

{PRIVADO }

## {PRIVADO }2. Representación del conocimiento

Procesos que se adelantan en las investigaciones de IA determinan que el desarrollo de sistemas inteligentes requieren adecuadas representaciones del conocimiento para que imiten la inteligencia. Conocimiento no debe ser confundido con datos o información. El conocimiento incluye y requiere el uso de datos e información. Además, combina relaciones, dependencias, y la noción del saber con datos e información. El conocimiento posee ciertas propiedades: voluminoso, difícil de caracterizarlo, dinámico, con incertidumbres, y debe estructurarse de la manera como se va a utilizar.

La IA busca representar generalizaciones, es decir, no representar cada situación individual, sino agrupar las situaciones que comparten propiedades importantes. De tal forma que pueda: a) ser entendido por las personas que lo provean; b) ser modificado para corregir errores y reflejar cambios en el mundo; c) usarse en muchas situaciones aun sin ser totalmente exacto o completo; d) caracterizarse las técnicas de IA con independencia del problema a tratar; e) usarse superando su propio volumen, y disminuir el rango de posibilidades que normalmente deben considerarse. La IA determina que deberían generarse grandes bases de conocimiento, bases de los sistemas para expertos.

El conocimiento puede ser de tipo procedimental, declarativo o heurístico. Conocimiento procedimental es aquel que es compilado que se refiere a la forma de realizar una cierta tarea (*el saber cómo hacerlo*). Por ejemplo, el proceso estándar para el ensamble de un vestido, un computador, una máquina; la realización de cierta pintura, la resolución de ecuaciones algebraicas.

El conocimiento declarativo es conocimiento pasivo, sentencias que expresan hechos del mundo que nos rodea (el saber que hacer). Por ejemplo, la información en una base de datos.

El conocimiento heurístico es algo especial para resolver problemas complejos. Es un criterio, estrategia, método o proceso que simplifica resolver problemas.

Una base de conocimiento (BC) consiste de esquemas y estructuras de representación y organización del conocimiento suficientemente amplias de una determinada área, a fin de resolver problemas. Por ejemplo, si se tiene un archivo con los datos azul, rojo y amarillo, lo mismo que los valores 10, 20 y 30, entonces las personas que lo lean harán diferentes asociaciones, como color de lápices y su tamaño o volumen, o color de carros y cantidad, o color de luces e intensidad, etcétera. Pero si el archivo contiene además la sentencia **flores**(color, cantidad), entonces se tiene conocimiento. La

representación del conocimiento para sistemas inteligentes parte de considerar el manejo simbólico, la abducción o extracción de patrones dados, y extrapolación en términos del contenido. El significado y la pragmática de un texto son aproximaciones al pensamiento al efectuar un razonamiento. El razonamiento, dicen algunos autores, es difuso, ambiguo, de mucha vaguedad; sea que se realice inductiva o deductivamente. Formaliza<sup>1</sup>, representar o estructurar el conocimiento lleva a especificar adecuadas relaciones para reducir la ambigüedad, la vaguedad, la difusidad en un grado alto y considerar los métodos de inferencia disponibles.

Una representación de conocimiento puede ser un esquema o dispositivo para capturar los elementos esenciales del dominio de un problema, es decir, se hace para:

- \* Manipular conocimiento cualitativo.
- \* Estructurar el conocimiento.
- \* Capturar significado complejo.
- \* Procesar interpretaciones del conocimiento.
- \* Inferir conocimiento a partir de hechos establecidos.
- \* Representar tanto principios generales como específicos.

Por ello, el conocimiento se organiza en una o más configuraciones. Claro está que no puede separarse la representación del conocimiento del uso posterior de *ese* conocimiento.

El conocimiento para involucrar en los esquemas o estructuras se encontraría clasificado en las siguientes categorías:

- \* Objetos. Todo sistema debería tener información de las propiedades físicas de objetos y conceptos.
- \* Eventos. Importante especificar las acciones que pueden llevar a cabo los objetos.
- \* Desempeño. Información sobre cómo realizar ciertas tareas.
- \* Metaconocimiento. Es conocimiento acerca del mismo conocimiento.

Las representaciones combinan declaraciones y procedimientos; el principio de organización fundamental, al hablar de marcos, es el empaquetamiento del conocimiento. Los guiones (*script*, en inglés) permiten razonamiento basado en expectación acerca de situaciones para tener. Un guión es una secuencia de episodios de cierto tipo. Esta secuencia tiene dos propósitos: (1) organizar un conjunto de acciones; (2) predecir la presencia de actividades que no se dan específicamente.

Los métodos orientados a objetos<sup>2</sup> para representación de conocimiento proveen una

---

<sup>1</sup> Determinar cómo formalizar el conocimiento es algo que tiene discusión científica desde el ámbito filosófico, psicológico, neurológico, sistémico y holístico; ante esto no nos queda otra alternativa que cuestionarnos cómo realizar la mejor estructuración y en particular, para el desarrollo de sistemas inteligentes.

<sup>2</sup> En el paradigma orientado a objetos, cada objeto se comunica con otro enviando y/o recibiendo

#### *Representación del conocimiento*

metodología para compartir características con marcos y redes semánticas. El conocimiento está en términos de un conjunto de objetos, cada uno de los cuales posee capas que exhiben ciertas características. Cada objeto se sitúa jerárquicamente en una red y accede a propiedades de otros.

Cuando se trata una nueva clase de problemas, siempre resulta un conjunto de preguntas acerca del conocimiento, como:

- ¿Qué clase de conocimiento está comprometido?
- ¿Cómo será el conocimiento representado?
- ¿Cuánto conocimiento se requiere?
- ¿Qué exactitud del conocimiento se requiere?

Unas características de la representación del conocimiento son:

- \* Alcance y granularidad. El alcance de la representación del conocimiento es la medida del dominio. La resolución o detalle que está en el conocimiento representado es la granularidad.
- \* Modularidad y entendimiento.
- \* Conocimiento explícito y conocimiento implícito.

Las situaciones particulares en las cuales el conocimiento se usa son:

- \* Adquisición. Integración de nueva información al sistema.
- \* Recuperación. Clasificar conocimiento para obtener hechos y relaciones particulares asociados a una tarea específica.
- \* Razonamiento. Proceso en que se combina el conocimiento almacenado.

### **{PRIVADO }2.1. TRIPLETA OBJETO, ATRIBUTO Y VALOR**

Una forma común de representar conocimiento es usar la tripleta objeto, atributo y valor (OAV). Los objetos son físicos o son conceptuales. Un objeto puede tener varios atributos. Los atributos son elementos que caracterizan un objeto. Los valores son las medidas específicas de los atributos en una situación dada (ver tabla 2.1).

Para la representación OAV se usan nodos y relaciones (redes) o tablas conceptuales (ver figura 2.1). Objetos y valores se designan como nodos, mientras los atributos se designan como enlaces (arcos) de ellos. La representación OAV se usa para mostrar orden e interrelación. Las tripletas OAV no muestran herencias, sólo relaciones causales o enlaces, partes o subpartes.

<i>Objeto</i>	<i>Atributo</i>	<i>Valor</i>
---------------	-----------------	--------------

---

*mensajes. Un objeto que recibe un mensaje chequea la base de datos y decide qué acción tomar. Las acciones se realizan invocando un método.*

Objeto	Atributo	Valor
Casa	Alcoba	2, 3, 4, etc
Casa	Color	Verde, rosa, gris
Inventarios	nivel	15, 20, 30
Alcoba	área	9*10' o 15*20 m
María	esposa	Ingeniera
Luis	padre	Médico

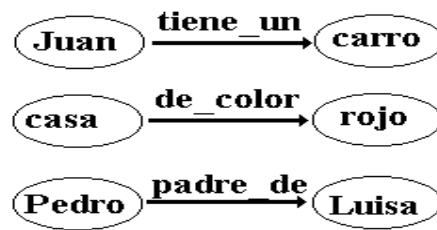


Figura 2.1. Ejemplos de representación de OAV.

## {PRIVADO }2.2. REDES SEMÁNTICAS

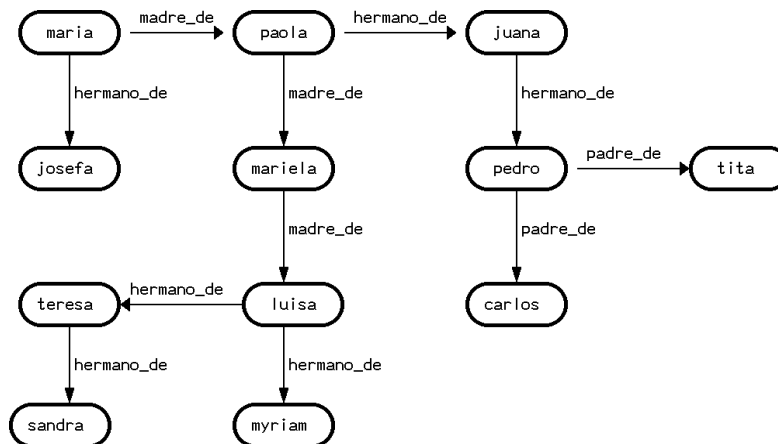


Figura 2.2. Red semántica: relaciones familiares.

