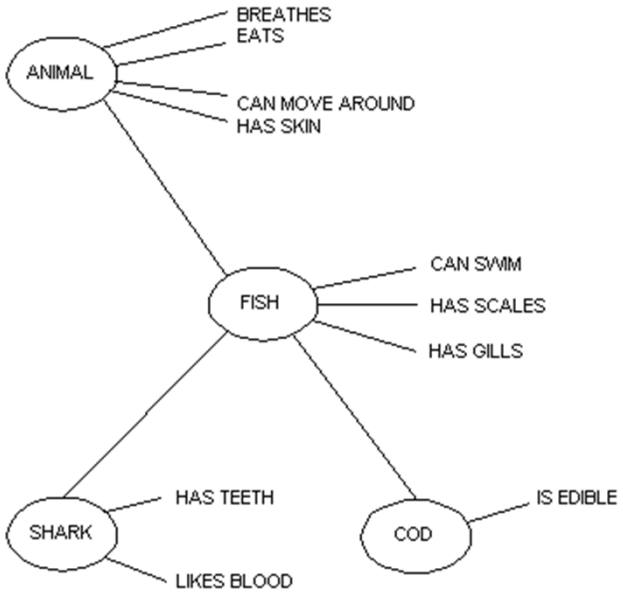
Tema 5. Comportamiento inteligente: Representación del Conocimiento e inferencia basados en lógica



Objetivos

- Entender que la resolución de problemas en IA implica definir una representación del problema y un proceso de búsqueda de la solución.
- Comprender la necesidad de representar el conocimiento y realizar inferencia para que un sistema pueda exhibir comportamiento inteligente.
- Conocer los fundamentos de la representación del conocimiento en lógica proposicional y de predicados y sus mecanismos de inferencia asociados.
- Aplicar los aspectos de representación basada en la lógica y mecanismos de inferencia, mediante técnicas y herramientas de programación lógica.

Estudia este tema en …

- Nils J. Nilsson, "*Inteligencia Artificial: Una nueva síntesis",* Ed. Mc Graw Hill, 2000. pp. 215-284

Contenido

- Representación del conocimiento en IA
- El cálculo proposicional
- Cálculo de predicados
- Introducción a los Sistemas Basados en el Conocimiento

Representación del conocimiento en IA

- Hemos estudiado varias formas de modelar el mundo de un agente, entre ellas:
  - **Representaciones icónicas:** Simulaciones del mundo que el agente podía percibir.
  - **Representaciones descriptivas:** Valores binarios que describían aspectos ciertos o falsos sobre el mundo.
- Las representaciones descriptivas tienen ciertas ventajas sobre las icónicas:
  - Son más sencillas.
  - Son fáciles de comunicar a otros agentes.
  - Se pueden descomponer en piezas más simples.

Representación del conocimiento en IA

- Además, hay información del entorno del agente que no se puede representar mediante modelos icónicos, tales como:
  - **Leyes generales**. "Todas las cajas azules pueden ser cogidas".
  - **Información negativa**. "El bloque A no está en el suelo", sin decir dónde está el bloque A.
  - **Información incierta.** "O bien el bloque A está sobre el bloque C, o bien el bloque A está sobre el bloque B".
- Sin embargo, este tipo de información es fácil de formular como conjunto de restricciones sobre los valores de las características binarias del agente.
- Estas restricciones representan **conocimiento sobre el mundo**.

Representación del conocimiento en IA

- A menudo, este **conocimiento sobre el mundo** puede utilizarse para razonar sobre él y hallar nuevas características del mismo.
  Ejemplo:
  - El conocimiento que se tiene es "Todos los pájaros vuelan"; y "Piolín es un pájaro".
  - Se puede ***razonar***, por tanto, que "Piolín vuela".
- **Otro Ejemplo:** Un robot sólo puede levantar un bloque si tiene suficiente batería y el bloque es elevable. Entonces, el conocimiento sobre el mundo es: "Si el bloque es elevable y hay suficiente batería, entonces es posible levantar el bloque".
- El robot "sabrá" si es capaz de levantar el bloque a partir de este **conocimiento** sobre su entorno.

Representación del conocimiento en IA

- Estudiaremos 2 tipos básicos para representar el conocimiento y razonar sobre él:
  - Cálculo proposicional.
  - Cálculo de predicados.

Cálculo Proposicional

- Elementos de representación: proposiciones y conectivas

$$\wedge(\text{y}), \vee(\text{o}), \rightarrow(\text{implica}), \neg(\text{no})$$

- Inferencia: deducciones con reglas, hechos y Modus-Ponens
- Ejemplos: llueve, (¬Nieva∧llueve)vHay-hielo
- Ventaja: representación de tipo general, y decidible (en tiempo finito es capaz de decidir si una proposición es deducible de la información disponible o no)
- Problema: si se quiere razonar sobre conjuntos de cosas. Por ejemplo, grafos, o jerarquías de conceptos.

