

# Métodos de Inteligencia Artificial

---

L. Enrique Sucar (INAOE)

[esucar@inaoep.mx](mailto:esucar@inaoep.mx)

[ccc.inaoep.mx/esucar](http://ccc.inaoep.mx/esucar)

Tecnologías de Información

UPAEP

# Repaso

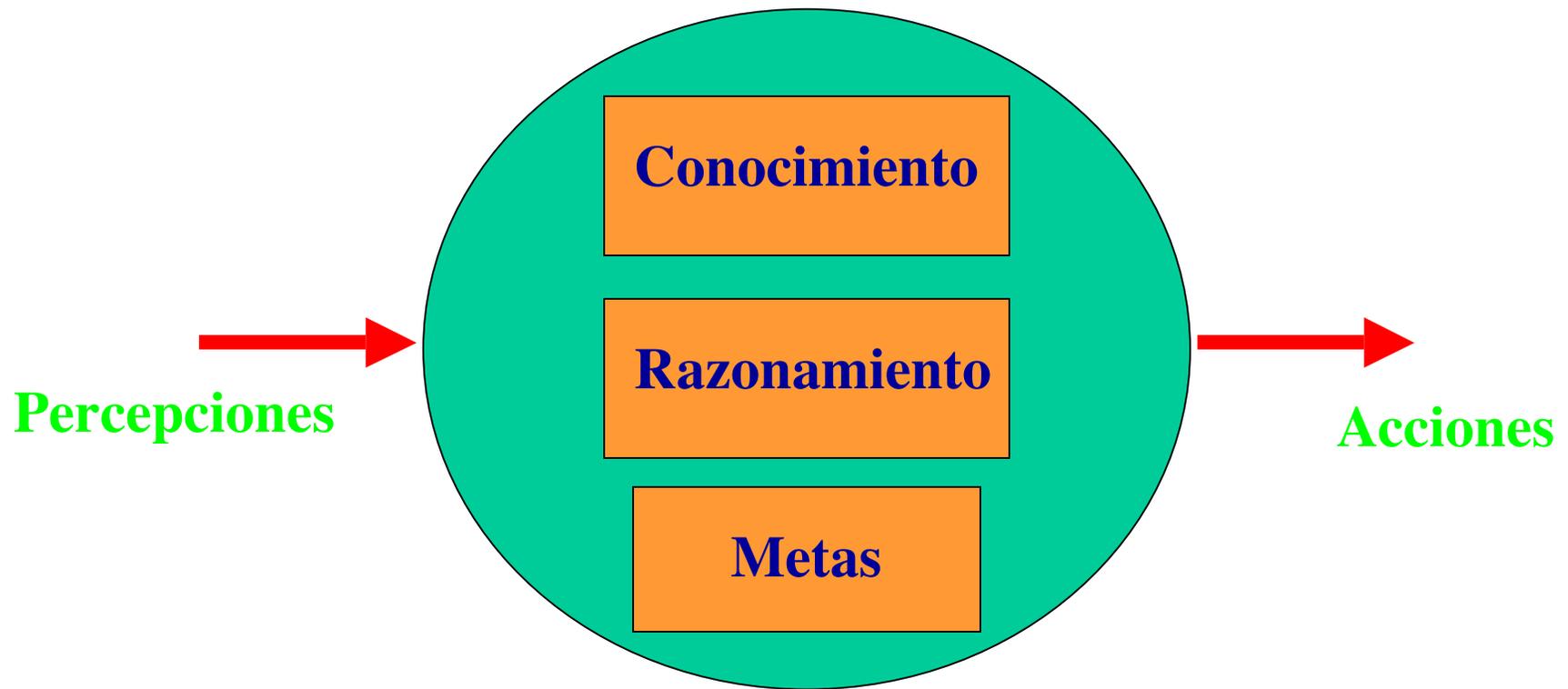
- Agentes Inteligentes
- Búsqueda
- Agentes basados en conocimiento
  - Lógica
  - Reglas
  - Representaciones estructuradas
  - Híbridas
- Agentes que manejan incertidumbre
  - Redes bayesianas

# Agentes

Un agente es cualquier ente que se pueda ver como *percibiendo* su ambiente a través de *sensores* y *actuando* en el ambiente por medio de *efectores*.

Un agente racional es un agente que hace “buenas” acciones. Para ésto, necesitamos saber cómo y cuándo evaluar al agente.

# Agente



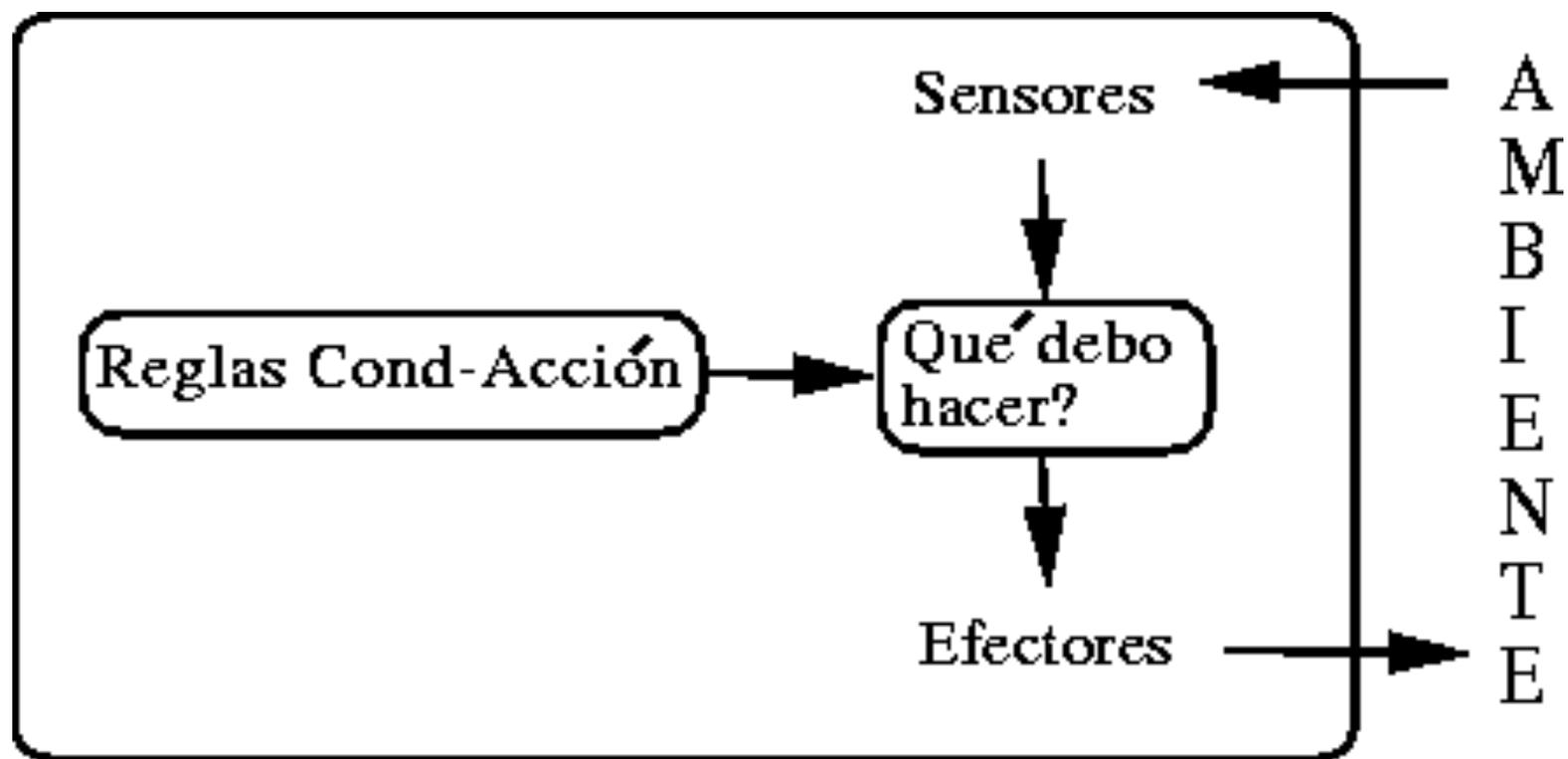
**AMBIENTE**

Para construir un mapeo entre percepciones y acciones podemos pensar en varios tipos de agentes.