Las redes semánticas (*Semantic networks*) son representaciones básicas de conocimiento, compuestas por nodos y enlaces que muestran relaciones entre objetos (ver figuras 2.2, 2.3 y 2.4). Los objetos pueden ser cualquier elemento físico: un libro, un carro, un escritorio, una tela o una persona. También pueden ser conceptos, eventos o acciones. Los conceptos se representan en el modelo como unidades independientes que expresan relaciones de inclusión (es...) y de propiedad (tiene...). Un concepto

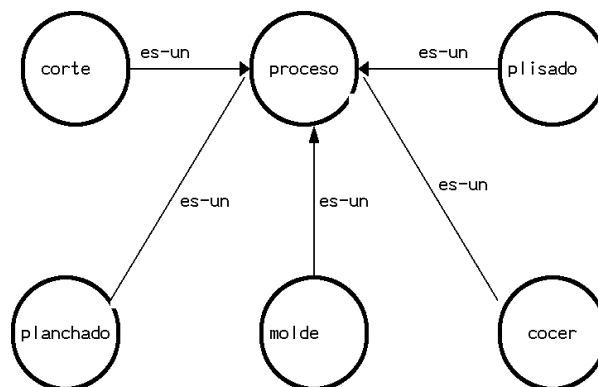
#### Representación del conocimiento

puede ser una relación de oferta y demanda en economía, una elección o una acción tal como producir un vestido. En general, una red semántica es un conjunto de OAV, donde los atributos generalmente representan tamaño, color, clase, origen u otras cualidades<sup>3</sup>.

En una red semántica los atributos son los enlaces (arcos) entre objetos. Los enlaces muestran las relaciones o el vínculo entre un objeto y otro; describen factores. En sí, una red semántica es una representación en la cual:

1. Léxicamente, hay nodos y enlaces entre nodos especificando una aplicación.
2. Estructuralmente, cada enlace conecta un nodo con otro en un orden especial.
3. Semánticamente, los nodos y los enlaces denotan una aplicación específica.
4. Procedimentalmente, existen procesos para:
  - i) Construir nodos.
  - ii) Construir enlaces entre nodos con un nombre.
  - iii) Deducir conocimiento.

La figura 2.2 muestra una estructura de relaciones que existe en un conjunto de personas; de ella puede extractarse los enlaces como {madre\_de, hermano\_de, padre\_de}, de las que se deduce que la aplicación es: relaciones familiares.



0Figura 2.3. Red semántica con atributo "es\_un".

Los nombres dados a cada enlace indican la clase de vínculo que hace corresponder un objeto a una categoría. El enlace **tiene-un** se usa para identificar características o atributos de un objeto. Otros enlaces se usan para definir propósitos. Como puede observarse en la figura 2.3, el dominio del conocimiento es el concepto "proceso"; en

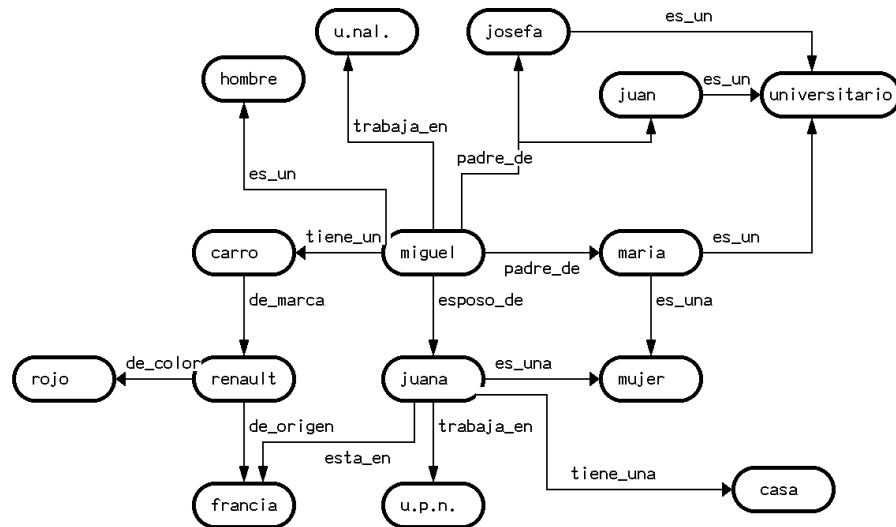
<sup>3</sup> El concepto de red semántica fue introducido en 1968 por Ross Quilliam. Fue diseñada como un modelo psicológico de la memoria humana asociativa. Las redes semánticas fueron desarrollándose por quienes trabajaban en el área de IA. El objetivo de estas redes es la organización y representación del conocimiento general acerca del mundo. El objetivo inicial para el desarrollo de las redes semánticas fue entender el lenguaje natural, más que la clasificación de los datos

la figura 2.4 se ven muchas relaciones y, por tanto, el dominio es más amplio<sup>4</sup>. La unidad del conocimiento es un enlace y dos nodos. Una red semántica ofrece la visión sobre las relaciones y dependencias de un área de conocimiento (dominio) y es muy apropiada para la estructuración del conocimiento. Sin embargo, los enunciados de las relaciones mencionadas en los arcos deben formularse fuera de la red. La red semántica no da información sobre el procesamiento de la red. Algunos nodos y enlaces pueden mostrar otras características acerca de cada uno de los procesos que están tratándose.

Uno de los factores interesantes y útiles sobre las redes semánticas es que a través de ellas pueden verse las características que algunos nodos heredan de otros.

Características:

- Diferencian entre tipos de objetos de los casos. Así, se llama *clasificación* al proceso de ir de ejemplos de objetos a tipos de objetos.
- Se introduce el concepto de *distancia semántica*: cantidad de enlaces que separan un nodo de otro.



0Figura 2.4 Red semántica con diferentes relaciones.

En las redes semánticas, la distancia puede ser importante, se utiliza para localizar objetos poco o muy relacionados, dependiendo de la distancia. En algunos casos, puede disminuirse la distancia eliminando enlaces.

<sup>4</sup> Las unidades conceptuales se organizan en la memoria en niveles que van de los subordinados (más específico) a lo superordinado (más abstracto). El concepto más general e inclusivo se ubica en el tope de la estructura (red) y mantiene, con el resto de los conceptos, relaciones indirectas subordinadas.

#### *Representación del conocimiento*

c. En las redes semánticas también se tiene la idea de *partición*: es el contexto de una red, en el sentido de tener una subred.

d. Existen jerarquías en las redes semánticas<sup>5</sup>. Las principales son: **parte\_de** y **es\_un**. La existencia de jerarquías permite la herencia en la que un objeto que pertenece a una clase hereda todas las propiedades de la clase.

### {PRIVADO }2.3. REGLAS DE PRODUCCIÓN<sup>6</sup>

Para un programa de cálculo numérico no cabe pensar que funcione con parte del programa o con datos incompletos; esto no es posible sino que además da error. Sin embargo, en un SE esto es posible y además es correcto; es decir, debe funcionar con BC incompletas (éstas, poco a poco, van completándose). Esto hace que las funciones de control, la BC y las reglas sean completamente independientes. La BC es la estructura de conocimiento central para desarrollar una tarea particular. Dependiendo de la aplicación, la base de conocimiento puede ser una simple matriz de números o una compleja estructura de archivos indexados relacionados o un conjunto de bases de datos definidas separadamente.

En los sistemas de producción (SP), cada regla tiene una precondition (parte izquierda), y una parte derecha que describe la operación que debe llevarse a cabo si se aplica la regla. Es decir, los sistemas de producción (SP) están compuestos de reglas de producción y hechos.

Los SP fueron desarrollados por Newell y Simons, para modelar el conocimiento humano. La idea básica en estos sistemas es que el conocimiento sea representado como reglas en forma de pares condición-acción:

SI ( condiciones/premisas )	"hechos".
ENTONCES acción-a-ejecutar	"conclusión/acción".

Cada regla contempla premisas que pueden modificarse independientemente de otras reglas. Cuando se combina y alimenta el MI por el conjunto de reglas funcionales, se producen mejores resultados porque se parte de la suma de resultados por reglas individuales<sup>7</sup>. Los SP poseen claramente la separación de los elementos computacionales y en los que se observa la esencia de operación de los sistemas de IA.

---

<sup>5</sup> Un principio organizativo importante en este modelo es el de la "economía cognitiva". Cada concepto articula su significado al asociarse con los nodos de propiedades, pero éstas no se repiten en todos los niveles de jerarquía.

<sup>6</sup> Los teóricos del procesamiento de la información sostienen que el conocimiento se adquiere por la construcción de una representación del mundo exterior. El ser humano reconstruye la realidad elaborando representaciones mentales precisas como las redes de proposiciones, esquemas y reglas de producción condición-acción.

<sup>7</sup> Por tanto, se entiende por SP el conjunto de reglas organizadas que garantizan las transformaciones que el sujeto lleva a cabo para solucionar un problema.

Existen dos tipos comunes de SP:

- a) **Conocimiento declarativo**, el cual hace énfasis en la representación del conocimiento como una acumulación de hechos estáticos junto con una información limitada que describe cómo se va a emplear el conocimiento.
- b) **Conocimiento procedimental**, que enfatiza la representación en forma de reglas describiendo los procesos en el momento de usar el conocimiento. El SP advierte al usuario cómo solucionar un problema con base en hechos conocidos.

Considérense los siguientes ejemplos:

- 1.     **SI** máquinas de corte inactivas,  
          **ENTONCES** programar producción.  
  