Demostración

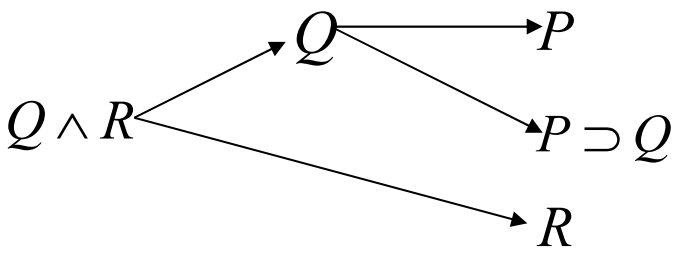
- Supongamos Δ un conjunto de FBFs, y una secuencia de **n** FBFs $\{\mathbf{w_1}, \mathbf{w_2}, \mathbf{w_3}, \ldots, \mathbf{w_n}\}$.
- Esta secuencia de FBFs se llama **demostración** o **deducción** de $\mathbf{w_n}$ a partir de Δ si, y sólo si, cada $\mathbf{w_i}$ de la secuencia pertenece a Δ o puede inferirse a partir de FBFs en Δ.
- Si existe tal demostración, entonces decimos que $\mathbf{w_n}$ es un **teorema de** Δ, y decimos que $\mathbf{w_n}$ puede demostrarse desde Δ con la siguiente notación: $\Delta \vdash \mathbf{w_n}$,
- o como $\Delta \vdash_{\mathbf{R}} \mathbf{w_n}$ para indicar que $\mathbf{w_n}$ se demuestra desde Δ mediante las reglas de inferencia **R**.

Demostración

- **Ejemplo:**
  - Sea el conjunto de FBFs Δ, $\Delta = \{P, R, P \supset Q\}$
  - Entonces, la siguiente secuencia es una demostración de la Fórmula Bien Formada $R \wedge Q$:

$$\{P, P \supset Q, Q, R, Q \wedge R\}$$

  - La demostración se puede llevar a cabo fácilmente a través del siguiente **árbol de demostración**, utilizando Δ y las reglas de inferencia:



Reglas de inferencia

- Reglas de equivalencia:
  - $\neg (p \wedge q) = (\neg p) \vee (\neg q)$
  - $\neg (p \vee q) = (\neg p) \wedge (\neg q)$
  - $\neg \neg p = p$
- Inferencia (o deducción):
  - p,q; por tanto $p \wedge q$
  - **Modus-Ponens** ($p \rightarrow q$ y p; por tanto q)

Ejemplo: proposiciones

- Una fabrica tiene cuatro sensores que detectan fuego y dos sensores que detectan fugas en el circuito del agua. Existen tres alarmas que se producen en diferentes ocasión

$$s_1, s_2, s_3, s_4, f_1, f_2, a_1, a_2, a_3$$

- Si el detector 3 de fuego o el detector 2 de fugas saltan, se debe producir la alarma 1

$$R1: s_3 \vee f_2 \rightarrow a_1$$

- Si saltan los detectores de fuego 1 y 4, se debe producir la alarma 2

$$R2: s_1 \wedge s_4 \rightarrow a_2$$

- Si salta la alarma 1, y el detector de fuego 2 o el de fugas 1, se debe producir la alarma 3

$$R3: a_1 \wedge (s_2 \vee f_1) \rightarrow a_3$$

Ejemplo: proposiciones

- Una fabrica tiene cuatro sensores que detectan fuego y dos sensores que detectan fugas en el circuito del agua. Existen tres alarmas que se producen en diferentes ocasión

$$s_1, s_2, s_3, s_4, f_1, f_2, a_1, a_2, a_3$$

- Si el detector 3 de fuego o el detector 2 de fugas saltan, se debe producir la alarma 1

$$R1: s_3 \vee f_2 \rightarrow a_1$$

- Si saltan los detectores de fuego 1 y 4, se debe producir la alarma 2

$$R2: s_1 \wedge s_4 \rightarrow a_2$$

- Si salta la alarma 1, y el detector de fuego 2 o el de fugas 1, se debe producir la alarma 3

$$R3: a_1 \wedge (s_2 \vee f_1) \rightarrow a_3$$

Ejemplo: proposiciones

- Una fabrica tiene cuatro sensores que detectan fuego y dos sensores que detectan fugas en el circuito del agua. Existen tres alarmas que se producen en diferentes ocasión

$$s_1, s_2, s_3, s_4, f_1, f_2, a_1, a_2, a_3$$

- Si el detector 3 de fuego o el detector 2 de fugas saltan, se debe producir la alarma 1

$$R1: s_3 \vee f_2 \rightarrow a_1$$

- Si saltan los detectores de fuego 1 y 4, se debe producir la alarma 2

$$R2: s_1 \wedge s_4 \rightarrow a_2$$

- Si salta la alarma 1, y el detector de fuego 2 o el de fugas 1, se debe producir la alarma 3

$$R3: a_1 \wedge (s_2 \vee f_1) \rightarrow a_3$$

Ejemplo: proposiciones

- Una fabrica tiene cuatro sensores que detectan fuego y dos sensores que detectan fugas en el circuito del agua. Existen tres alarmas que se producen en diferentes ocasión

$$s_1, s_2, s_3, s_4, f_1, f_2, a_1, a_2, a_3$$

- Si el detector 3 de fuego o el detector 2 de fugas saltan, se debe producir la alarma 1

$$R1: s_3 \vee f_2 \rightarrow a_1$$

- Si saltan los detectores de fuego 1 y 4, se debe producir la alarma 2

$$R2: s_1 \wedge s_4 \rightarrow a_2$$

- Si salta la alarma 1, y el detector de fuego 2 o el de fugas 1, se debe producir la alarma 3

$$R3: a_1 \wedge (s_2 \vee f_1) \rightarrow a_3$$

Ejemplo de Inferencia: deducción

- Han saltado el detector de fuego 2 y el de fugas 2.
- ¿Qué alarmas saltaran?
  a. s_2
  b. f_2
  c. (R1 y a) a_1
  d. (R3, a y c) a_3