***Agentes de reflejos simples:***

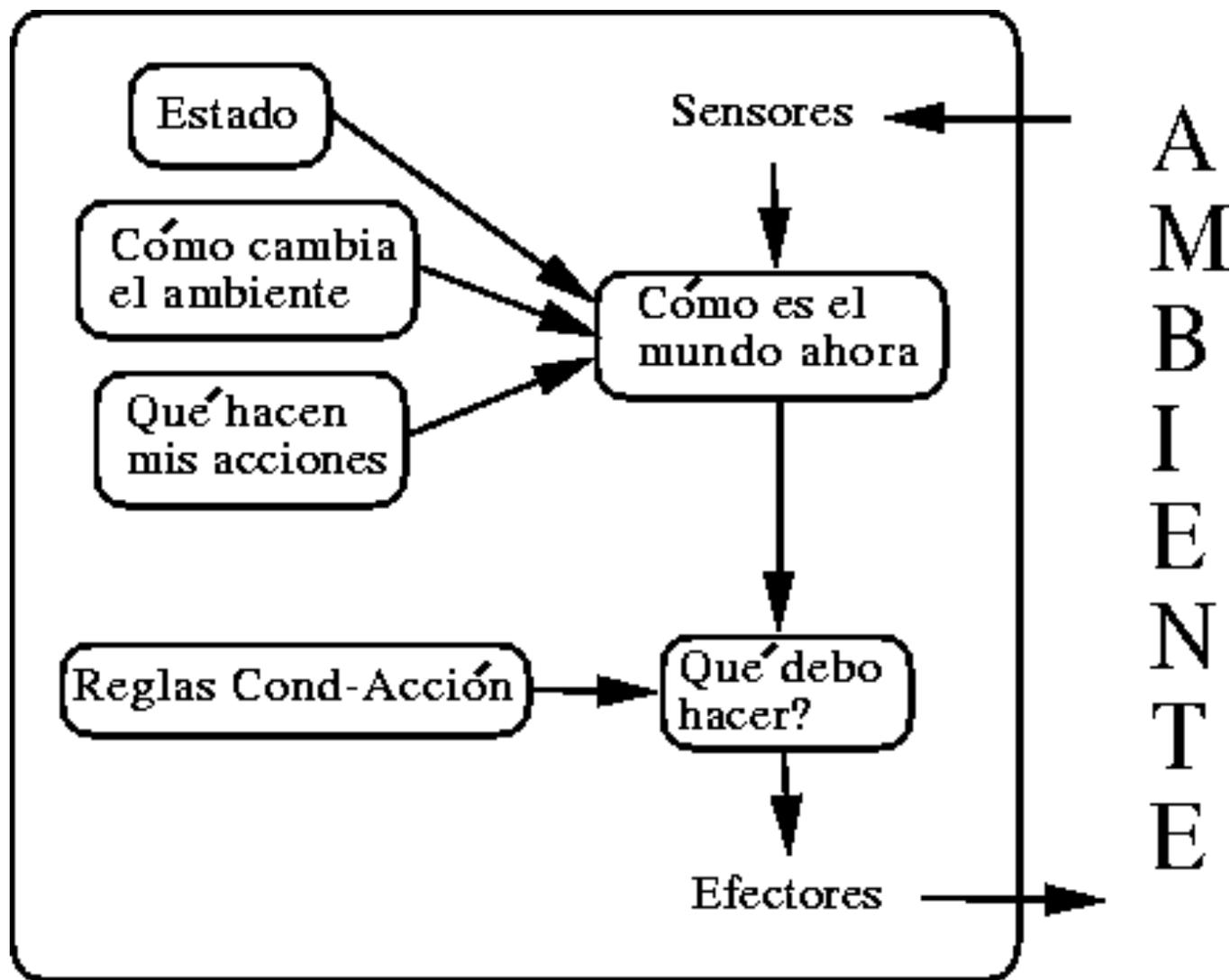
Construir una tabla es imposible, pero podemos considerar algunas asociaciones comunes en términos de reglas *situación-acción*. e.g.,

**IF**            el coche de enfrente está frenando  
**THEN**       empieza a frenar



## ***Agentes de reflejos simples y estado interno.***

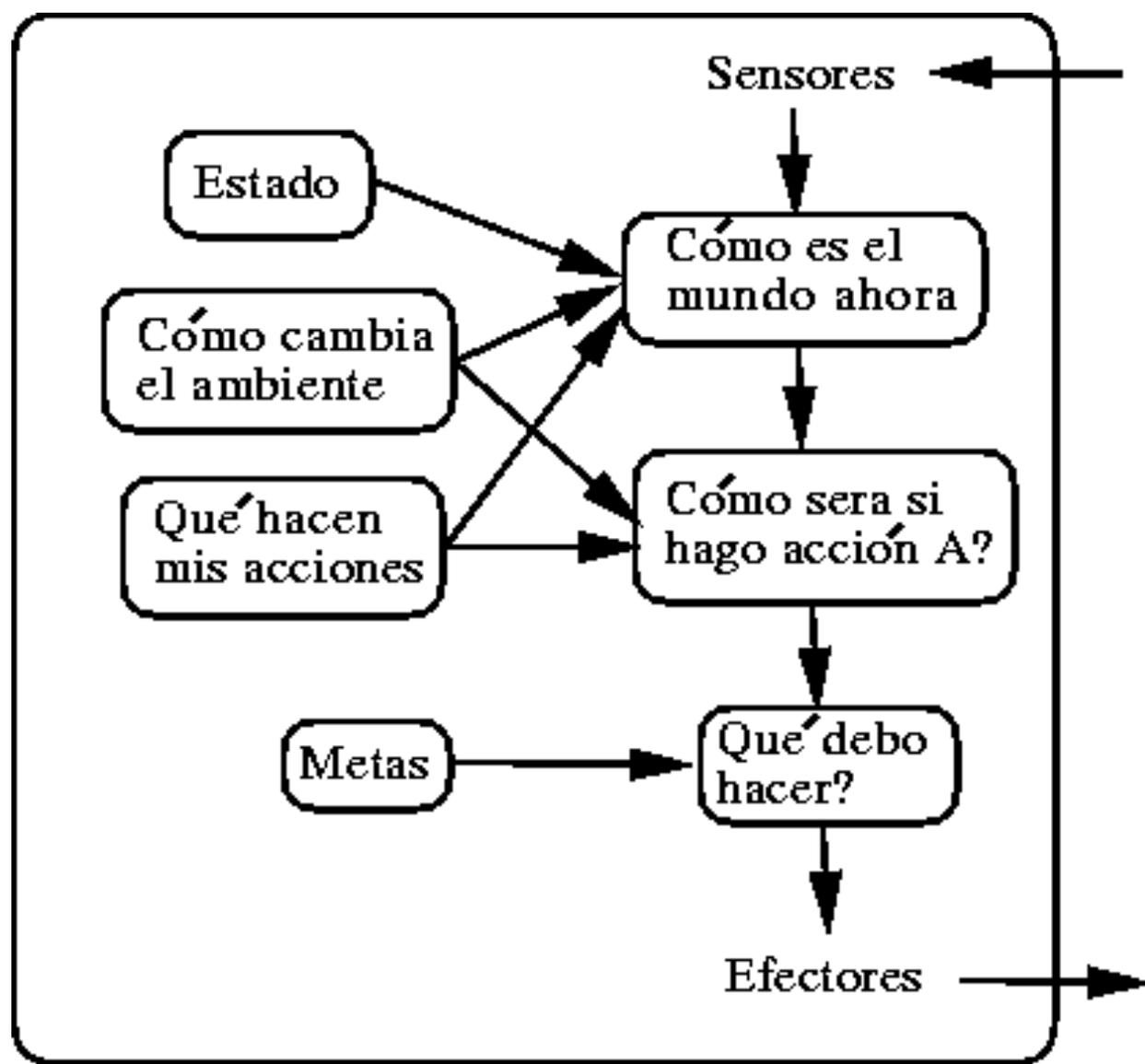
**Esto es muy simple, y normalmente se necesita un estado interno para (i) ver como cambia el ambiente independientemente del agente y (ii) como afectan las acciones del agente al ambiente.**



## ***Agentes con información de metas.***

**El agente necesita información de sus metas para escoger qué acciones las pueden cumplir (pueden usarse técnicas de búsqueda y planificación).**