          **SI** orden compra de materia prima aprobada,  
          & necesidad de materia prima,  
          **ENTONCES** solicitar envío de materia prima.  
  
          **SI** materia prima ordenada,  
          & inconvenientes de transporte,  
          **ENTONCES** buscar viabilidad de transporte.
- 2.     **SI** fiebre & infección,  
          **ENTONCES** gripe.  
  
          **SI** temblor & dolor huesos,  
          **ENTONCES** infección.

**La estrategia de control del motor de inferencia escoge las reglas a aplicar**, resuelve los conflictos que surjan cuando varias reglas pueden ser aplicadas a la vez y deja el proceso cuando una condición de terminación se satisface. Esta estrategia repetidamente aplica reglas a la descripción de estados hasta que la descripción del estado meta se produce. También se guardan datos de las reglas que han sido aplicadas para evitar secuencias repetitivas que no lleven a la condición final.

En muchas aplicaciones con técnicas de IA, la información disponible no es suficiente para permitir la selección de la regla más apropiada. La operación de un SP puede entonces caracterizarse como un **proceso de búsqueda** en el que las reglas son usadas hasta que alguna secuencia de ellas encuentre la condición de terminación<sup>8</sup>.

Se distinguen dos grandes estrategias de control: irrevocable y tentativa.

**Irrevocable:** una regla aplicable es seleccionada, se aplica sin opción para ser

---

<sup>8</sup> Solucionar problemas aplicando una inapropiada regla puede sustancialmente retrasar la terminación. En esos casos se prefiere una estrategia de control que puede validar una regla y, si más tarde descubre que esta regla fue inapropiada, puede retornar y validar otra.



reconsiderada. (A primera vista, puede verse que este tipo de control no es apropiado para solucionar problemas).

**Tentativa:** una regla aplicable es seleccionada, la regla se aplica, pero se tiene previsión para retornar a un *punto de rompimiento (backtracking)* y aplicar otra regla "mejor". Se establece un punto de selección de reglas, si siguientes computaciones encuentran dificultad en producir una solución; el sistema se devuelve al punto previo de rompimiento, donde otra regla se selecciona para continuar el proceso.

{PRIVADO }Tabla 2.2 Diferencias en tipos de sistemas.

Sistema inteligente	Sistema computacional convencional
Estructurado en módulos. Requieren BC y reglas de producción. Representan el problema. Solucionan diferentes tipos de problemas. Explican su razonamiento	Estructurado en módulos jerárquicamente.  En general solucionan un único problema.

En algunos problemas, se desea la solución sujeta a restricciones adicionales. Por ejemplo, la solución del rompecabezas teniendo el menor número de movimientos, porque cada movimiento tiene asociado un costo y por tanto se requiere el mínimo costo.

Hay muchas diferencias entre la estructura de SI y los sistemas computacionales convencionales.

{PRIVADO }Tabla 2.3 Ventajas y limitaciones de las reglas de producción.

Ventajas	Limitaciones
Son fácilmente entendibles.	Conocimiento complejo requiere muchas (miles de) reglas. Problemas para el desarrollo del sistema.
La inferencia y explicación se deducen fácilmente.	Búsqueda en el programa de control.
Modificación y conservación relativamente sencilla.	Requerimiento de amplio conocimiento y experiencia en conocimiento complejo.
La incertidumbre es fácilmente implementada.	Las inferencias al no estar claramente definidas no son entendibles para el usuario.
Cada regla es independiente de las demás.	Dependencia de reglas en conocimiento complejo.

Los cambios o adiciones en los sistemas convencionales pueden requerir extensiva labor en la adecuación de los programas, estructurar datos u organizar módulos (subrutinas).

Métodos de prueba y error son inherentes para solucionar rompecabezas, por ejemplo, puede argumentarse que si una estrategia de un SP posee suficiente conocimiento acerca del rompecabezas para aplicar una regla a cada estado de descripción, entonces él construye una solución.

Tal argumento falla por desconocer la distinción entre conocimiento local explícito, acerca de cómo proceder para localizar un estado y el conocimiento global implícito de una solución completa.

**El algoritmo básico** del sistema de producción para solucionar problemas es:

```

Algoritmo()
  R: regla
  CR: conjunto de reglas
  DATOS <-- BC inicial
  HQ DATOS satisface la condición de meta:
    SELECCIONAR R de CR y aplicar a DATOS.
    DATOS <-- aplicar(R, DATOS)
  FHQ
FIN Algoritmo()

```

El procedimiento anterior es nodeterminado porque aún no se especifica cómo se va a seleccionar una regla.

**{PRIVADO }Tabla 2.4** Características de las reglas.

	<i>Primera parte</i>	<i>Segunda parte</i>
Nombre	Premisa Antecedente Situación IF x	conclusión consecuente acción THEN y
Naturaleza	Condiciones, similar a conocimiento declarativo	Resolución, similar a conocimiento procedimental.
Tamaño	Puede haber muchos IF	Usualmente una conclusión.
Conector	AND  OR	Todas las premisas deben ser verdaderas para que la conclusión sea verdadera. Si cualquier premisa es verdadera, la conclusión es verdadera.

La representación por reglas es aplicable especialmente cuando hay necesidad de recomendar un curso de acción basado en eventos observados. Las reglas poseen ventajas y limitaciones sobresalientes; algunas de ellas se presentan en la tabla 2.3.

Las principales características de las reglas se resumen en la tabla 2.4.

## {PRIVADO }2.4. REPRESENTACIÓN FORMAL

Algunos métodos en IA se usan para convertir declaraciones y procesos de razonamiento en una forma que sea conveniente para la computación. El resultado es la representación de conocimiento utilizando una lógica.

Se habla de representación formal del conocimiento cuando se aplica lógica de predicados, lógica modal, multivaluada o difusa. La forma general de cualquier proceso lógico se ilustra en la figura 2.5. Primero se da información, se hacen declaraciones y se anotan observaciones. Las entradas a los procesos lógicos se llaman *premisas* o *antecedentes*; la salida se llama *inferencia*. En estos procesos, los hechos verdaderos se usan para derivar nuevos hechos que también son verdaderos.



**Figura 2.5** Proceso lógico.

La lógica de predicados, en primer nivel, representa hechos o datos y los expresa como estructuras. Toda proposición lógica se trata explícitamente como una estructura. La lógica de predicados se ocupa de las relaciones entre antecedentes y consecuentes. De otra manera, la lógica tiene sentido cuando se elaboran argumentos y en ellos se analiza si de algunas suposiciones o antecedentes se derivan conclusiones; por ejemplo:

- H1: Felipe\_Alejandro es el padre de María\_Lucía
- H2: María\_Lucía es hermana de Luis\_Carlos.
- H3: Una persona X es hija de otra Y, si X es hermana de alguna hija Z de la persona Y.

implicación: Luis\_Carlos es hijo de Felipe\_Alejandro<sup>9</sup>.

Una de las formas más sencillas de representación de los componentes de una implicación es la sentencia atómica. Una sentencia atómica se compone de una relación y un conjunto de argumentos entre los cuales se da una relación. La tabla 2.5 muestra ejemplos de sentencias representadas en lógica de predicados.

**Tabla 2.5** Proposiciones en lógica de predicados.

---

<sup>9</sup> En la realidad, esta implicación puede no ser siempre cierta, pero las hipótesis planteadas aquí (sin mirar entornos mayores) nos dicen que la conclusión es verdadera.

<i>Sentencia</i>	<i>Representación</i>
María es la profesora de Juan.	Profesora(maría, juan)
El cobre es un metal.	Metal(cobre)
García Márquez escribió en Aracataca.	Escribió_en(garcía_márquez, aracataca)
El presidente Samper es del Valle y tiene dos hijos de 12 y 15 años.	Presidente(samper,del_valle, 2_hijos, 12_15_años)

Puede hallarse sentencias moleculares como estructuras compuestas de sentencias atómicas y conectores lógicos (AND "&", OR "v", NOT "~", THEN "->").

Para hacer inferencia se necesitan hechos. Éstos son afirmaciones generales o particulares. La tabla 2.6 registra ejemplos de representación en la lógica de predicados.

**{PRIVADO }Tabla 2.6 Ejemplos de hechos.**

<i>Relación o predicado</i>	<i>Argumentos</i>	<i>Hecho</i>	<i>Significado</i>
Llamarse	director, Jaime	llamarse(director, jaime)	El director llamarse Jaime
ser_mayor_que	Andrés, Miguel, 10	ser_mayor_que(andrés, miguel, 10)	Andrés ser mayor que Miguel en 10 años.
Libro	Eneida, García Márquez, colombiano	~libro(eneida, garcía_márquez, colombiano)	Es falso que el libro la Eneida sea de García Márquez, colombiano

Los hechos de la misma clase son siempre isomórficos, es decir, tienen la misma estructura. Tener la misma estructura en este caso significa que la relación es la misma, el número de argumentos es igual y los tipos de argumentos se corresponden.

Los componentes elementales del cálculo de predicados son objetos. Un "OBJETO" puede ser expresado como:

*Variable.* Es un término que permite identificar diferentes entidades o individuos de una clase; por ejemplo: X, Y, Z.

*Constante.* Es un término que identifica un único objeto o entidad de una clase; por ejemplo: Juan, Bogotá, 124, casa.

*Función.* Es una representación de transformaciones del dominio; por ejemplo: t(x), madre(f(X)), marido(x,g(x)).