Ejemplo de Inferencia: deducción

- Han saltado el detector de fuego 2 y el de fugas 2.
- ¿Qué alarmas saltaran?
  a. s_2
  b. f_2
  c. (R1 y a) a_1
  d. (R3, a y c) a_3

Ejemplo de Inferencia: deducción

- Han saltado el detector de fuego 2 y el de fugas 2.
- ¿Qué alarmas saltaran?
  a. s_2
  b. f_2
  c. (R1 y a) a_1
  d. (R3, a y c) a_3

Resolución en el cálculo proposicional

- **La refutación** es útil para demostrar que la negación de una cláusula es inconsistente en el sistema, quedando así demostrada, por tanto, la veracidad de dicha cláusula.
- **Ejemplo:**
  - 1. Convertir las FBFs de Δ como conjunciones de cláusulas

$a) \; BATERIA\_OK$

$b) \; \neg ROBOT\_SE\_MUEVE$

$c) \; \neg BATERIA\_OK \vee \neg OBJETO\_ELEVABLE \vee ROBOT\_SE\_MUEVE$

  - 2. Convertir **¬w** como conjunción de cláusulas

$$OBJETO\_ELEVABLE$$

Resolución en el cálculo proposicional

- **Ejemplo:**
  - 3. Unir el resultado de los pasos 1 y 2 en un único conjunto **Γ**

$$\Gamma = \{$$
$$a) BATERIA\_OK$$
$$b) \neg ROBOT\_SE\_MUEVE$$
$$c) \neg BATERIA\_OK \vee \neg OBJETO\_ELEVABLE \vee ROBOT\_SE\_MUEVE$$
$$d) OBJETO\_ELEVABLE$$
$$\}$$

Resolución en el cálculo proposicional

- **Ejemplo:**
  - 4. Aplicar la resolución a las cláusulas de **Γ** de forma iterativa, hasta que no haya nada más que resolver o se llegue a **Nil**
    - Resolviendo c) y d), tenemos que

$$e) \neg BATERIA\_OK \vee ROBOT\_SE\_MUEVE$$

    - Resolviendo e) y b), tenemos:

$$f) \neg BATERIA\_OK$$

    - Resolviendo f) y a), tenemos **Nil**.
    - Queda demostrado que ¬OBJETO_ELEVABLE

Resolución en el cálculo proposicional

- Como hemos visto, el procedimiento de refutación mediante resolución consiste en "*aplicar resoluciones hasta que se genere la cláusula vacía o no se puedan hacer más resoluciones*".
- La selección de cláusulas para su resolución, de forma manual, es sencilla.
- Existen distintas estrategias que permiten determinar la selección de las cláusulas a resolver para conseguir una mayor eficiencia.

Dificultades de representación

- Una empresa tiene 10 empleados. Los empleados pueden trabajar en tres tipos de puestos: director, jefe o administrativo

  Empleado1TrabajaDeDirector, Empleado2TrabajaDeJefe, . . .

- Si es director gana 60000 euros brutos al año, si es jefe 30000 y, si no, 20000. Además, si tiene mas de dos hijos, gana 10000 euros mas al año

  Empleado1Tiene1Hijo, Empleado2Tiene3Hijos, . . .

  Empleado1Tiene1Hijo ∧ Empleado1TrabajaDeDirector → Empleado1Gana60000

  Empleado2Tiene3Hijos ∧ Empleado2TrabajaDeJefe → Empleado2Gana40000

Cálculo de Predicados

- Elementos de representación:
  - Términos: Constantes (UGR), Variables (X), Funciones (siguiente(X))
  - Fórmulas atómicas: Predicados definidos sobre términos
    - trabaja-como(empleado1,director)
    - tiene-hijos(empleado1,1)
  - Fórmulas bien formadas (fbf): Fórmulas atómicas unidas por conectivas ($\wedge,\vee,\neg,\rightarrow$) y cuantificadas ($\forall,\exists$)
    - $\forall$ X,Y trabaja-como(X,director), tiene-hijos(X,Y), Y<=2 $\rightarrow$ gana(X,60000)
    - $\forall$ X,Y trabaja-como(X,director), tiene-hijos(,;Y), Y>2 $\rightarrow$ gana(X,70000)

Reglas de inferencia

- Inferencia: Todas las de lógica proposicional + instanciación universal
- Instanciación universal: si tenemos ∀ X p(X) entonces se puede deducir p(a), p(Y) . . .
- Ejemplo: Todos los hombres son mortales, Sócrates es un hombre, por tanto Sócrates es mortal:
  a. R1: ∀ X hombre(X) → mortal(X)
  b. hombre(sócrates)
  c. R1 y X=sócrates: hombre(sócrates) → mortal(sócrates)
  d. (b y c) mortal(sócrates)

Representación

- Una universidad imparte un conjunto de titulaciones en un conjunto de centros y campus.
  - imparte(Universidad,Titulación,Centro,Campus)
  - imparte(UGR,Informática,ETSIIT,Aynadamar)
  - Imparte(UGR,Matemáticas,FC,Ciencias)
- La representación no es única
  - imparte-titulación(Universidad,Titulación)
  - imparte-titulación(UGR,Informática)
  - imparte-titulación(UGR,Matemáticas)
  - imparte-centro(Informática,ETSIIT)
  - imparte-centro(Matémáticas,FC)
  - centro-en-campus(ETSIIT,Aynadamar)
  - centro-en-campus(FC,Ciencias)