**Esto lo puede hacer más flexible (v.g., si está lloviendo ajustar la efectividad de los frenos).**

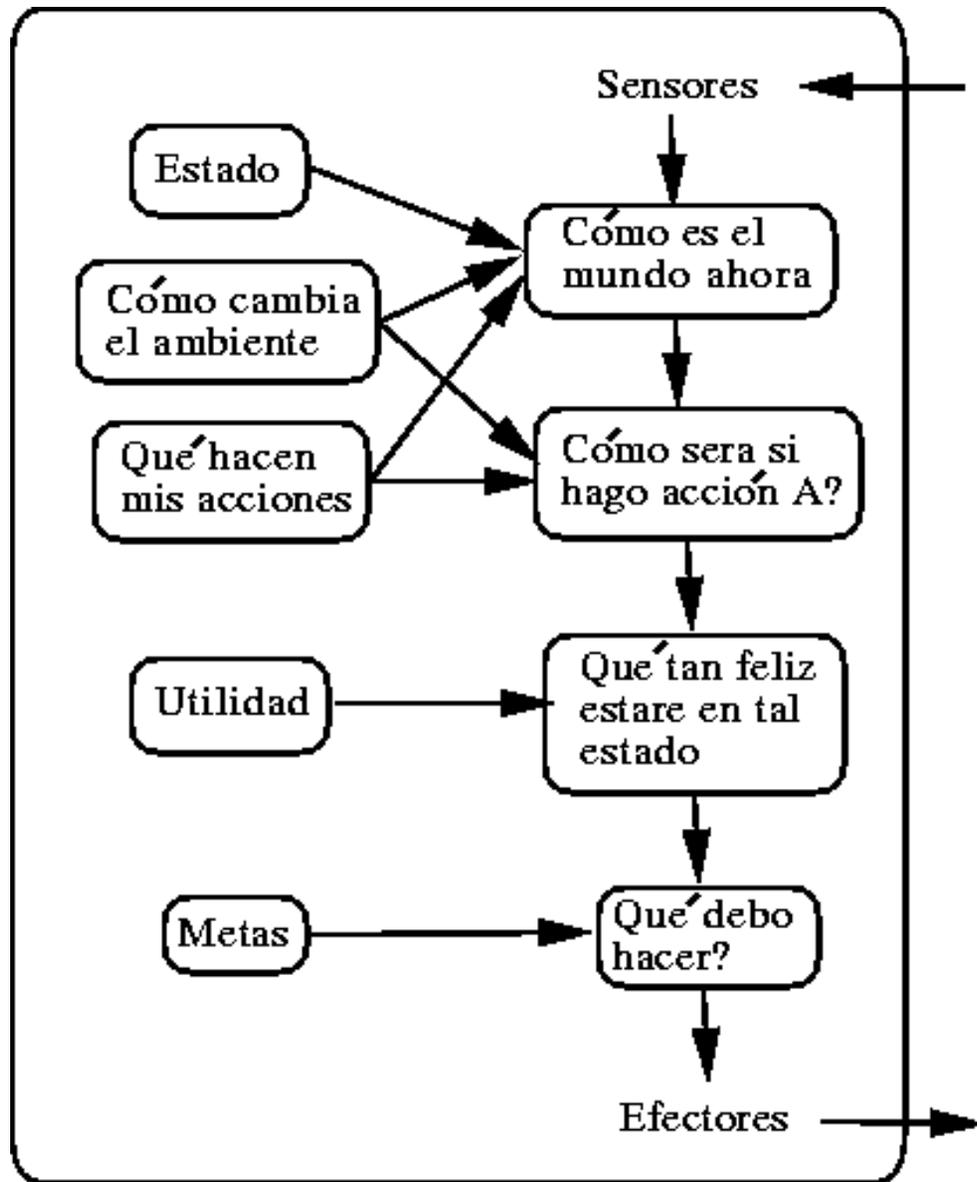


A  
M  
B  
I  
E  
N  
T  
E

## ***Agentes con medida de utilidad.***

**Las metas por si solas no son suficientes para generar un comportamiento de buena calidad.**

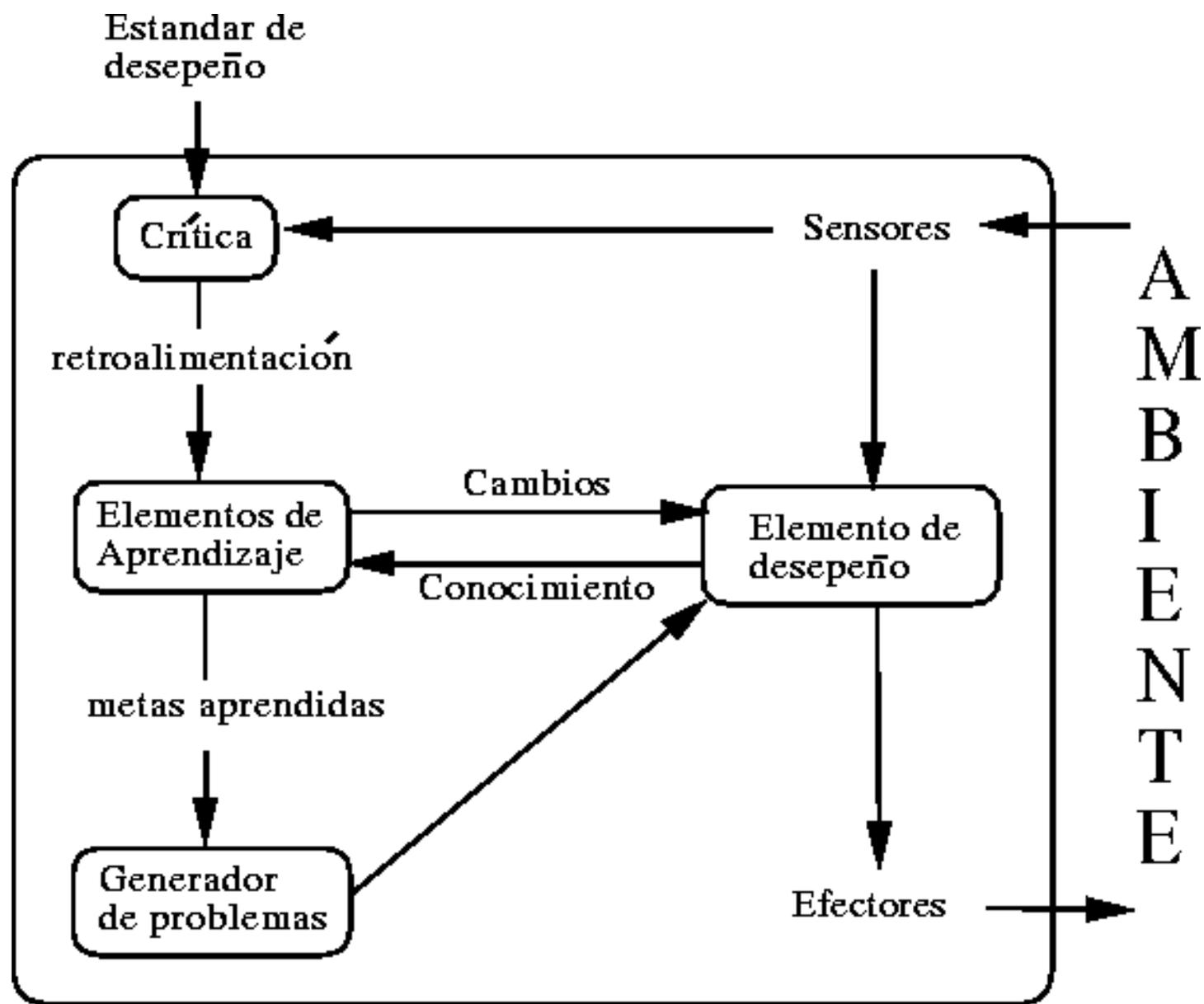
**Para esto necesitamos una medida de utilidad (función que mapea un estado o secuencia de estados con un número real).**



A  
M  
B  
I  
E  
N  
T  
E

## ***Agentes con aprendizaje.***

**La idea es que las percepciones no se usen sólo para actuar, sino también para mejorar su desempeño en el futuro.**



# Búsqueda

- Representación
- Tipos búsquedas:
  - Sin información
  - Con información
  - Óptimas
- Análisis de complejidad

# Solución de Problemas

## Asociado a la inteligencia

- Identificación y definición del problema
- Identificación del criterio de evaluación
- Generación de alternativas

Solución de muchos problemas en IA:  
básicamente búsqueda y evaluación

# Representación del espacio de estados

- define un espacio de estados (explícito / implícito)
- especifica los estados iniciales
- especifica los estados finales (metas)
- especifica las reglas que definen las acciones disponibles para moverse de un estado a otro

## Representación de espacio de estados

En este contexto:

El proceso de solución de problemas =  
encontrar una secuencia de operaciones  
que transformen al estado inicial en uno  
final

Se requiere una estrategia de búsqueda

## Propiedades de algoritmos de búsqueda (heurísticas):

1. *Completo*: si encuentra una solución cuando ésta existe
2. *Admisible*: si garantiza regresar una solución óptima cuando ésta existe

## Propiedades de algoritmos de búsqueda (heurísticas):

3. *Dominante*: un algoritmo  $A1$  se dice que domina a  $A2$  si todo nodo expandido por  $A1$  es también expandido por  $A2$  (“más eficiente que”)

4. *Óptimo*: un algoritmo es óptimo sobre una clase de algoritmos si domina a todos los miembros de la clase

