentes. No puede emplearse, por ejemplo el encuadre «Cronología del siglo IX» para describir el Encuadre «Cerámica de Barniz Rojo» si éste último cuenta con un atributo contrario a los que definen «Cronología del siglo IX».
- *coherencia estática de una regla*.- La misma relación que existe entre los atributos de un Hecho, debe existir entre las condiciones y consecuencias de una regla.
- *coherencia estática de un conjunto de reglas*.- En la Base de Reglas pueden coexistir, sin problemas, consecuencias distintas (por ejemplo: siglo IX a.C; siglo VIII a.C.). Lo que hay que evitar es

que ambas consecuencias sean activadas por el mismo conjunto de condiciones (y, por extensión, como resultado de la introducción de un mismo hecho).

Por lo tanto, al enunciado declarativo de la Teoría Computacional habrá que añadirle un mecanismo de control de la coherencia estática. La función de un subprograma como éste sería identificar la existencia de:

- *coherencia interna de una regla*: Las premisas (condiciones) de una regla pueden definirse como la especificación de los conjuntos de hechos sobre los cuales esa regla es aplicable. Para estudiar la coherencia estática de una especificación basta con estudiar la coherencia estática de los conjuntos de hechos que constituyen una instancia potencial de esa especificación. La coherencia estática de una regla se verifica cuando las especificaciones de la regla son estáticamente coherentes con las especificaciones de un objeto inicial.
- *redundancia de una regla*: Una regla R es redundante si existe una regla R' tal que el conjunto de los consecuentes de R constituya un subconjunto de los consecuentes de R' y que el conjunto de las premisas de R sean un subconjunto de las premisas de R', sin que varíe el resultado.
- *conflicto entre dos reglas*: Las reglas R y R' están en conflicto si existe una sustitución s de las variables de R y una sustitución s' de las variables de R' tales que, tras la sustitución, el conjunto de las premisas de las dos reglas constituyan una especificación coherente, mientras que el conjunto de consecuentes es una especificación incoherente.
- *reglas con premisas inútiles*: Las reglas R<sub>i</sub> tienen una premisa inútil si tienen todas el mismo consecuente y si el conjunto de las partes de las premisas que las distinguen una de otra es una tautología
- *bucles en un conjunto de reglas*: Un bucle de reglas es un conjunto ordenado de reglas tal que los consecuentes de una regla y una parte de las premisas de la regla siguiente son agrupables, mientras que los consecuentes de la última regla y una parte de las premisas de la primera, también lo son.
- *cadenas contradictorias*: Una cadena contradictoria es un conjunto ordenado de reglas tal que los consecuentes de una regla y una parte de las premisas de la regla siguiente son agrupables, mientras que los consecuentes de la última regla y las premisas de la primera constituyen una especificación estáticamente incoherente.

Si mantenemos la coherencia estática en una Teoría Computacional, posiblemente mantendremos también la coherencia dinámica, esto es, la falta de inconsistencias derivadas del funcionamiento del Sistema. Sin embargo este último punto no puede ser asegurado, ya que dependerá siempre de la coherencia externa de la Teoría que sustenta el Sistema. Precisamente el objetivo de la automatización de una Teoría será encontrar casos empíricos que refuten los conocimientos existentes acerca de la mejor manera de resolver un problema específico.

En definitiva, un Sistema Experto proporcionará resultados «verdaderos» exclusivamente en el caso en que la Teoría Científica que contiene y representa sea capaz de ello. Un Sistema Experto puede representar una Teoría Científica errónea y proporcionar así resultados absurdos, aunque formalmente válidos dentro de la teoría. Si las condiciones de activación de las soluciones y la propagación de la función de activación son fruto de la imaginación del programador y no reproducen una Teoría Científica, el Sistema Experto tendrá el mismo uso que una obra artística: será muy bonito, impresionante quizás, pero inútil para los investigadores.

Por consiguiente, aunque el Sistema Experto reproduzca adecuadamente una Teoría Científica, no habremos avanzado nada en su posible «verdad» empírica.

Obviamente, resulta imposible construir un Sistema Experto que sea siempre consistente y en el que la aparición de anomalías se evite en todo momento. Ello se debe, fundamentalmente a que los criterios operativos utilizados para detectar la «coherencia» no son formales, sino que dependen del dominio de aplicación del programa y derivan, por tanto, de la Teoría Científica que éste implementa. Son «declarativos» y no universales. Es tarea del usuario decidir cuando la aparición de inconsistencias se debe a la deficiente configuración de la base de Hechos (selección errónea de datos empíricos) o a la inconveniencia de la Teoría Científica que se pretende usar para resolver el problema.