Representación

- Una universidad imparte un conjunto de titulaciones en un conjunto de centros y campus.
  - imparte(Universidad,Titulación,Centro,Campus)
  - imparte(UGR,Informática,ETSIIT,Aynadamar)
  - Imparte(UGR,Matemáticas,FC,Ciencias)
- La representación no es única
  - imparte-titulación(Universidad,Titulación)
  - imparte-titulación(UGR,Informática)
  - imparte-titulación(UGR,Matemáticas)
  - imparte-centro(Informática,ETSIIT)
  - imparte-centro(Matémáticas,FC)
  - centro-en-campus(ETSIIT,Aynadamar)
  - centro-en-campus(FC,Ciencias)

Representación

- Una universidad imparte un conjunto de titulaciones en un conjunto de centros y campus.
  - imparte(Universidad,Titulación,Centro,Campus)
  - imparte(UGR,Informática,ETSIIT,Aynadamar)
  - Imparte(UGR,Matemáticas,FC,Ciencias)
- La representación no es única, alternativa:
  - imparte-titulación(Universidad,Titulación)
  - imparte-titulación(UGR,Informática)
  - imparte-titulación(UGR,Matemáticas)
  - imparte-centro(Informática,ETSIIT)
  - imparte-centro(Matémáticas,FC)
  - centro-en-campus(ETSIIT,Aynadamar)
  - centro-en-campus(FC,Ciencias)

Sigue el ejemplo

- Cada titulación tiene un plan de estudios formado por un conjunto de asignaturas troncales, obligatorias, optativas y de libre elección.
  - asignatura-en-plan(Asignatura,Titulación)
  - asignatura-en-plan(IA,Informática)
  - tipo-asignatura(Asignatura,Tipo)
  - tipo-asignatura(IA,obligatoria) o
  - tipo-asignatura(IA,troncal)
- Cada asignatura se imparte en un curso y cuatrimestre determinados y tiene un determinado numero de créditos.
  - curso-asignatura(Asignatura,Curso)
  - cuatrimestre-asignatura(Asignatura,Cuatrimestre)
  - créditos-asignatura(Asignatura,Créditos)

o

  - asignatura(Asignatura,Curso,Cuatrimestre,Creditos)

Inferencia

- Cuando un alumno se matricula por primera vez en primero, debe matricularse de todas las asignaturas del primer curso.

R1: ∀ X,U,Y primera-matrícula(X,U), curso-asignatura(Y,1) → matriculado-en(X,Y)

Inferencia: Deducción con Modus-Ponens

- En primero del grado en Informática de la UGR se imparte las asignaturas de FundamentosdePrograмación, . . .
  1. curso-asignatura(FundamentosdePrograмación,1)
  2. curso-asignatura(FundamentosdelSoftware,1)

- Ana Morales Pérez acaba de matricularse en primero de la titulación.
  a) primera-matricula(anaMorales,UGR)

- Si X=anaMorales, U=UGR, e Y =FundamentosdePrograмación, (unicación) por el Modus-Ponens, a partir de la regla R1, de 1 y de a), se puede deducir que

  matriculado-en(anaMorales, FundamentosdePrograмación)
- Si X=anaMorales, U=UGR, y Y =FundamentosdelSoftware, por el Modus-Ponens, a partir de la regla R1, de 2 y de a), se puede deducir que

  matriculado-en(anaMorales, FundamentosdelSoftware)

Deducción hacia atrás

- Y si se desea conocer ¿en qué asignaturas se debe matricular Ana Morales?
- Pregunta: matriculado-en(anaMorales,Y)
- Se busca una implicación lógica en la que aparezca matriculado-en(V, V1) en la parte derecha (puede haber más de una).
- Si se pueden unificar, se intentan deducir los literales que aparezcan en la parte izquierda de la implicación.
- Si existe alguna asignación de valor a las variables de la parte izquierda que permita deducir como ciertas las condiciones, se podrá deducir la pregunta de diferentes formas (diferentes valores de Y: FundamentosdePrograma ción, FudamentosdelSoftware, . . .)

Ejemplo

- El gran problema en IA no es cómo representar, sino **qué** representar. Requiere de una gran habilidad del diseñador. Ejemplos:
  - *"Todos los paquetes que están en la habitación 27 son más pequeños que los de la 28"*

    $$(\forall x,y)[[\mathbf{Paquete(x)} \wedge \mathbf{Paquete(y)} \wedge \mathbf{EnHabitacion(x,H27)} \wedge \mathbf{EnHabitacion(y,H28)}] \supset \mathbf{MasPequeño\,(x,y)}]$$

  - *"Cada paquete de la habitación 27 es más pequeño que uno de los paquetes de la habitación 29"*
    - Esta frase es ambigua (cosa frecuente en el lenguaje natural), ya que puede representar dos situaciones.