# Procedimientos de Búsqueda

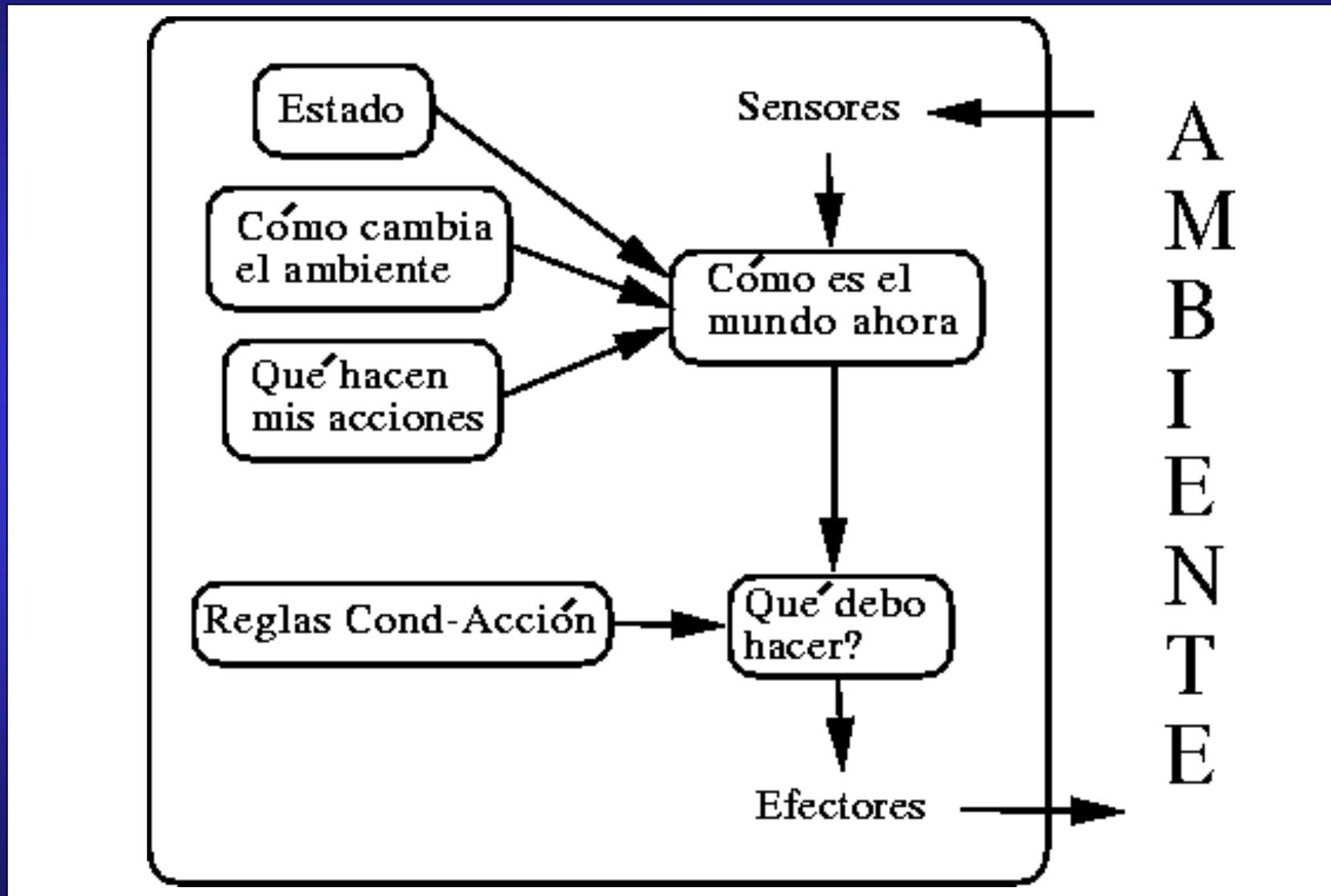
## Algún camino:

- Sin información:
  - depth-first (en profundo)
  - breadth-first (a lo ancho)
- Con información:
  - hill climbing
  - beam search
  - best first

- El mejor camino (óptimo):

- British museum
- branch and bound
- A\*

# Agentes basados en conocimiento



# Contenido

- Sistemas basados en conocimiento
- Partes principales
- Consideraciones
- Ontologías
- Tipos de representación

## Definición

Un sistema basado en conocimiento se puede definir como:

“...sistema que resuelve problemas utilizando una representación simbólica del conocimiento humano” [Jackson 86].

Tienen una separación entre:

1. Conocimiento específico del problema
  - Base de Conocimiento
  
2. Metodología para solucionar el problema
  - Máquina de Inferencia

# Características importantes

- Representación explícita del conocimiento
- Capacidad de razonamiento independiente de la aplicación específica
- Capacidad de explicar sus conclusiones y el proceso de razonamiento

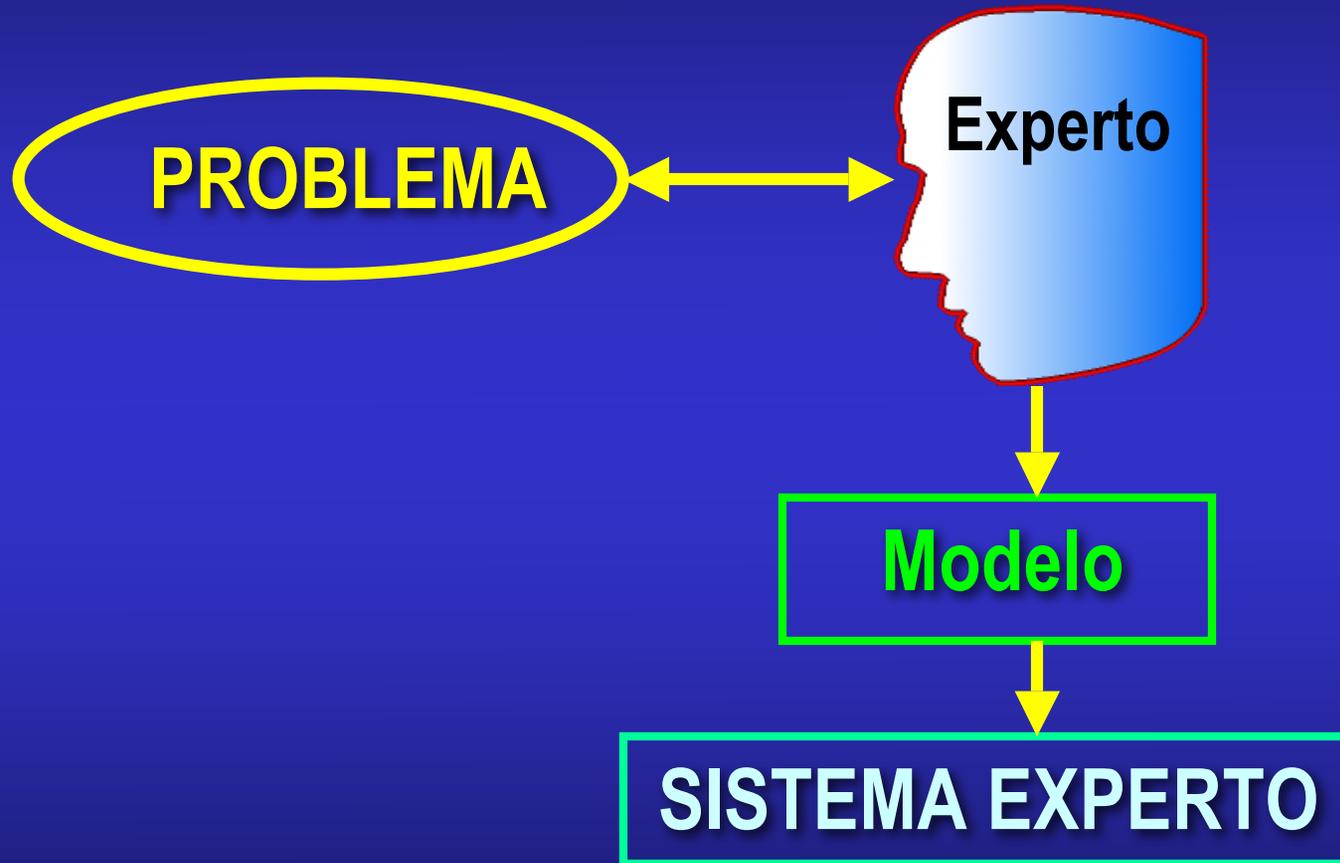
# Características importantes

- Alto rendimiento en un dominio específico
- Uso de heurísticas vs. modelos matemáticos
- Uso de inferencia simbólica vs. algoritmo numérico

# Importancia del Conocimiento

Basan su rendimiento en la cantidad y calidad del conocimiento de un dominio específico y no tanto en las técnicas de solución de problemas.

# Enfoque de Inteligencia Artificial



# Componentes básicos

1. Base de Conocimiento (BdeC)
2. Máquina de Inferencia
3. Memoria de Trabajo
4. Interfaz de Usuario
5. Interfaz de Adquisición

# Arquitectura

adquisi-  
ción

usuario

Base de Conocimiento

Máquina de Inferencia

Memoria de Trabajo

## Lógica:

Lenguaje que permite expresar conocimiento y razonar a partir de ciertas expresiones para deducir otras (deducción)

## Características:

- sintaxis y semántica bien definidas
- reglas de inferencia

# Lógica Proposicional

Permite expresar y razonar con declaraciones que son o verdaderas o falsas

Ejemplos:

- la clase de IA es lo mejor que me ha pasado en mi vida
- lógica es fácil

# Lógica Proposicional

Este tipo de declaraciones se llaman *proposiciones* y se denotan en lógica proposicional con letras mayúsculas

(v.g.,  $P, Q, \dots$ )

$P$ 's y  $Q$ 's también se llaman *proposiciones atómicas o átomos*

Los átomos se pueden combinar con *conectores lógicos* (dando proposiciones compuestas)

negación:  $\sim$ ,  $\neg$

conjunción:  $\&$ ,  $\wedge$

disjunción:  $\vee$

implicación:  $\supset$ ,  $\rightarrow$

doble implicación:  $\leftrightarrow$

Por ejemplo,

$G$  = “esto ya lo ví”

$D$  = “me estoy aburriendo”

$G \wedge D$  = “esto ya lo ví” Y “me estoy aburriendo”

Sólo algunas combinaciones de átomos y conectores son permitidas: *fórmulas bien formadas* (*wff*)

# Reglas de Derivación

- Modus ponens

$$A, A \rightarrow B \models B$$

- Modus tollens

$$\neg B, A \rightarrow B \models \neg A$$

# Lógica de predicados de primer orden

En lógica proposicional los átomos son los constituyentes de las fórmulas y son:  
*verdaderos o falsos*

Limitación: no puede expresar propiedades generales de casos similares.