La figura 2.2 puede escribirse también en forma de predicados:

```
madre_de(maría, paola)
madre_de(paola, mariela)
madre_de(mariela, luisa)
hermano_de(paola, juana)
hermano_de(juana, pedro)
hermano_de(maría, josefa)
hermano_de(luisa, teresa)
hermano_de(teresa, sandra)
hermano_de(luisa, myriam)
padre_de(pedro, tita)
padre_de(pedro, carlos)
```

Esta red escrita en Prolog<sup>10</sup>, queda:

```
/******
Ejemplo: de relaciones familiares
******/
DOMAINS
persona = symbol
PREDICATES
madre_de(persona, persona)
hermano_de(persona, persona)
padre_de(persona, persona)
padre(persona, persona)
madre(persona, persona)
hermano(persona, persona)
abuela(persona, persona)
CLAUSES
madre_de(maría, paola).
madre_de(paola, mariela).
madre_de(mariela, luisa).
hermano_de(paola, juana).
hermano_de(juana, pedro).
hermano_de(maría, josefa).
hermano_de(luisa, teresa).
hermano_de(teresa, sandra).
hermano_de(luisa, myriam).
padre_de(pedro, carlos).
padre_de(pedro, tita).
abuela(X, Y) :- madre(X, Z), madre(Z, Y).
madre(X, Y) :- madre_de(X, Z), hermano_de(Z, Y), Z <> Y
OR madre_de(X, Y).
hermano(X, Y) :- hermano_de(Y, X), X <> Y
```

<sup>10</sup> Las redes semánticas pueden implementarse en casi cualquier lenguaje de computación. LISP y PROLOG son los más populares pero existen facilidades para implementarlas con FORTRAN, SNOBOL, PL/I, PASCAL, APL. C++, KES, KNOWLEDGE PRO, NEXPERT OBJECT, OPS5, SMALLTALK, VISUAL BASIC. Se requiere memoria amplia para almacenar nodos y enlaces.

OR madre\_de(Z, X), madre\_de(Z, Y).  
 padre(X, Y) :- padre\_de(X, Z), hermano\_de(Z, Y), Z <> Y.

El programa en Prolog cuenta con reglas adicionales para hacer inferencias con el fin de proveer otras relaciones.

## {PRIVADO }2.5. OTRAS REPRESENTACIONES

También existen representaciones del conocimiento que ayudan enormemente cuando se usan para configurar las anteriormente mencionadas, pues dan mayor aplicación que al tratarse aisladamente.

### {PRIVADO }2.5.1. Guión, Listas

Un **guión** es la representación de conocimiento por medio de descripciones de una secuencia de eventos. Un guión es una estructura que organiza información describiendo una secuencia estereotipada de eventos en un contexto concreto<sup>11</sup>. Los elementos de un guión típico incluyen: condiciones de entrada, apoyos, funciones, rutas y escenas. Las condiciones de entrada describen situaciones que deben satisfacerse antes de los eventos, que pueden ocurrir y ser válidos de acuerdo con el proceso a seguir. Los apoyos se refieren a objetos que se usan en la secuencia de eventos que ocurren. Las funciones se refieren al personal involucrado en los guiones. El resultado es hallar condiciones viables que existan después de los eventos de acuerdo con el guión. Las rutas se refieren a alternativas que se dan en un guión particular. Finalmente, la escena describe la secuencia de eventos que deberían ocurrir.

Un guión es útil prediciendo qué sucederá en una situación específica, aunque ciertos eventos no se hayan observado. La entrada original permite predecir qué sucederá y cuándo. Si el computador sigue un guión, las preguntas existentes buscan deducir de acuerdo con las limitaciones.

Restaurante:

Objetos: (restaurante, dinero, comida, menú, mesas, ...)

Roles: (personas, meseros, cocineros)

Punto de vista: personas

Tiempo de ocurrencia: 1 p.m.

Lugar de ocurrencia: restaurante Doña Alicia

Secuencia:

1o: entrar al restaurante

...: Buscar mesa

<sup>11</sup> El concepto de guión surge del trabajo de Schank y Abelson en la Universidad de Yale. El objetivo fue desarrollar el entendimiento cognitivo en el computador. El conocimiento es generalmente tomado a partir de situaciones en la solución de problemas; tal conocimiento debe simularse con frecuencia para resolver un problema particular utilizando IA.

*Representación del conocimiento*

...: Ordenar comida  
...: comida  
...: salir del restaurante

Los programas que utilizan guiones almacenan la información mediante los grafos<sup>12</sup> de dependencia conceptual. Sin embargo, la representación en guiones va más allá de la representación de frases aisladas. A parte que los guiones *enlazan causalmente* secuencias de acciones, la diferencia principal consiste en que cada guión posee *roles* correspondientes a las personas que intervienen. Los *roles* y algunos de los *objetos* se representan mediante variables y esto permite que puedan ser asignados a diferentes personas o cosas. El imponer *restricciones* sobre el tipo de personas que pueden desempeñar ciertos roles y sobre el tipo de objetos que cumplen cierta función será de gran utilidad en el proceso de interpretación de textos.

Un guión completo se compone principalmente de los siguientes elementos:

*Escenas*: los sucesos descritos en el guión, enlazados causalmente en forma de secuencia.

*Roles y objetos*: corresponden las personas y las cosas que intervienen; incluyen restricciones para indicar qué personas u objetos pueden ser asignados a las variables.

*Cabeceras*: además de la que da nombre al guión, hay otras que representan condiciones, instrumentos y lugares; su misión es activar el guión en el momento oportuno.

El objetivo de utilizar guiones está orientado ante todo a la comprensión del lenguaje natural (a la interpretación de textos escritos). En la práctica, para comprobar que el sistema ha "comprendido" el texto, suele seguirse uno de estos dos métodos:

- 1) Formularle *preguntas* concretas y ver si responde adecuadamente.
- 2) Pedirle que repita el mismo texto con otras palabras (*paráfrasis*).

En el proceso de inferencia, el primer paso consiste en *seleccionar el guión* que mejor explica la historia que se desea analizar. Posteriormente, hay que *asignar las variables* (identificar los roles, objetos y lugares que interviene). A partir de ahí, el guión "instanciado" (dependiente) permite extraer la información que no aparecía explícitamente en la historia escrita; en esto consiste precisamente la inferencia.

En general, la interpretación de una historia no activa un solo guión sino varios de ellos, por lo que se hace necesario un proceso de *selección* posterior con el fin de determinar cuál o cuáles de ellos son válidas y cuáles se rechazan.

Una de las cuestiones básicas de la interpretación de un texto es averiguar los *referentes de los pronombres* y el *sujeto* de la frase cuando éste se ha omitido. La concordancia de género y número es una condición que limita el espacio de búsqueda,

---

<sup>12</sup> La función principal del grafo es ilustrar una idea y aclarar un concepto, pero pueden tener distintas lecturas ya que el lenguaje gráfico no dispone de reglas concretas.

pero en muchos casos los criterios sintácticos son insuficientes.

Otro tipo de inferencia consiste en dar por supuesto que han ocurrido *los demás sucesos* que se encuentran en el guión, aunque no aparezcan explícitamente en la historia escrita.

Por último, señalar que existen dos métodos básicos de afrontar el problema de la compresión de textos mediante guiones:

El método *ascendente* consiste en analizar palabra por palabra, activando los marcos que mejor explican toda la información que va apareciendo.

El método *descendente* selecciona un marco y trata de "encajar" en él la información de forma predictiva. Este segundo método se utiliza cuando solamente se quiere extraer la información más relevante, pues tiene la ventaja de ser más eficiente y más robusto.

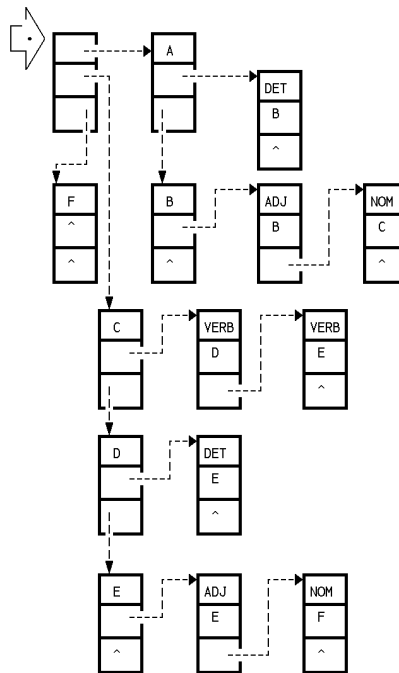


Figura 2.6 Ejemplo de una lista.

Ventajas:

- Poder "leer" textos relativamente breves sobre temas concretos, extrayendo información.
- Representar el conocimiento sin depender de ningún idioma particular: muy importante para los "traductores automáticos".
- Permiten integrar la información con el fin de formar una historia coherente,



*Representación del conocimiento*

y realizan solamente las inferencias relativas a la cadena causal de sucesos.

Inconvenientes:

- *Rigidez del mecanismo de representación*: cada guión representa una secuencia fija de acciones.
- *Incapacidad para compartir información entre ellos*: sería deseable que se combinarán entre sí de forma dinámica.
- *Incapacidad para expresar las "motivaciones e intenciones"*: se observa la necesidad de comprender una historia desde distintos puntos de vista.

Extensiones:

Una solución aportada por Schank para superar la rigidez de los marcos consiste en *representar el conocimiento en forma de "paquetes de organización de memoria o MOPs"*, del inglés, *memory organization packets*.