Ejemplo

- El gran problema en IA no es cómo representar, sino **qué** representar. Requiere de una gran habilidad del diseñador. Ejemplos:
  - *"Cada paquete de la habitación 27 es más pequeño que uno de los paquetes de la habitación 29"*
    - Situación 1:

$$(\exists y)(\forall x)[\mathbf{Paquete(x)} \wedge \mathbf{Paquete(y)} \wedge \mathbf{EnHabitacion(x, H27)} \wedge \mathbf{EnHabitacion(y, H29)}] \supset \mathbf{MasPequeño\,(x, y)}]$$

    - Situación 2:

$$(\forall x)(\exists y)[\mathbf{Paquete(x)} \wedge \mathbf{Paquete(y)} \wedge \mathbf{EnHabitacion(x, H27)} \wedge \mathbf{EnHabitacion(y, H29)}] \supset \mathbf{MasPequeño\,(x, y)}]$$

Ejemplo

- El cálculo de predicados nos permite también conceptualizar la noción del tiempo. Por ejemplo, si en una empresa de mensajería se desea decir que "*El paquete A ha llegado antes que el paquete B*", con cálculo de predicados lo haríamos de la siguiente forma, introduciendo una nueva variable que modele el tiempo:

$$(\exists z1, z2)[\mathbf{HaLlegado(A, z1)} \wedge \mathbf{H}aLlegado\mathbf{(B, z2)} \wedge \mathbf{A}ntes\mathbf{(z1, z2)}]$$

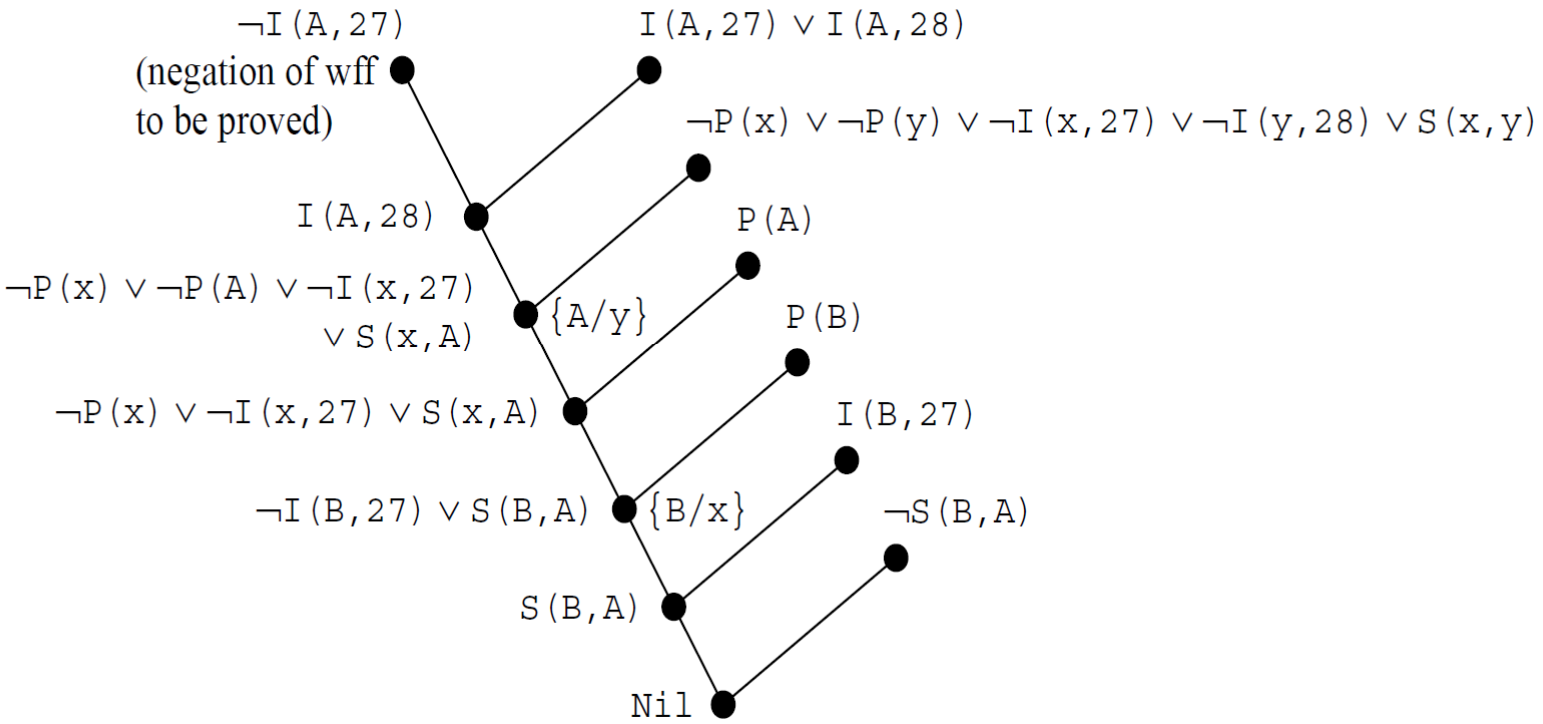
Resolución para demostrar teoremas

- **Se hace de forma análoga al procedimiento seguido en el cálculo proposicional.**
- **Ejemplo:** La Base de Conocimiento de un agente contiene los siguientes elementos:

$$(\forall x, y)[\{Paquete(x) \wedge Paquete(y) \wedge$$
$$\wedge EnHabitacion(x, H27) \wedge EnHabitacion(y, H28)\} \supset MasPequeño(x, y)]$$
$$Paquete(A)$$
$$Paquete(B)$$
$$EnHabitacion(B, H27)$$
$$EnHabitacion(A,27) \vee EnHabitacion(A,28)$$
$$\neg MasPequeño(B, A)$$

- **Pregunta: Queremos demostrar que el paquete está en la habitación 27**

Resolución para demostrar teoremas



Características del cálculo de predicados

- Ventaja: representación de tipo general mas rica que la proposicional
- Características de un sistema de razonamiento lógico:
  - solidez: para estar seguro que una conclusión inferida es cierta.
  - completitud: para estar seguros de que una inferencia tarde o temprano producirá una conclusión verdadera.
  - decidibilidad: para estar seguros de que la inferencia es factible.