Por ejemplo, “**todos los alumnos de Met. de I.A. se están durmiendo**”

## **Símbolos:**

- Símbolos de predicados (mayúsculas) asociados con su aridad (N) o número de argumentos (Si aridad = 0  $\Rightarrow$  proposiciones (átomos))
- Variables: minúsculas (x,y,z)
- Símbolos funcionales: minúsculas asociados con su número de argumentos (funciones con aridad = 0  $\Rightarrow$  constantes)

- Conectores lógicos
- Cuantificadores: **universal** (para toda  $x$ )  $\forall x$  y **existencial** (existe una  $x$ )  $\exists x$
- Símbolos auxiliares '(', ')', ',', '!'.

Un **término** es: una constante, variable o una función de términos

Una **fórmula atómica** o **átomo** es un predicado de  $N$  términos

Una fórmula bien formada (*wff*) en lógica de predicados es:

- un átomo
- si  $F$  es *wff* entonces  $\neg F$  también lo es
- Si  $F$  y  $G$  son *wff*,  $F \wedge G$ ,  $F \vee G$ ,  $F \longrightarrow G$ ,

$$F \leftrightarrow G$$

son *wff*

- Si  $F$  es *wff* y  $x$  es una variable libre en  $F$ , entonces  $\forall x F$  y  $\exists x F$  son *wff* (la variable  $x$  se dice *acotada* o “*bounded*”)
- ninguna otra fórmula es *wff*

Ejemplo:

$$\forall x F(x) \rightarrow G(x)$$

# Reglas de Producción

- Introducción
- Partes principales
- Tipos de razonamiento
- Explicación
- Resolución de conflictos
- Apareamiento de patrones

## Estructura de una regla:

$$P_1 \dots, P_m \longrightarrow Q_1 \dots, Q_n$$

Que significa:

IF las condiciones  $P_1$  y  $P_2$  y... y  $P_m$  se cumplen THEN realiza las acciones (o concluye)  $Q_1$  y... y  $Q_n$

**IF Animal es\_un carnívoro AND**

**Animal color café AND**

**Animal tiene rayas**

**THEN Animal es tigre**

## Propiedades de las reglas:

- **Modular:** cada regla define un pequeño y relativamente independiente pedazo de conocimiento
- **Incremental:** nuevas reglas pueden ser añadidas a la base de conocimiento relativamente independiente de las demás

- **Modificable:** como consecuencia de ser modular, las reglas pueden ser modificadas o se pueden añadir/quitar reglas fácilmente
- **Transparencia:** habilidad de explicar sus decisiones y soluciones

# Partes principales

## Un sistema de producción tiene:

- Un conjunto de reglas (base de conocimiento)
- Un interprete de reglas o máquina de inferencia (que decide qué regla aplicar, controla la actividad del sistema)
- Una memoria de trabajo (que guarda los datos, metas, y resultados intermedios)

# Memoria de Trabajo

Guarda inferencias/aseveraciones temporalmente.

Es la estructura de datos que es observada y manipulada/cambiada por las reglas.

## **Memoria de Trabajo**

**Los datos de la memoria de trabajo son los que permiten cumplir las condiciones de las reglas y dispararlas (las reglas verifican la existencia de elementos en la memoria de trabajo para disparar).**

**Las acciones de las reglas: modifican, añaden o quitan elementos de la memoria de trabajo (o producen efectos secundarios).**

# Máquina de Inferencia

El interprete o máquina de inferencia realiza el ciclo *reconoce-actua*:

1. Aparea las condiciones (o acciones) de las reglas con los elementos de la memoria de trabajo

**2.**

Si existe más de una regla que puede disparar, escoge una (resolución de conflictos)

**3.**

Aplica la regla (ejecuta las acciones/conclusiones) que puede involucrar cambios la memoria de trabajo

# **Máquina de Inferencia**

**Generalmente el ciclo empieza con unos datos iniciales y para cuando no hay reglas aplicables (o por una regla).**

**El sentido/importancia/contribución de cada regla depende de su contribución dentro de todas las reglas para solucionar el problema.**

## **Resolución de Conflictos**

**En razonamiento hacia adelante pueden existir más de una regla que puede disparar.**

**En razonamiento hacia atrás pueden existir más de una regla que cumple con las metas.**

**Lo que se necesita es una estrategia de resolución de conflictos que guíe (decida cual de las posibles reglas disparar) y evite que el proceso sea exponencial.**

## Tipos de Control

Para controlar ésto se utilizan: **control global** (independiente del dominio) y **control local** (dependiente del dominio).

El control local puede ser por medio de meta-reglas (reglas que controlan reglas).

Puntos: **sensibilidad** (responder rápidamente a cambios en el medio) y **estabilidad** (mostrar cierta continuidad en la línea de razonamiento).

## Estrategias globales

Las estrategias globales más utilizadas son:

- No se permite disparar una regla más de una vez con los mismos datos (*refractorines*)
- Preferir reglas que utilizan datos más recientes. La idea es de seguir una línea de razonamiento (*recency*)

## Estrategias globales

- Preferir reglas que son más específicas, i.e., reglas que tienen una mayor cantidad de condiciones y por lo tanto son más difíciles de disparar (*specificity*)

# Representaciones estructuradas

- Introducción
- Redes semánticas
- Prototipos Marcos (*frames*)

# Redes Semánticas

Modelo de memoria humana para capturar la semántica de las palabras y lograr un uso del significado parecido a los humanos [Quillian'66].

Un tipo de red en la cual los nodos representan objetos, conceptos o situaciones y los arcos representan relaciones entre ellos.

Se llama red semántica porque se usaron originalmente para representar el sentido en expresiones de lenguaje natural.

Ejemplo: Definición en un Diccionario

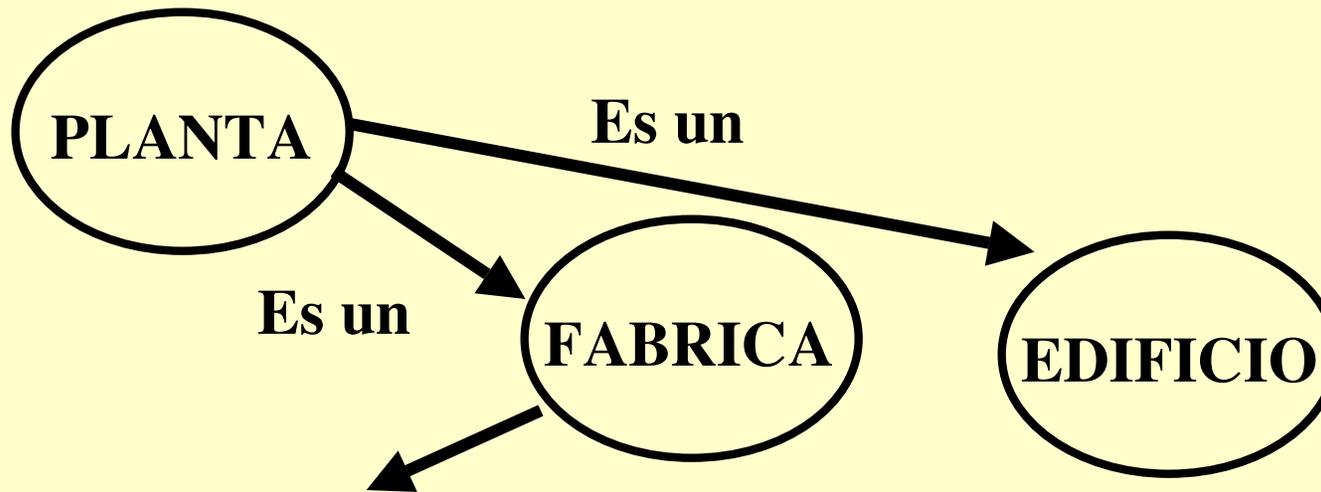
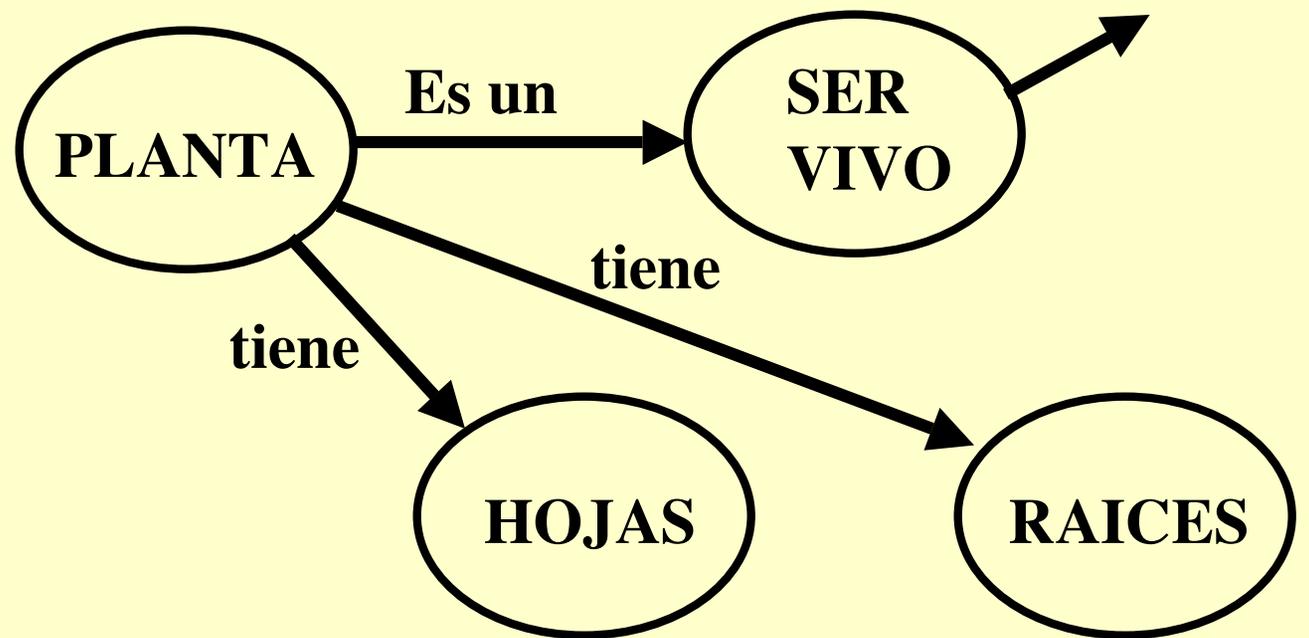
## **PLANTA**