Una **lista** es una serie de nodos (información) relacionados entre sí. Puede ser una lista de nombres de personas que se conocen, de cosas para comprar en la tienda, de una relación de asuntos para esta semana o de productos en un catálogo.

Las listas se usan normalmente para representar conocimiento jerárquico, cuando los objetos están agrupados o categorizados según un orden o relaciones (ver figura 2.6).

Los objetos se dividen en grupos o clases de elementos similares. Sus relaciones se presentan por enlaces. La forma más simple de representar conocimiento es una lista, pero debe existir una jerarquía cuando dos o más listas relacionadas se combinan.

## **{PRIVADO }2.5.2. Índices, tablas y árboles de decisión**

Los índices, las tablas y los árboles, son herramientas muy útiles para analizar la información o el conocimiento que tiene que ver con las situaciones que se presentan para solucionar un problema, o para relacionar los diferentes eventos que se mezclan en esa solución.

**Índices.** En un índice de decisión el conocimiento se organiza en un formato extendido, usando columnas y filas. El índice se divide en dos partes: 1) lista de atributos; 2) valores posibles de asociación. Las diferentes configuraciones de atributos se emparejan hacia una conclusión.

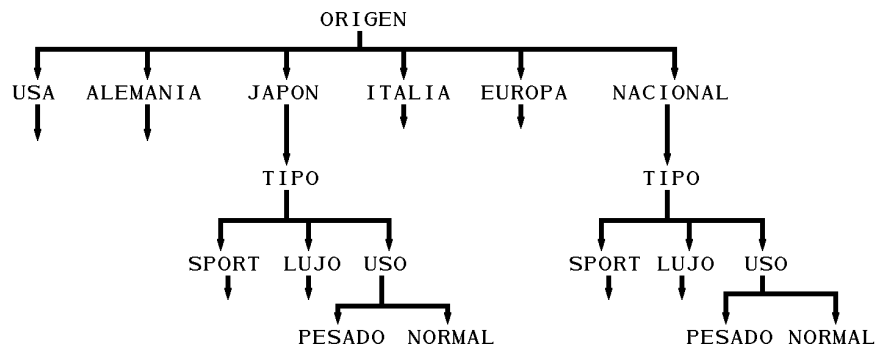
El conocimiento para utilizar en el índice se reúne en secciones de conocimiento. Una vez se detalle el conocimiento para el índice, puede usarse como una entrada a otro método de representación de conocimiento. (No es posible hacer inferencias con el dominio de índices por sí mismos, excepto cuando se usan reglas de inferencia).

PROBLEMAS METAS	PROBLEMA 1	PROBLEMA 2	PROBLEMA 3	.....	PROBLEMA N
META 1					
META 2					
META 3					
⋮					
META M					

**Figura 2.7** Tabla (matriz) de metas contra problemas.

**Tablas de decisión.** En una tabla de decisión, el conocimiento se organiza en un formato, usando columnas y filas (ver figura 2.7).

La tabla se divide en varias partes: 1) atributos; 2) valores o elementos; 3) lista de conclusiones. Una vez esté construida una tabla, el conocimiento en ella puede utilizarse como entrada a otras representaciones de conocimiento. (No es posible hacer inferencias con el dominio de las tablas por sí mismo, excepto cuando una regla de inducción se usa).



**Figura 2.8** Ejemplo de un árbol de decisión.

**Árboles de decisión.** Los árboles de decisión están muy relacionados con las tablas y se usan frecuentemente para análisis de sistemas. Un árbol de decisión puede verse como una red semántica jerárquica limitada por una serie de reglas (ver figura 2.8), que se acoplan para la búsqueda estratégica con las relaciones de conocimiento. Los árboles son similares a los árboles de decisión usados en la teoría de las decisiones.

Los árboles están compuestos de nodos que son metas. La raíz del árbol es la primera de arriba (o a la izquierda<sup>13</sup>) y los niveles están abajo (o a la derecha). Todos los nodos terminales, excepto el nodo raíz, son ejemplos de posibles metas.

<sup>13</sup> También se construye un árbol de izquierda a derecha (el ejemplo de la figura 2.8 está arriba-abajo).

### **{PRIVADO }2.5.3. Objetos**

La representación de conocimiento orientada a objetos suministra una alternativa, pero cierra relativamente la metodología para representar conocimiento que comparten un conjunto de estructuras como las redes semánticas y los marcos. Esta representación es vista en términos de un conjunto de objetos, los cuales son capaces de seguir ciertas conductas. Cada objeto se sitúa jerárquicamente en una red y puede acceder propiedades e información de los objetos de un nivel superior.

Los argumentos hechos para los objetos como estructuras de conocimiento se relacionan con el estilo de descripción inicialmente usado, más que las propiedades fundamentales de objetos. El estilo de descripción de objetos es un proceso de comparación, así que un nuevo objeto se describe para determinar la forma en que sea similar o diferente a los objetos ya conocidos.

En el paradigma orientado a objetos, éstos se comunican unos con otros por envío y recibo de mensajes.

- Un objeto, muy similar a los marcos, recibe mensajes de verificación de conocimiento y decide qué acción tomar.
- La descripción para comparar muchas de las similitudes es mostrar una especialización en casos usando los marcos.
- Los objetos usan atributos que están especificados en ranuras (*slot*, en inglés); se organizan jerárquicamente de manera que los de menor nivel comparten propiedades con los del nivel mayor (herencia).
- Los objetos contienen métodos, mientras los marcos tienen procedimientos asociados.
- Los marcos pueden usar reglas para realizar cálculos.

Es necesario aclarar un conjunto de definiciones, con el fin de hacer más explícito el concepto de representación orientada a objetos.

**Objeto.** Un objeto es una abstracción de un elemento de la realidad. Es una entidad independiente, compuesta por datos (atributos) y procesos (objetos).

**Atributo.** Es una característica del elemento de la realidad, que se pretende representar, la cual resulta de interés en el mundo del problema. Un atributo tiene un nombre y un valor. El valor es otro objeto que representa la situación de esa característica, en un momento dado. El estado de un objeto es la tupla de valores de sus atributos<sup>14</sup>, dado por:

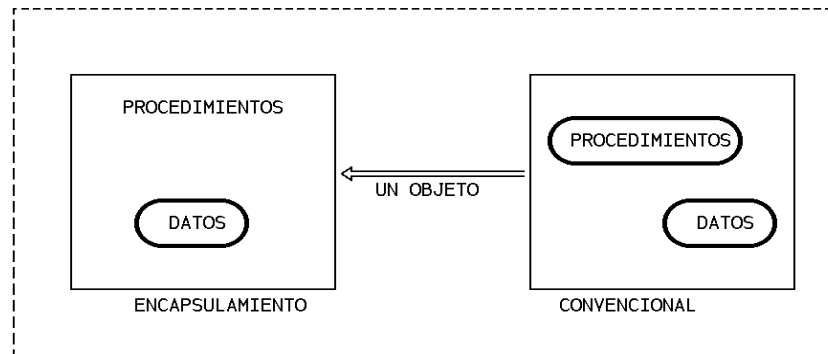
ATRIBUTO (variable\_caso, miembro)

---

<sup>14</sup> En los lenguajes de programación orientada a objetos, el polimorfismo se presenta como un resultado natural de la relación es\_un y de los mecanismos de paso de mensajes y relación.

**Mensaje.** Estímulo que recibe un objeto receptor de otro objeto del modelo, para que genere una reacción. Estas reacciones se llaman métodos.

**Método.** Está compuesto por mensajes definidos por sus propios atributos, para



**Figura 2.9** Organización convencional versus encapsulamiento.

simular la reacción que tendría el elemento del mundo que representa. Son dependientes del dominio o problema concreto que está tratándose, se activan automáticamente al recibir un mensaje. Esta propiedad de un objeto se denomina encapsulamiento, dado por:

MÉTODO (función\_miembro, rutina, operación)

Existen dos clases básicas de métodos: los modificadores, que cambian el estado del receptor y los analizadores, que sólo extraen información del objeto para contestar mensajes.

**Encapsulamiento.** Un objeto del mundo se representa mediante una variable principal que también a su vez se compone de variables y de funciones miembros, de manera que se pueda afirmar que está encapsulado. Un objeto encapsulado trabaja como una caja negra, es decir, las variables son independientes del diseño de las funciones que las manejan.

**Protocolo.** Es el conjunto de mensajes a los cuales sabe responder un objeto; está dado por:

PROTOCOLO (signatura, interfaz).

**Clase.** Una clase es un patrón para un objeto<sup>15</sup>. Todo objeto es un ejemplo (caso) de una clase. En el momento de reaccionar un objeto, revisa las características de su clase y toma de allí el método adecuado. Una clase especifica la estructura y el comportamiento de cada uno de sus casos; por eso, cada objeto tiene su propio conjunto de atributos, aunque comparte los métodos de las demás casos de la misma clase; está dada como:

CLASE (tipo).

<sup>15</sup> Una clase es un tipo y un objeto es un ejemplo de clase.

**Atributo de clase.** Un atributo de clase es el que indica las características de una misma clase. Esto implica que si algún objeto lo modifica, queda modificado para todos los casos de dicha clase. Está dado por:

ATRIBUTO\_DE\_CLASE (variable\_de\_clase, miembro).