- La refutación mediante resolución es sólida y completa.
- Problema: el cálculo de predicados es semidecidible y además en los casos en que la refutación mediante resolución termina, el procedimiento es NP-duro.

Características del cálculo de predicados

- Solución: subconjuntos decidibles de lógica de predicados (clausulas de Horn)
- Existe un lenguaje de programación que permite crear y ejecutar programas en lógica de predicados: PROLOG

```
conectados(X,Y) :- conectados(Y,X).
alcanzable(X,Y) :- conectados(X,Y).
alcanzable(X,Y) :- conectados(X,Z), alcanzable(Z,Y).
```

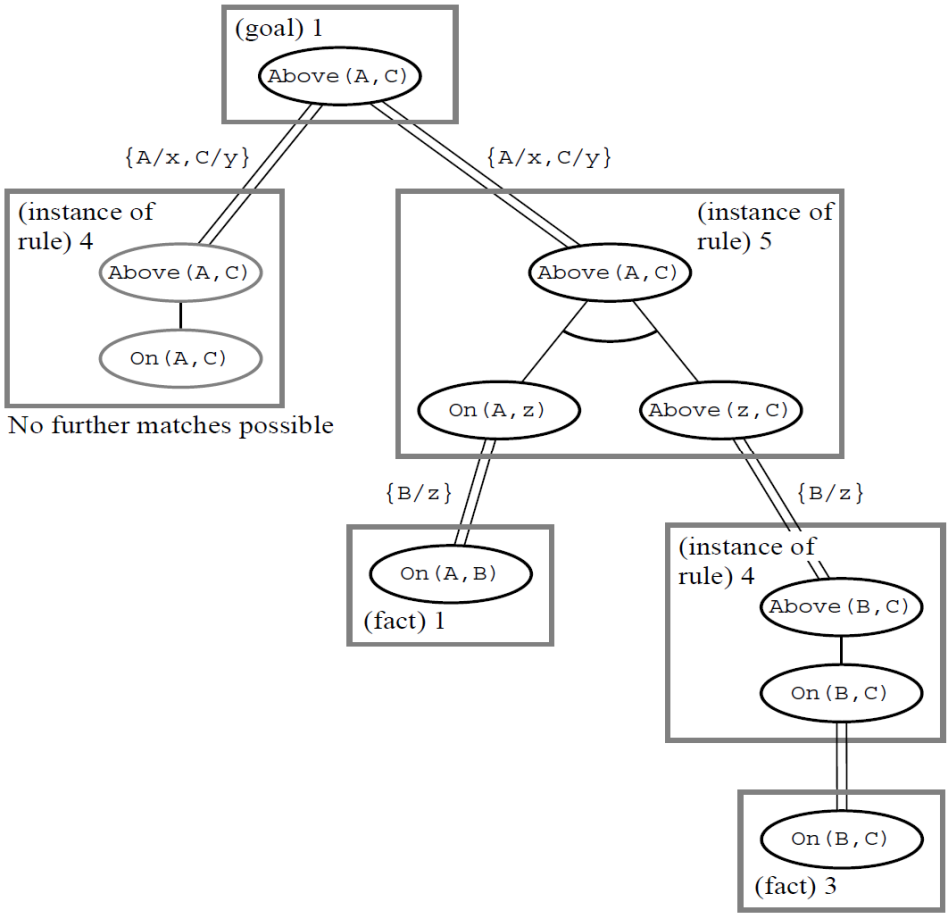
Ejemplo de PROLOG

$$(\forall x, y, z)[Sobre(x, y) \supset Encima(x, y)]$$

$$(\forall x, y)\{(\exists z)[Sobre(x, z) \wedge Encima(z, y)] \supset Encima(x, y$$

1. :- Encima(A,C)
2. Sobre(A,B) :-
3. Sobre(B.C) :-
4. Encima(x,y) :- Sobre(x,y)
5. Encima(x,y) :- Sobre(x,z), Encima(z,y)

Ejemplo de Prolog



Otras lógicas

- Lógicas de segundo orden (o de orden superior)
  - tienen dos (o tres) tipos definidos: los objetos y los conjuntos o funciones sobre los mismos (o ambos)
  - es equivalente a decir que los predicados pueden tomar otros predicados como argumentos
- Lógicas modales y temporales
  - necesario y posible
- Lógica difusa
  - grados de pertenencia
- Otras: multi-valuadas, no-monótonas, cuánticas, . . .

Lógica difusa

- Extensión de la lógica clásica diseñada para permitir el razonamiento sobre conceptos imprecisos
  - ”la velocidad del motor es muy alta”
  - ”el paciente tiene fiebre moderada”
  - ”si el paciente tiene fiebre muy alta y es muy joven, entonces la dosis debe de ser moderada”
- Ejemplo: control del movimiento de un robot

Sistemas Basados en el Conocimiento

- Una gran cantidad de aplicaciones reales de la IA se basan en la existencia de una gran masa de conocimiento:
  - Diagnóstico médico.
  - Diseño de equipos.
  - Sistemas de Recomendación.
  - Etc.
- Este tipo de sistemas se denominan **Sistemas Basados en el Conocimiento**, ya que este ocupa la parte central de la solución al problema a resolver.