(1) cosa viviente con hojas y raíces

(2) **fábrica** o edificio industrial

## **FÁBRICA**

lugar donde se fabrican bienes con  
máquinas



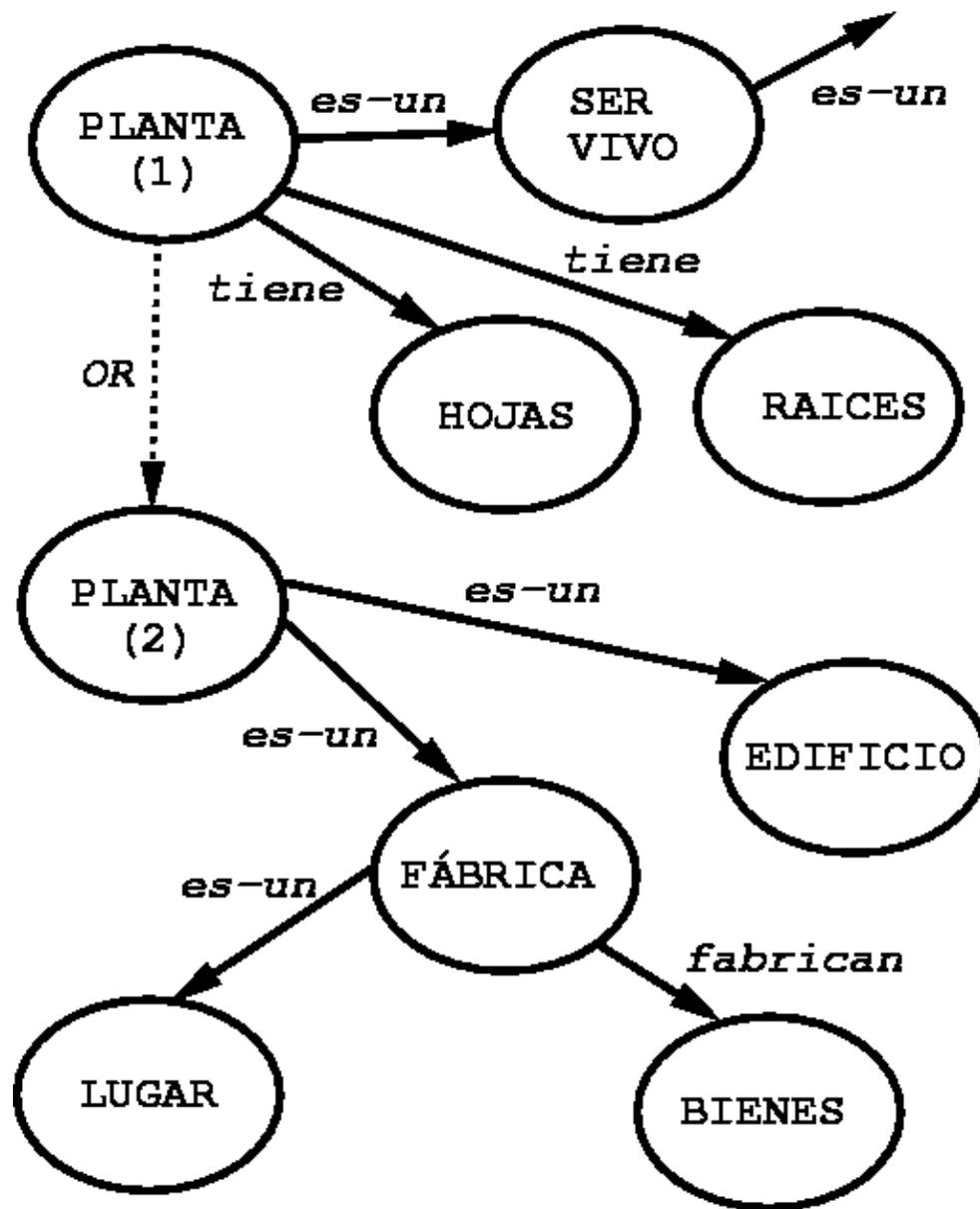
# Representación

- Los **nodos**: conceptos de palabras
- Los **arcos**: ligan conceptos para establecer la definición
- Cada palabra o nodo conceptual se consideraba la cabeza de un “plano” que tiene su definición (p. ej., si *banco* tiene 3 significados, entonces existen 3 planos para él).

Las ligas en el plano representan su definición.

Apuntadores fuera del plano hacen referencia a otros objetos (y planos) en donde se definen.

# RED SEMÁNTICA



# Tipos de Razonamiento

## 1. Búsqueda asociativa:

Encontrar si están relacionados dos o más conceptos, y su tipo de relación mediante el seguimiento de la red hasta encontrar las interacciones.

## 2. Reconocimiento:

Dada una serie de características (nodos “token”) encontrar el concepto (nodo clase) que mejor las define mediante su búsqueda y seguimiento en la red (reconocimiento de imágenes).

### 3. Descripción:

Expresar un concepto en base a sus componentes y relaciones entre ellas (lenguaje natural).

# FRAMES

**Frames:** estructuras de datos representando situaciones prototípicas (Minsky '75)

Una de las ideas intuitivas detrás de los Frames, es que la memoria se basa mucho en **estereotipos** (propiedades típicas de los objetos)

Los sistemas de *frames* razonan acerca de clases de objetos usando representaciones prototípicas, pero que pueden modificarse para capturar las complejidades del mundo real.

# Representación

**Idea:** tener una sola estructura de datos para poner el conocimiento relevante acerca de una clase de objetos, en lugar de tener el conocimiento distribuido en forma de reglas o fórmulas lógicas.

Permite construir conocimiento declarativo y procedural en un registro con *slots* y *fillers* o *facets*

Los *slots* son atributos y los *fillers* o *facets* son los valores, v.g.,

(frame (nombre camión)  
    (is-a objeto)  
    (color rojo)  
    (llantas 10)  
    ...)

Los slots pueden tener valores múltiples

# Jerarquías de *Frames*

Frames están puestos en una jerarquía en donde los frames de “abajo” pueden heredar los valores de los *slots* de los frames de “arriba”

Normalmente la herencia se hace por medio de los arcos: *is-a* (al final *instance-of*)

En general los frames de “arriba” tienen información típica (poco variable) mientras que los de “abajo” tienen información más específica. En ausencia de ésta, se utiliza la de los padres.

Se pueden hacer deducciones a través de la jerarquía (se distinguen entre los frames clases o genéricos y los frames instancias).