**Herencia.** La herencia se basa en propiedades que los miembros de un grupo comparten. La herencia les permite a los objetos organizarse en taxonomías donde los objetos especializados heredan las propiedades de objetos más generales. Las clases de objetos comparten propiedades y funciones que pueden estar modeladas en alguna superclase, las que definen las partes comunes y derivan clases especializadas (subclase).

Ejemplo

```
Clase Criatura
  Propiedades
    tipo: string;
    peso: real;
    hábitat: (...algún hábitat...);
  Operaciones
    creación() ->criatura;
    predador(criatura)->conjunto(criatura);
    expectativa_de_vida(criatura) ->Entero;
  ...
```

**Fin** criatura.

```
Clase mamífero
  Herencia criatura;
  Propiedades
    período_de_gestación: real;
  Operaciones
  ...
```

**Fin** mamífero.

```
Clase persona
  Herencia mamífero;
  Propiedades
    alias,primer_nombre: string;
    fecha_nacimiento: fecha;
    origen: País;
```

**Fin** persona.

```
Clase hombre
  Herencia persona;
  Propiedades
    esposa: mujer;
  ...
```

**Fin**      hombre.

**Clase**    mujer

**Herencia** persona;

**Propiedades**

              esposo: hombre;

              ...

**Fin**      mujer.

En el ejemplo, la clase mamífero se define como una subclase de criatura; la clase persona, como una subclase de mamífero; hombre y mujer, como una subclase de persona.

## {PRIVADO }2.6. MARCOS

Un marco (*frame*, en inglés) es una estructura de datos que posibilita "hablarle" al computador e incluye conocimiento sobre un objeto en particular. Tal conocimiento se organiza y con unos métodos para almacenamiento y recuperación de conocimiento permite un diagnóstico independiente del conocimiento, e incluso permite la combinación con la lógica de predicados para generar un razonamiento; llega así a crear un motor de inferencia sobre el conocimiento y además permite realizar construcciones jerárquicas basadas en la herencia.

Ejemplo.

Conferencia

    Fecha:

    Lugar:

        Tema:

        Expositor:

        Hora:

Los marcos son básicamente principios de Programación Orientada a Objetos (POO) pero aplicados en IA, porque su desarrollo se soporta con la idea básica que la conducta humana se caracteriza por tomar estándares y proceder en situaciones familiares para generar conocimiento a partir de inferencias.

Los marcos ofrecen una representación concisa y estructurada del conocimiento en una forma natural. Los marcos<sup>16</sup> constituyen de hecho un mecanismo general de

---

<sup>16</sup> Para entender mejor el nombre de marco conviene recordar que el término original *frame* tiene muchos significados, condición o estado, armazón, estructura básica alrededor de la cual se construye "algo", conjunto de circunstancias que rodean un suceso, imagen en un rollo de película y de forma más general, los objetos en que se centra la atención de una cámara de cine. Realmente *frame*, en el sentido que le da Minsky, toma un poco de todos estos significados y por eso, al traducirlo por "marco" estamos

*Representación del conocimiento*

representación de conocimiento, ya que permiten usar el poder de los objetos incluyendo a otros procesos y mecanismos. En contraste con otros métodos de representación, los valores que describen un objeto están agrupados en una sola unidad. Así, un marco abarca objetos complejos, situaciones completas, o el manejo de problemas en una sola entidad. El conocimiento se particiona dentro de un marco por ranuras. Una ranura puede tener conocimiento declarativo o procedimental.

*Un marco es una estructura de datos para representar una situación estereotipada. Cada marco lleva asociadas varias clases de información: cómo usar el marco, qué se espera que suceda a continuación, qué hacer cuando no se confirman tales expectativas.*

Cada marco se caracteriza por un conjunto de *campos o slots* que sirven para identificar los marcos. Los marcos están especialmente concebidos para tareas de reconocimiento. El mecanismo es el siguiente, la información recibida hace que se activen unos marcos y esta a su vez provoca la activación de otros marcos conectados con los primeros, dando lugar así a una *red de activación*, cuyo objetivo es predecir y explicar la información que se va a encontrar en esa situación. Este reconocimiento basado en expectativas se denomina a veces *reconocimiento descendente*. Otra de las ideas novedosas de Minsky es la posibilidad de tener distintos marcos para definir una misma entidad desde distintos *puntos de vista*. El propósito de Minsky consistía en dar sugerencias e ideas más que en concretar los detalles de un cierto tipo de representación.

Los marcos tienen algunas capacidades que es importante resaltar, debido al gran uso que está dándoseles en la actualidad; estas capacidades se resumen en:

- a. Habilidad para documentar información acerca de un modelo; por ejemplo, maquinaria de una planta y atributos asociados.
- b. Habilidad para restringir posibles valores que un atributo puede tomar.
- c. Modular la información, permitiendo facilidad de expandir y mantener el sistema.
- d. Sintaxis consistente y fácil de leer, para referenciar objetos del área.
- e. Restringir la secuencia de hechos considerados durante encadenamientos.
- f. Posibilitar acceso a mecanismos que soportan la herencia de información.

Los marcos se usan para representar conocimiento estereotipado o construido a partir de experiencias y características bien conocidas. Todos tenemos gran cantidad de

---

*perdiendo casi todos los matices del término original. También conviene señalar que, según Minsky, cada marco posee varios slots o terminals que significan "ranura", es decir, un "hueco" destinado a contener "algo que encaja allí". Un marco posee conocimiento sobre un objeto, evento, lugar, situación u otro; un marco ofrece los medios para organizar conocimiento en ranuras que contienen características y atributos. En forma física, un marco es algo como un recuento de categorías y subcategorías; el marco puede describirse con gran detalle; el detalle se da en forma de ranuras que describen varios atributos o características del objeto o situación.*

conocimiento de sentido común y experiencias guardadas en el cerebro para analizar un objeto o experiencia nueva al resolver un problema. La organización del conocimiento jerárquicamente busca acople estructural a como se almacena, modifica y recupera conocimiento de forma natural. La jerarquía y la herencia están en evolución, en la forma de crear y almacenar conceptos, en el propio conocimiento que se quiere modelar y en las configuraciones subjetivas que se derivan del modo humano de organizar su percepción del mundo que lo rodea, clasificando los sucesos y situándolos en una estructura multicapa organizada en clases de conceptos.

### Ejemplo

**Frame:** Empleado

**Hereda de:** persona

Número\_empleado: **Valor:**xx

SI adiciona: verifique secuencia: número

Salario **Valor:**yy

SI necesita: verifique palabra clave: Clav

**Frame:** Luis C. Torres S.

**Hereda de:** Empleado

Número\_empleado **Valor:** 1320

Salario **Valor:** 1 475000

Los sistemas actuales basados en marcos se basan sobre todo en la herencia a partir de una red jerárquica. El punto de partida consiste en la creación de una red de marcos. Cada marco representa un concepto o una clase, y cada objeto (instancia) representa un elemento dentro de la clase. En algunas herramientas cada caso (ejemplo, instancia) puede pertenecer a varios marcos. Cada marco hereda los campos de todos sus antepasados en la red, y cada objeto hereda campos y valores de todos los marcos a los que pertenece. Algunas herramientas sólo permiten *herencia descendente*, mientras que otras admiten *herencia bidireccional* (el valor asignado a un objeto puede convertirse en valor por defecto del marco al que pertenece).

### Facetas

Definen las propiedades de un campo:

- *Valor por defecto*: es el valor que toma el campo (salvo que se indique lo contrario). En general se asigna al marco y es heredado por todas los casos creados posteriormente dentro de él.
- *Multivaluado*: indica si el campo es univaluado (la introducción de un nuevo valor desplaza el anterior) o multivaluado (pueden coexistir varios valores simultáneamente).
- *Restricciones*: limitan el conjunto de valores que puede recibir el campo. El sistema reconoce un error de asignación y toma las medidas oportunas.
- *Certeza*: indica la credibilidad del valor asignado al campo (es semejante al factor de certeza asociado a cada proposición dentro de un sistema basado en reglas).
- *Facetas de interfaz*: contienen texto, preguntas y mensajes destinados a la



*Representación del conocimiento*  
interacción con el usuario.

Una de las características más importantes de los marcos es la posibilidad de asociar procedimientos llamados *demonios* a cualquier elemento de la red jerárquica, los cuales se ejecutan según las funciones especificadas. Algunos sucesos son:

- a. De modificación. Siempre que varíe el valor del campo.
- b. De borrado. Al ser eliminado el valor de un campo.
- c. De necesidad. Cada vez que se requiere saber cuál es el valor de un campo y éste no tiene ningún valor asignado.
- d. De acceso. Cuando se solicita el valor de un campo, aunque ya tuviera un valor asignado.

Los marcos pasan a ser considerados como *objetos* autónomos que tienen la capacidad de comunicarse entre ellos mediante el paso de mensajes.

Ejemplo.

Suceso:

-lugar:

-día:

-hora:

Desastre natural:

-muertos:

-heridos:

-personas sin vivienda:

-daños materiales:

Terremoto:

-falla:

-magnitud:

Inundación:

-río:

Huracán:

-nombre:

-velocidad del viento:

Acontecimiento deportivo:

deporte:

-ganador:

-resultado:

Acontecimiento social:

-anfitrión:

-número de invitados:

Boda:

-novia:

-padres novia:

-novio:

-padres novio:

Cumpleaños:

-nombre:

-edad:  
                     Julio Guarín      -caso de:  
                     -40  
                     Claudia Tovar    -caso de:  
                     -22  
 Acontecimiento político:  
 -político:  
 -actividad:  
                     Miguel García    -caso de:  
                     -reunión plaza CHE

Algunas herramientas avanzadas permiten que un campo tenga varios valores correspondientes a distintos puntos de vista (el término original es *views*).