Sistemas Basados en el Conocimiento

- Un **Sistema Basado en el Conocimiento (SBC)** necesita 3 componentes básicas:
  - Una **Base de Conocimiento (BC)**, que contenga el conocimiento experto necesario sobre el problema a resolver. Puede ser:
    - **Estática**, si la BC no varía a lo largo del tiempo.
    - **Dinámica,** cuando se añaden nuevos hechos o reglas, o se modifican las existentes a lo largo del tiempo.
  - Un **Motor de Inferencia**, que permite razonar sobre el conocimiento de la BC y los datos proporcionados por un usuario.
  - Una **interfaz de usuario** para entrada/salida de datos.

Sistemas Basados en el Conocimiento

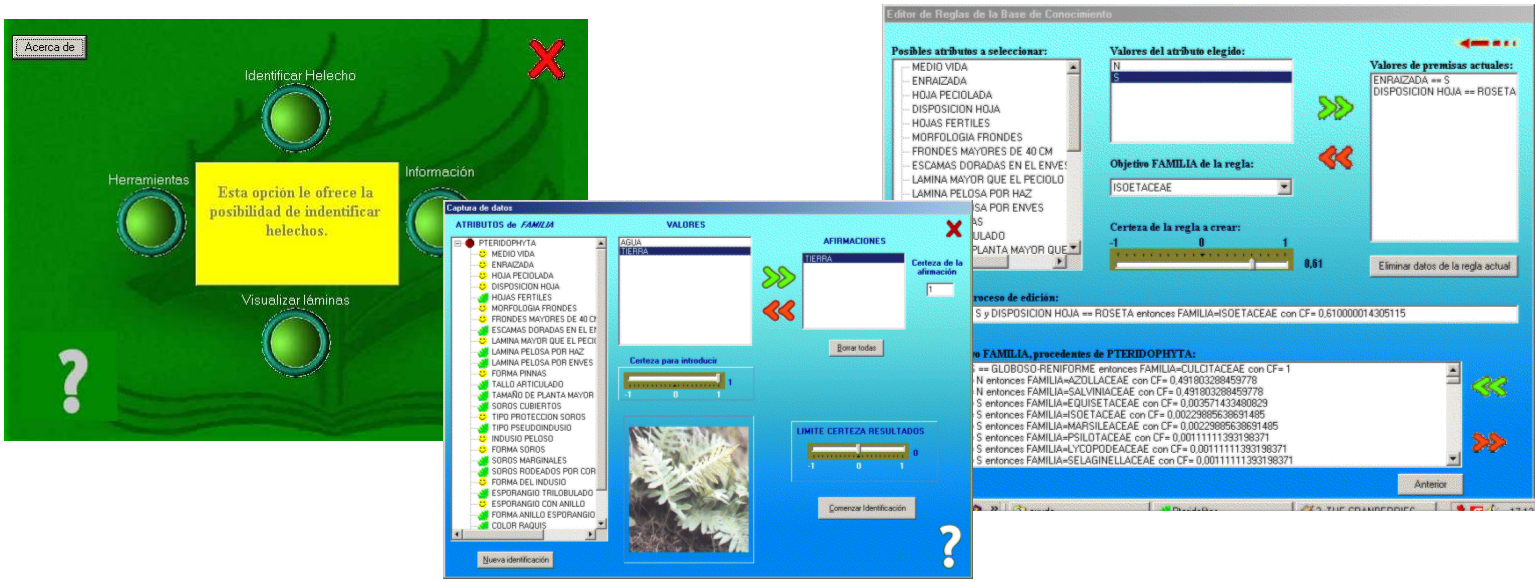
- Ejemplos de **SBC** que incorporan conocimiento específico sobre algún tema:
  - **Anna** (Agente conversacional de IKEA). Su Base de Conocimiento está descrita en lenguaje AIML (*Artificial Intelligence Markup Language*).

Sistemas Basados en el Conocimiento

- Ejemplos de **SBC** que incorporan conocimiento específico sobre algún tema:
  - El **Paciente Simulado Virtual (PSV)** es un agente que simula enfermedades. Se utiliza para la formación de especialistas en medicina: El médico pregunta y/o visualiza síntomas que el paciente presenta y debe decidir un diagnóstico válido.

Sistemas Basados en el Conocimiento

- Ejemplos de **SBC** que incorporan conocimiento específico sobre algún tema:
  - **Pteridophita** es un experto en helechos que, de forma interactiva con un usuario, permite identificar tipos de helechos y proporcionar información sobre los mismos.



Sistemas Expertos basados en Reglas (SEBR)