# Ejemplo de Jerarquía

Componentes

Tanque

Válvula

Sistema de enfriamiento

Con aceite

Control

Salida

Con aceite

Alta presión

Corriente directa

Con aceite

Con

**Jerarquía de los componentes de los sistemas de lubricación de una planta de energía.**

# Representaciones híbridas

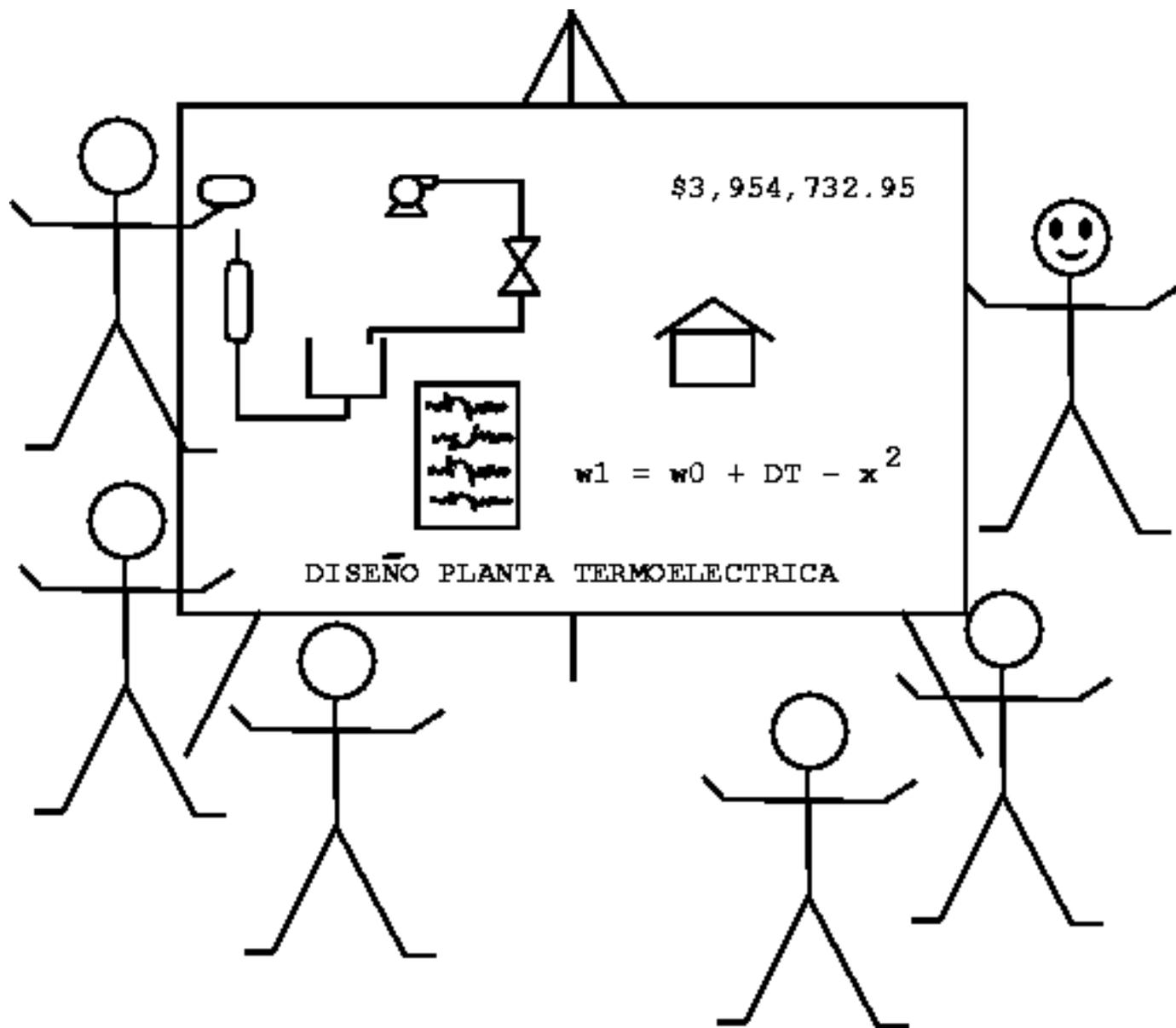
- Introducción
- Híbridos internos
- Híbridos externos
- Sistemas de pizarrón
- Sistemas de capas

## **Arquitectura de Pizarrón**

**Surgió a principios de los 70's para resolver problemas en donde existían varios tipos de “expertos” independientes.**

## **Metáfora:**

- **Una variedad de expertos cooperando a través de un pizarrón**
- **Cada experto interviene cuando ve que el estado del pizarrón es tal, que puede contribuir con algo**



# Componentes

## 1. Pizarrón: una estructura de datos multi-dimensional

- Juega el papel de memoria común de comunicación para las KS's
- Almacena toda la información relevante al problema

- **La información puede estar almacenada de diferentes maneras: jerarquías, niveles de abstracción, número de hipótesis, etc**
- **Puede tener información de control**

## **2. Fuentes de Conocimiento (KS's): se pueden ver como sistemas basados en conocimiento**

- **Juegan el papel de operadores que transforman progresivamente los estados de solución del problema**

- **Partes:**

- **Activación:** condiciones para utilizarse
- **Evaluación:** estimación (subjetiva) de resultados (recursos, tiempo, beneficios, # de hipótesis)
- **Acciones:** solución de problemas

## **Las fuentes de conocimiento pueden ser:**

- **Genéricas o específicas**
- **Únicas o redundantes**
- **Locales o distribuidas**
- **Homogéneas o híbridas**

## **Las acciones pueden ser:**

- **Algorítmicas**
- **Heurísticas**

### **3. Mecanismo de Control (*scheduler*): es el que lleva el razonamiento.**

**El algoritmo de ejecución “típico” es:**

- Ve las nuevas entradas al pizarrón**
- Ve cuáles KS's pueden hacer algo**
- Construye una agenda de registros de activación de fuentes de conocimiento**
- Ordena la agenda con un algoritmo**
- Evoca al KS ganador**

**El mecanismo de control funciona en base a un foco de atención el cual puede estar sobre:**

- **Las fuentes de conocimiento**
- **Los objetos del pizarrón**
- **Una combinación**

**En algunos sistemas de pizarrón, existen KS's cuyo trabajo es controlar la activación de otros KS's**

# Agentes que razonan bajo incertidumbre

- Introducción
- Repaso de probabilidad
- Sistemas basados en reglas
  - Factores de certeza (MYCIN)
- Sistemas expertos probabilistas

# Introducción

Los sistemas inteligentes deben ser capaces de representar y razonar con *incertidumbre*.

Existen varias causas de incertidumbre que tienen que ver con la información, el conocimiento y la representación.

# Información

- Incompleta
- Poco confiable
- Ruido, distorsión

# Conocimiento

- Impreciso
- Contradictorio

# Representación

- No adecuada
- Falta de poder descriptivo

## **Efectos de Incertidumbre**

**Si pierden varias propiedades de los sistemas que no tienen incertidumbre, basados en lógicas o reglas, lo cual hace el manejo de incertidumbre más complejo. Las principales dos características que, en general, ya no aplican son:**

- 1. Modularidad**
- 2. Monotonicidad**

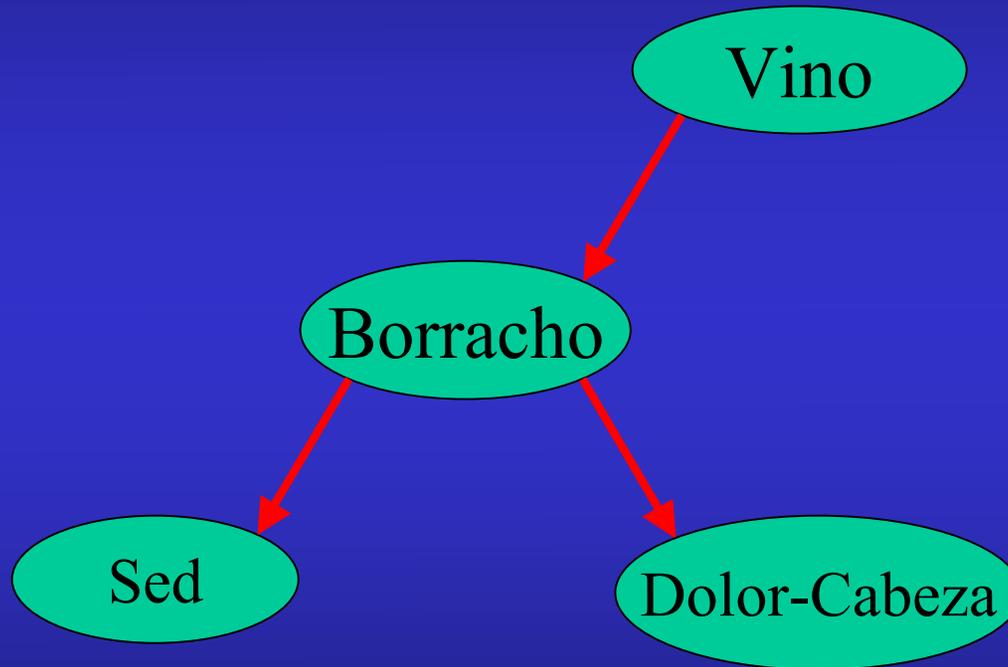
# Redes Bayesianas: Parte I

- Introducción
- Representación
- Inferencia
  - Propagación en árboles

# Representación

- Las redes bayesianas son una representación gráfica de dependencias para razonamiento probabilístico, en la cual los nodos y arcos representan:
  - Nodos: Variables proposicionales.
  - Arcos: Dependencia probabilística
- La variable a la que apunta el arco es dependiente (causa-efecto) de la que está en el origen de éste.