## Arqueología Automática

«Si queremos saber cómo funciona la mente, debemos preguntarnos qué es lo que hace y por qué hace lo que hace»

Esta cita es fácilmente aplicable al ámbito de la investigación arqueológica: si deseamos saber qué es una interpretación o una teoría debemos averiguar qué es lo que realmente hacen los arqueólogos cuando «razonan», esto es, cuando interpretan las evidencias arqueológicas. En el ámbito de las Ciencias Cognitivas ha adquirido una relativa fortuna la distinción entre una Teoría en el nivel computacional y una teoría en el nivel algorítmico (Marr 1982). En su nivel computacional, una teoría está caracterizada por aquello que se computa, por qué está siendo computado y a qué condiciones debe ajustarse el procedimiento. Una teoría en el nivel algorítmico especifica cómo se realiza la computación, y debiera ser lo suficientemente precisa como para poder ser simulada mediante un ordenador. Según Chomsky (1965) una teoría en el nivel computacional es una explicación de la competencia humana, la capacidad del investigador para resolver sus problemas, en definitiva, sus *objetivos*. En su realización algorítmica, la teoría debe explicar las características de ejecución humana, dónde falla y por qué nos conduce, en ocasiones, al error; debiera facilitar, igualmente, su integración con las restantes acciones cognitivas.

La Arqueología Automática es, obviamente, una teoría del razonamiento arqueológico en su nivel algorítmico. Su principal objetivo es el análisis de la racionalidad de una teoría por medios computacionales. En otras palabras, averiguar el significado de las interpretaciones arqueológicas en términos de los mecanismos computacionales que permitieron construir esa interpretación y que determinan su uso durante la resolución de un problema. (Thagard 1988: 68).

El estudio de esa a racionalidad suele emprenderse distinguiendo en la Teoría un *componente lógico* (sintáctico) de un *componente subjetivo* (semántico). El primero está incorporado a cualquier programa de ordenador (depurador), y su función es determinar la gama de inferencias incompatibles, lo cual debiera preservar la consistencia del conocimiento arqueológico. El segundo determinará la efectividad de las inferencias.

El objetivo final de cualquier análisis de la racionalidad de una Teoría parte del principio según el cual la calidad de una Teoría Computacional no sólo depende de la calidad del conocimiento que contiene, sino del uso al que se destina esa Teoría. Ahora bien, hemos visto que la efectividad del uso de la Teoría computacional sólo puede realizarse *empíricamente*. La propiedad de la pluri-extensionalidad del razonamiento científico impone, además, una serie de condiciones y límites difícilmente formalizables a la selección del «mejor» contexto de aplicación del conjunto de soluciones posibles. Todo ello hace referencia al *componente subjetivo* (semántico) de las Teorías Computacionales, que podemos describir en tanto que representación del usuario o del acto de *utilizar* el conocimiento.

Podemos programar una Teoría Computacional cuyo dominio de aplicación sea, precisamente, el *uso* de una Teoría anterior, lo cual simularía la actuación de un usuario hipotético. De este modo implementaremos el componente semántico de la racionalidad de la Teoría. Conviene tener presente que el *uso* de la Teoría, esto es, la resolución de un determinado problema científico debe ser consistente con los objetivos marcados, objetivos que constituyen las unidades de conocimiento características de la *meta-teoría*.

En definitiva, el propósito de estos programas de ordenador no es sustituir a los arqueólogos por máquinas inteligentes, sino estudiar los mecanismos de resolución de problemas utilizados por los científicos sociales, o los mecanismos de resolución que *debieran* utilizar. Automatizando el procedimiento de interpretación del registro arqueológico descubriremos los mecanismos cognitivos subyacentes en todo razonamiento científico. La ventaja de la automatización y del empleo de las técnicas de Inteligencia Artificial radica en la superación del límite al que habían llegado los positivistas: las inferencias no deben representarse, exclusivamente, recurriendo a la Lógica de Predicados o a inferencias estadísticas simples. La informática ofrece unos formatos de representación que permiten superar las limitaciones semánticas, tanto del lenguaje natural como de los lenguajes formales al uso. Obviamente, esta representación computacional también está afectada por los teoremas de la incompletitud de los sistemas formales: no se busca construir la máquina que acabe con todas las máquinas, sino analizar cómo interpretan los arqueólogos sus evidencias y, lo más importante, por qué lo hacen de ese modo y no de otro.

A buen seguro habrá quien encuentre las páginas anteriores un nuevo ejemplo de palabrería técnica alejada de las preocupaciones de todo buen arqueólogo práctico. Mi único propósito ha sido el de señalar el principal peligro en cualquiera de las ramas de la Ciencia Social: la *trivialización de las inferencias*. Mientras este-

mos anclados en un modelo verbal del razonamiento y no tengamos conciencia de la diversidad de operaciones asociativas posibles, nos resultará de todo punto imposible comprender la complejidad de los fenómenos sociales.

Lo que propongo no es ninguna panacea universal. Tan sólo un conjunto de métodos técnicas y operaciones que nos permitan construir modelos sofisticados capaces de dar cuenta de la extraordinaria complejidad de la realidad social.

«La tarea de la ciencia no es explicar a qué sabe la sopa, sino explicar cómo ha llegado a tener el sabor que tiene».

Albert EINSTEIN