Veamos un ejemplo adaptado del sistema FRAME-KIT:

(definir-caso carne-de-vaca  
 (comestible (valor            (general sí)  
    (hindú no))))

Su objetivo (mecanismo de control del razonamiento) es similar al que pretenden las *perspectivas* que aparecen en el lenguaje KRL<sup>17</sup>, aunque la forma de implementación es completamente diferente.

La determinación de cuál representación utilizar, ya se anotó, depende no sólo del conocimiento sino de la herramienta de programación (el lenguaje de programación o el Shell) que se emplee; si existe conocimiento puede representarse de diferentes maneras.

Ejemplo de marco:

Clase Vaso\_sanguíneo es:  
                     subclase de nill;  
                     contiene : sangre;  
                     forma : tubular;

---

<sup>17</sup> El propósito fundamental de KRL (*Knowledge Representation Language*) consistía en ofrecer los mecanismos necesarios para una **representación estructurada** del conocimiento. Por ello la información se agrupa en torno a los objetos o unidades, **units**, que constituyen el eje fundamental de la representación del conocimiento. Los campos de una unidad pueden tener **valores por defecto**, que provienen de la **herencia de propiedades** o bien de la existencia de **prototipos** (unidad que en vez de corresponder a un objeto concreto representa un elemento típico de una clase). Otro mecanismo de razonamiento muy importante es la **comparación o ajuste** ("**matching**", similar a la comparación de patrones de las reglas). Entre los objetivos básicos de KRL es la eficiencia, a costa de perder elegancia desde un punto de vista teórico. Una muestra de ello es la **redundancia** (incluir explícitamente cierta información para no tener que deducirla). Además, ofrece numerosas posibilidades de controlar el razonamiento. Una de ellas es la **asignación de procedimientos o procedural attachment** a los marcos o a sus campos los cuales pueden ser de dos tipos: los **sirvientes** (que son procedimientos activados por el marco para alcanzar cierto objetivo) y los **demonios** (se activan automáticamente en ciertas circunstancias).

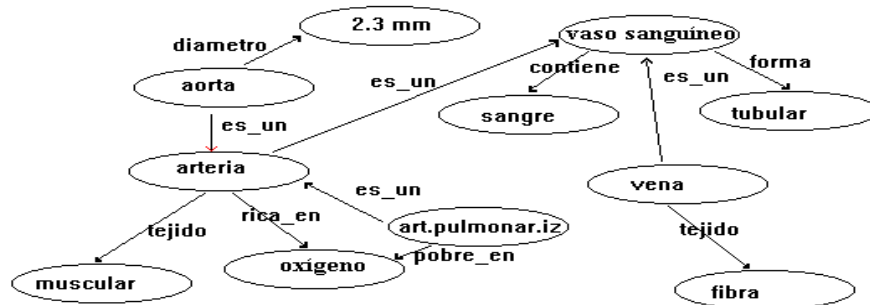
*Representación del conocimiento*

```

fin
Clase Arteria es:
    subclase de: Vaso_sanguíneo;
    tejido : muscular;
    pobre en : oxígeno;
fin
Clase Vena es:
    subclase de: Vaso_sanguíneo;
    tejido : fibra;
    rica en : oxígeno;
fin
Ejemplo Aorta es
    caso de: Arteria;
    diámetro : 2,3 mm;
fin
Ejemplo Arteria pulmonar izq. es
    caso de: Arteria;
    diámetro : 1,8 mm;
fin
Ejemplo Cava superior es
    caso de: Vena;
    diámetro : 1,7 mm;
fin

```

La red semántica que representa los marcos anteriores no son al azar, sino que deben seguir la herencia establecida entre los diferentes marcos.



**Figura 2.11** Red semántica.

En representación formal:

```

arteria(aorta)
diametro(aorta, 2,3 mm)
diametro(cava_superior, 1,7 mm)
diametro(arteria_pulmonar_izq, 1,8 mm)
arteria(arteria_pulmonar_izq)
vena(cava_superior)

```

```

pobre_en(arteria, oxígeno)
rica_en(vena, oxígeno)
 $\forall x$  arteria(x)  $\Rightarrow$  tejido(x, muscular)
 $\forall x$  arteria(x)  $\Rightarrow$  rica_en(x, oxígeno)
 $\forall x$  arteria(x)  $\Rightarrow$  vaso_sanguíneo(x)
 $\forall x$  vena(x)  $\Rightarrow$  tejido(x, fibra)
 $\forall x$  vena(x)  $\Rightarrow$  vaso_sanguíneo(x)
 $\forall x$  vaso_sanguíneo(x)  $\Rightarrow$  contiene(x, sangre)
 $\forall x$  vaso_sanguíneo(x)  $\Rightarrow$  forma(x, tubular)

```

Existen básicamente dos modos en que los marcos han sido y son utilizados:

### Patrones para la comparación o pattern matching:

Cuando la información disponible indica que el problema planteado se ajusta a un marco, el resto de la información contenida en dicho marco nos permite inferir la presencia de otros elementos. El proceso de razonamiento puede resumirse en: activación, reconocimiento y predicción.

### Almacenes de información:

Aquí, los marcos significan ante todo una forma estructurada de almacenar información. La distinción entre clases y objetos y su organización en forma de red jerárquica permite utilizar la herencia de propiedades como mecanismo de razonamiento por defecto. La capacidad de procesamiento asociada a cada marco varía notablemente de un sistema a otro. En un extremo, los marcos constituyen una especie de armario en cuyos cajones (los campos) se almacenan los datos (semejante a una base de datos relacional). En el otro extremo, son los propios objetos quienes realizan la inferencia (programación orientada a objetos).

La principal ventaja de los marcos frente a la lógica es que representan el conocimiento de forma estructurada. Así, toda la información relativa a un objeto queda reunida en un marco en vez de estar dispersa en un conjunto de proposiciones sin ninguna estructura, y esto significa mayor eficiencia de cara al razonamiento y mayor facilidad para mantener la base de conocimiento.

**Nombre:** Síndrome nefrótico

**Es-un-tipo-de:** Estado clínico

**Hallazgo:** Concentración de albúmina baja

...

**Hallazgo:** Lípidos en la orina presentes

**No-debe-haber:** Proteinuria ausente

**Es-suficiente:**

Edema masivo junto con proteinuria > 25g/24 horas

**Criterios mayores:**

Concentración de sero-albúmina

...

Proteinuria

*Representación del conocimiento*

...

**Criterios menores:**

Concentración de colesterol

...

Lípidos en la orina

...

**Puede-estar-producido-por:**

Glomerulonefritis

...

**Puede-complicarse-por:**

Hipovolemia

Celulitis

**Puede-producir:**

Retención de sodio

**Diagnóstico-diferencial:**

Si presión venosa yugular elevada,

Considerar: pericarditis constrictiva.

Si ascitis presente,

Considerar cirrosis.

## **{PRIVADO }2.7. AGENTES INTELIGENTES**

Un agente puede considerarse como aquel objeto que percibe su ambiente y que responde o actúa en dicho ambiente. Percibe por medio de sensores; en los humanos, los sensores son: ojos, oídos, olfato, tacto,... Respondemos por medio de efectores como: manos, pierna, boca. En una máquina, estos efectores son motores; en un programa son acciones. Un agente robótico tiene como sensores cámaras y telémetros infrarrojos, y los motores le sirven como efectores.

Un agente racional<sup>18</sup> es aquel que hace lo correcto (de acuerdo con ciertos parámetros o medidas de desempeño especificadas). Para determinar lo correcto hay que definir:

\*Cuál es el criterio que sirve para definir qué tan exitoso es un agente.

\*Cuándo evaluar el desempeño.

Hay que considerar que sería imposible diseñar un agente para que satisfaga adecuadamente la realización de lo correcto siempre, a menos que conozcamos la solución a todos los problemas. Con esto se dice: un agente racional deberá emprender todas aquellas acciones que favorezcan obtener el máximo de su medida de rendimiento, basándose en el conocimiento y en las percepciones. El concepto de agente permite pensar en él como herramienta para el análisis de sistemas inteligentes.

Una vez que se ha visualizado que el comportamiento de un agente depende exclusivamente de la secuencia de percepciones en un momento dado, podemos

---

<sup>18</sup> Si un agente no puede percibir todos los elementos del entorno o no emprende una acción no programada no significa que no sea racional.

elaborar una tabla de acciones que podría realizar un agente a partir de un conjunto de percepciones. Especificar qué tipo de acción deberá emprender un agente como respuesta a una determinada secuencia de percepciones constituye el diseño de un agente.

Importante considerar si el agente debe basarse exclusivamente en el "conocimiento integrado". En tal caso, se dice que el agente no tiene autonomía, pero podemos decir que un sistema será autónomo en la medida en que su conducta está definida por su propia experiencia. Así como la evolución nos ha dotado de eventos que facilitan sobrevivir hasta aprender correctamente, también se crearán agentes inteligentes (artificiales) con cierto conocimiento inicial y de capacidad para aprender.