# Ejemplo de una red bayesiana



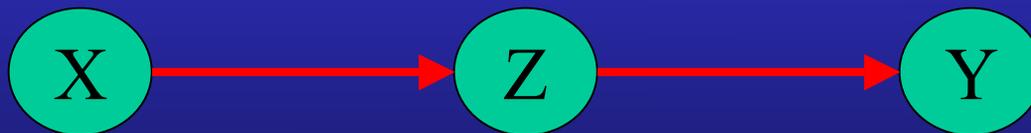
# Representación Gráfica

- Una red bayesiana representa en forma gráfica las dependencias e independencias entre variables aleatorias, en particular las independencias condicionales
- Independencia en la distribución
  - $P(X | Y, Z) = P(X | Z)$
- Independencia en el grafo
  - $X$  “separada” de  $Y$  por  $Z$

# Representación Gráfica

Notación:

- Independencia en la distribución
  - $I(X,Z,Y)$
- Independencia en el grafo
  - $\langle X | Z | Y \rangle$



# Separación “D”

- El conjunto de variables  $A$  es independiente del conjunto  $B$  dado el conjunto  $C$ , si no existe trayectoria entre  $A$  y  $B$  en que
  1. Todos los nodos convergentes están o tienen descendientes en  $C$
  2. Todos los demás nodos están fuera de  $C$

# Separación “D”

- Tres casos básicos
  - Arcos divergentes
  - Arcos en secuencia
  - Arcos convergentes

# Separación “D” – casos básicos

- caso 1: Secuencia:



- caso 2: Divergentes:



- caso 3: Convergentes:



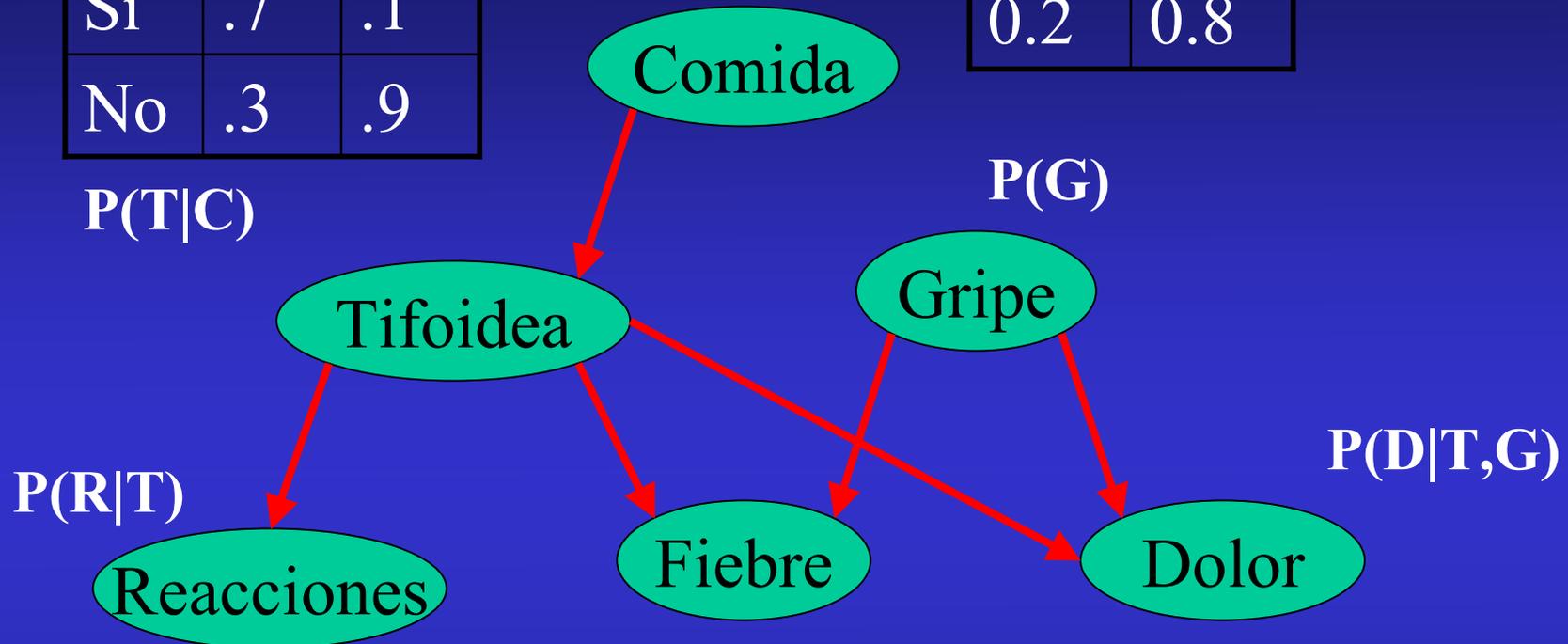
	Ins	Sal
Si	.7	.1
No	.3	.9

$P(T|C)$

Ins	Sal
0.2	0.8

$P(C)$

$P(G)$



$P(R|T)$

$P(D|T,G)$

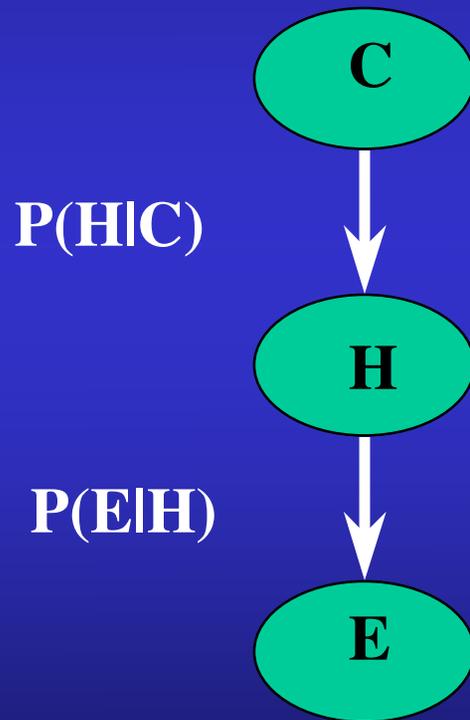
$P(F|T,G)$

	Si, Si	Si, No	No, Si	No, No
F	0.8	0.6	0.5	0.1
$\sim F$	0.2	0.4	0.5	0.9

# Inferencia probabilística

- En RB, la inferencia probabilística consiste en:  
“dadas ciertas variables conocidas (evidencia), calcular la probabilidad posterior de las demás variables (desconocidas)”
- Es decir, calcular:  $P(X_i | E)$ , donde:
  - $E$  es un subconjunto de variables de la RB (posiblemente vacío)
  - $X_i$  es cualquier variable en la RB, no en  $E$

# Inferencia bayesiana



*Causal:*

$$C \rightarrow H$$

*Evidencial:*

$$E \rightarrow H$$

*Mixta:*

$$C, E \rightarrow H$$

# Tipos de Técnicas

- Calcular probabilidades posteriores:
  - Una variable, cualquier estructura: algoritmo de eliminación (*variable elimination*)
  - Todas las variable, estructuras sencillamente conectadas (árboles, poliárboles): propagación
  - Todas las variables, cualquier estructura:
    - Agrupamiento (*junction tree*)
    - Simulación estocástica
    - Condicionamiento

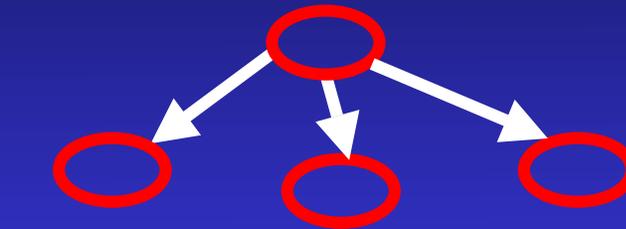
# Tipos de Técnicas

- Obtener variable(s) de mayor probabilidad dada cierta evidencia – abducción:
  - Abducción total
  - Abducción parcial

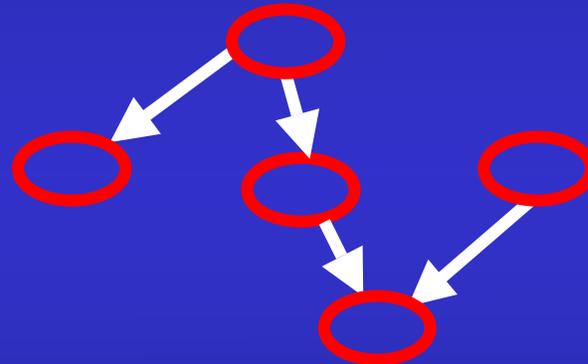
# Tipos de estructuras

- Sencillamente conectadas

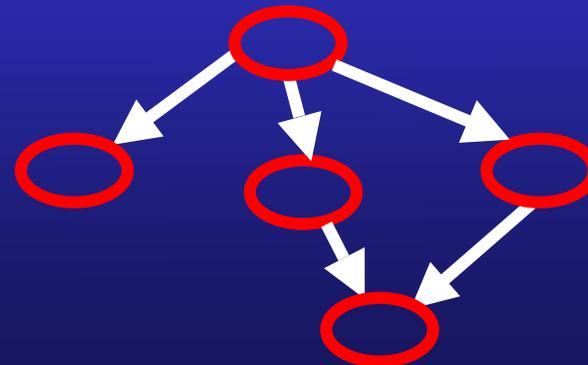
- Árboles



- Poliárboles



- Multiconectadas



# Algoritmo

- Mediante estas ecuaciones se integra un algoritmo de propagación de probabilidades en árboles.
- Cada nodo guarda los valores de los vectores  $\pi$  y  $\lambda$ , así como su matriz de probabilidad condicional (CPT),  $P$ .
- La propagación se hace por un mecanismo de paso de mensajes, en donde cada nodo envía los mensajes correspondientes a su padre e hijos

**Mensaje al padre (hacia arriba) –  
nodo B a su padre A:**

$$\lambda_B(A_i) = \sum_j P(B_j|A_i) \lambda(B_j)$$

**Mensaje a los hijos (hacia abajo) -  
nodo B a su hijo  $S_k$  :**

$$\pi_k(B_i) = \alpha \pi(B_j) \prod_{I \neq k} \lambda_I(B_j)$$

# Algoritmo

- Al instanciarse ciertos nodos, éstos envían mensajes a sus padres e hijos, y se propagan hasta a llegar a la raíz u hojas, o hasta encontrar un nodo instanciado.
- Así que la propagación se hace en un solo paso, en un tiempo proporcional al diámetro de la red.