La búsqueda y la planificación son áreas de la IA que se ocupan de encontrar las secuencias de acciones que permiten alcanzar las metas de un agente.

La intención de la IA es diseñar agentes (programas) con autonomía, ejecutándose en algún hardware (arquitectura). En general, el hardware pone al alcance del agente percepciones obtenidas mediante los sensores; lo ejecuta y alimenta el efector con las acciones elegidas.

El ambiente plantea tantos desafíos que algunas veces resulta muy difícil para un software desempeñarse adecuadamente.

```
función agente(percepciones) retornar acción
  memoria:
  memoria <- actualizar_memoria(memoria, percepciones)
  acción <- escoger_mejor_acción(memoria)
  memoria <- actualizar_memoria(memoria, acción)
  retornar acción
```

**Ejercicio.** Construir agentes teniendo como base la función agente, antes definida. Sea un agente conducido por una tabla:

```
función agente_conducido_por_tabla(percepciones) retornar acción
  percepciones:
  añadir_percepción_al_final_de_la_percepciones
  acción <- consulta(percepciones, tabla)
  retornar acción
```

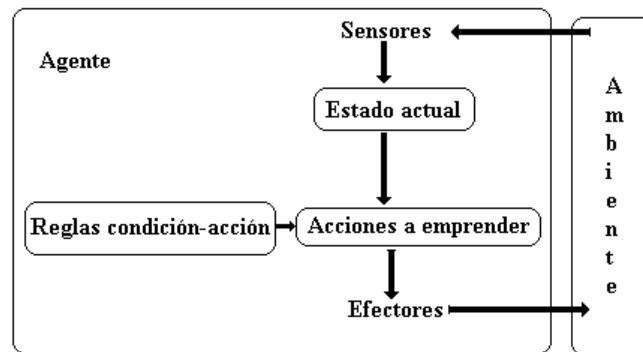
Consideremos que el agente es un taxista (o taxi). Él requiere saber cómo es el recorrido, quién más circula en la vía, a qué velocidad puede desplazarse. Claro que esta información la obtiene de percepciones que ofrecen: cámaras, velocímetro, contador. Sin olvidar que debe conocer el estado del vehículo, lo cual lo logra por medio de sensores. También contará con dispositivos que le permiten comunicarse con los pasajeros, quienes le informaran cuál es el destino.

Las acciones que ha de realizar este taxista deben concordar con las que efectúa un

*Representación del conocimiento*  
conductor humano: acelerar, frenar, virar,...

¿Cuál sería la medida de desempeño? Entre las medidas deseables es llegar al destino correcto, no realizar violaciones de tránsito ni molestias a otros conductores, ofrecer seguridad y comodidad a los pasajeros,...

Sin embargo, es necesario definir el ambiente en el cual se ha de movilizar. ¿Calles con huecos?, ¿vía circunvalar?, ¿autopista del norte?, ¿poco o mucho tránsito?, ¿con semáforos o sin semáforos?



0Figura 2.12. *Agente reflejo simple.*

Al mirar con mucho cuidado estas consideraciones, podemos proponer diversos tipos de programas agentes:

- \* Agentes de reflejo simple.
- \* Agentes bien informados.
- \* Agentes basados en metas.
- \* Agentes basados en utilidad.

### **Agentes de reflejo simple**

En general, estarían estructurados por un conjunto de reglas de condición-acción, tal como lo hacen los conductores humanos.

SI semáforo en amarillo ENTONCES empezar a frenar.  
SI carro adelante prende stop ENTONCES empezar a frenar.  
SI calle cerrada ENTONCES gire en dirección permitida.  
SI dirección permitida es derecha ENTONCES girar derecha.  
SI ambas direcciones permitidas ENTONCES girar lado inmediato.

En la figura 2.12 puede observarse cómo las reglas condición-acción le permiten al agente establecer la conexión entre percepciones y acciones.

El programa agente de reflejo simple sería:

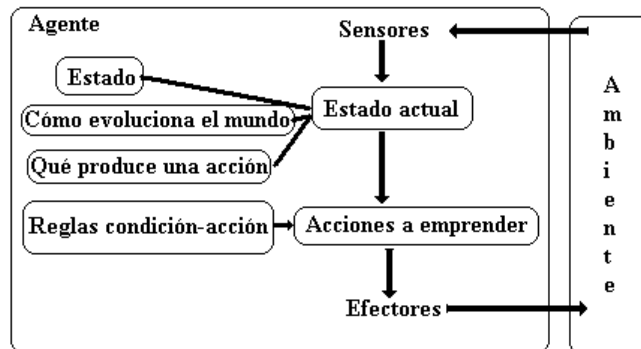


Figura 2.13. Agente reflejo con estado interno.

```

función agente_reflejo_simple(percepciones) retornar acción
percepciones:
reglas:
estado <- interpretar_entrada(percepciones)
regla <- regla_coincidencia(estados, reglas)
acción <- regla_acción(regla)
retornar acción

```

La función `interpretar_entrada` genera una descripción abstracta del estado prevaiente de la percepción; y la función `regla_coincidencia` produce la primera regla que hace juego con la descripción del estado ofrecida.

La figura 2.13 muestra la estructura del agente reflejo y también cómo se combinan las percepciones prevaientes con el estado interno anterior para generar la descripción actualizada del estado prevaiente.

### Agentes bien informados

Sólo tomando la decisión adecuada con base en la percepción de un momento dado, se entenderá que satisface el desempeño deseado.

Hay momentos en que el agente requiere actualizar algo de información en el estado interno que le permita discernir entre estados del mundo que generan la misma entrada de percepciones, pero que, son totalmente distintos; esto es, se requieren acciones distintas, lo cual indica requerirse cierta información sobre cómo evoluciona el mundo, independientemente del agente. También se requiere información sobre cómo las acciones del agente mismo afectan al mundo.

El programa agente reflejo bien informado podría ser:

```

función agente_reflejo_con_estado(percepciones) retornar acción
percepciones:
reglas:
acción:
estado <- actualiza_estado(estados, percepciones)

```



```

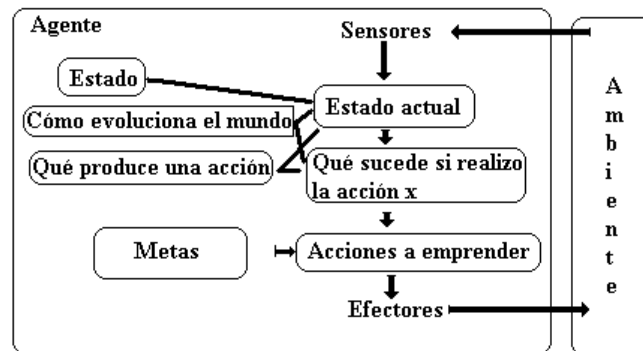
Representación del conocimiento
regla <- regla_coincidencia(estado, reglas)
acción <- regla_acción(regla)
estado <- actualizar_estado(estado, acción)
retornar acción

```

La función actualizar\_estado es la responsable de crear la nueva descripción del estado interno.

### Agentes basados en metas

En la regla, *SI calle cerrada ENTONCES gire en dirección permitida*, la decisión adecuada dependerá de a dónde desee llegar el taxista, cuál es la dirección permitida o cuál giro no obstaculiza ni comete algún error.



0Figura 2.14. Agente con metas explícitas.

El programa agente puede combinar lo anterior con la información relativa al resultado que producirán las posibles acciones que se emprendan y de esta manera alcanzar la meta deseada.

Un agente basado en metas razonaría considerando la forma como normalmente evolucionan las cosas del mundo.

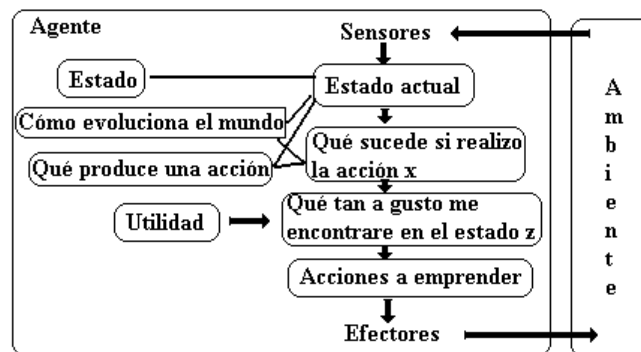
```

función agente_basado_metas(percepciones) retornar acción
percepciones:
acción:
estado <- actualiza_estado(estado, percepciones)
estado <- actualizar_estado(estado, meta)
retornar acción

```

### Agentes basados en utilidad

Las metas no bastan por sí mismas para generar una conducta de alta calidad, son muchas las acciones que permitirían que el agente llegue a la meta, pero de todas ellas, algunas son más rápidas, seguras y confiables. Estas decisiones llevan a los agentes a obtener mayor utilidad.



0Figura 2.15. Agente basado en utilidad.

La utilidad es una función que correlaciona un estado con un valor el cual caracteriza el grado de satisfacción (desempeño). La estructura general de un agente basado en utilidad se visualiza en la figura 2.15.

El criterio que sirve para definir qué tan exitoso es un agente en la elaboración de una tarea encomendada es la medida del desempeño; es decir, es el medio de evaluación de las acciones del agente. Sin embargo, no es posible utilizar una medida fija que pueda aplicarse por igual a todos los agentes, ya que depende de la tarea (problema, situación) y del criterio de cada evaluador.