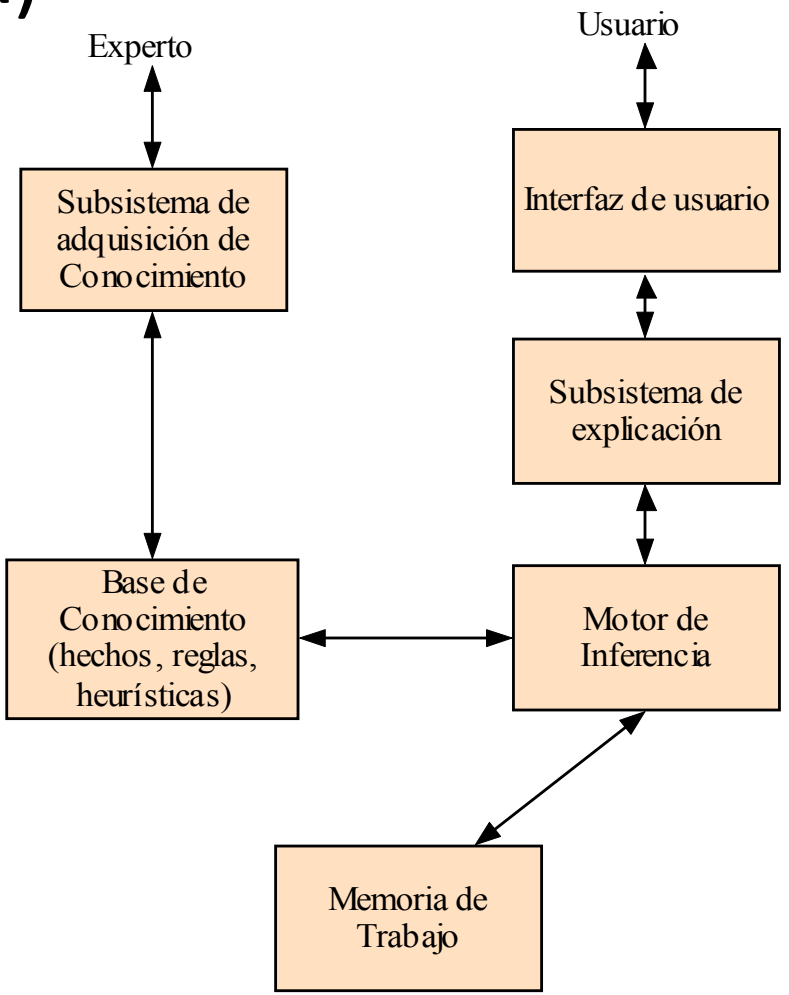
- Un **SEBR** es un **SBC** donde el conocimiento se incluye en forma de reglas y hechos.
- Estas reglas y hechos pueden implementarse, por ejemplo, mediante el **cálculo de predicados**.
- El proceso de construcción de un SEBR es el siguiente:
  - Se **extrae el conocimiento** experto (bibliografía, entrevistas a expertos reales, etc.).
  - Se **modela y se adquiere el conocimiento**, utilizando un lenguaje adecuado (cálculo de predicados, otras lógicas más avanzadas, etc.)
  - Se **crea la Base de Conocimiento** con el conocimiento adquirido.

Sistemas Expertos basados en Reglas (SEBR)

- Por otra parte, también se necesita:
  - Una **interfaz de usuario**, para poder utilizar el sistema y adquirir/enviar datos.
  - Un **subsistema de explicación**, para los casos en los que sea necesario indicar al usuario porqué se llega a las conclusiones que se llegan.
  - Un **Motor de Inferencia**, para razonar sobre la Base de Conocimiento y los datos proporcionados por el usuario.

Sistemas Expertos basados en Reglas (SEBR)

- El esquema general de diseño de un SEBR es el siguiente:
  - La **memoria de trabajo** contiene la información relevante que el Motor de Inferencia está usando para razonar las respuestas para el usuario.



Extracción del conocimiento en SEBR

- La forma más sencilla de representar los datos de un experto para la posterior extracción de conocimiento es mediante Bases de Datos relacionales y tablas relacionales. Ejemplo (Sistema Pteridophita). El objetivo es conocer la familia del helecho que se tiene

FAMILIA	MEDIO VIDA	DISPOSICIÓN HOJA	FRONDES > 40 CM	LAMINA > PECIOLO
ADIANTACEAE	TIERRA	INDIVIDUAL	S	S
PTERIDACEAE	TIERRA	INDIVIDUAL	S	S
ISOETACEAE	TIERRA	ROSETA	N	S
ADIANTACEAE	TIERRA	INDIVIDUAL	N	S/N
ADIANTACEAE	TIERRA	INDIVIDUAL	S	N
PTERIDACEAE	TIERRA	INDIVIDUAL	N	S

- Claramente, de la tabla se puede inferir visualmente que se extraerá la regla entre otras.

$$DisposicionHoja(ROSETA) \supset Familia(ISOETACEAE)$$

Extracción del conocimiento en SEBR

- Los métodos para extraer el conocimiento se estudian en el área de **aprendizaje automático** dentro de la IA. En SEBR, la extracción del conocimiento también se conoce como **extracción de reglas** o **aprendizaje de reglas.**
- Además, normalmente, la percepción del mundo por parte de un agente no es perfecta.
- Del mismo modo, es posible que una regla no sea aplicable siempre (aunque sí en un gran número de casos). Este hecho no permite que la regla sea admitida en un sistema de cálculo proposicional o de predicados, dado que daría lugar a sistemas inconsistentes.
- Se hace necesario establecer mecanismos para **tratar con incertidumbre**.

Otros modelos/problemas de representación del conocimiento

- Representación del conocimiento de sentido común
- Organización jerárquica del conocimiento
- Razonamiento temporal
- …

Organización jerárquica del conocimiento

- Organización jerárquica del conocimiento
  - Snoopy es una impresora láser
  - Todas las impresoras láser son impresoras
  - Todas las impresoras son máquinas

$$Impresora.laser(Snoopy)$$

$$(\forall x)[Impresora.laser(x) \supset Impresora(x)]$$

$$(\forall x)[Impresora(x) \supset Maquina.de.oficina(x)]$$

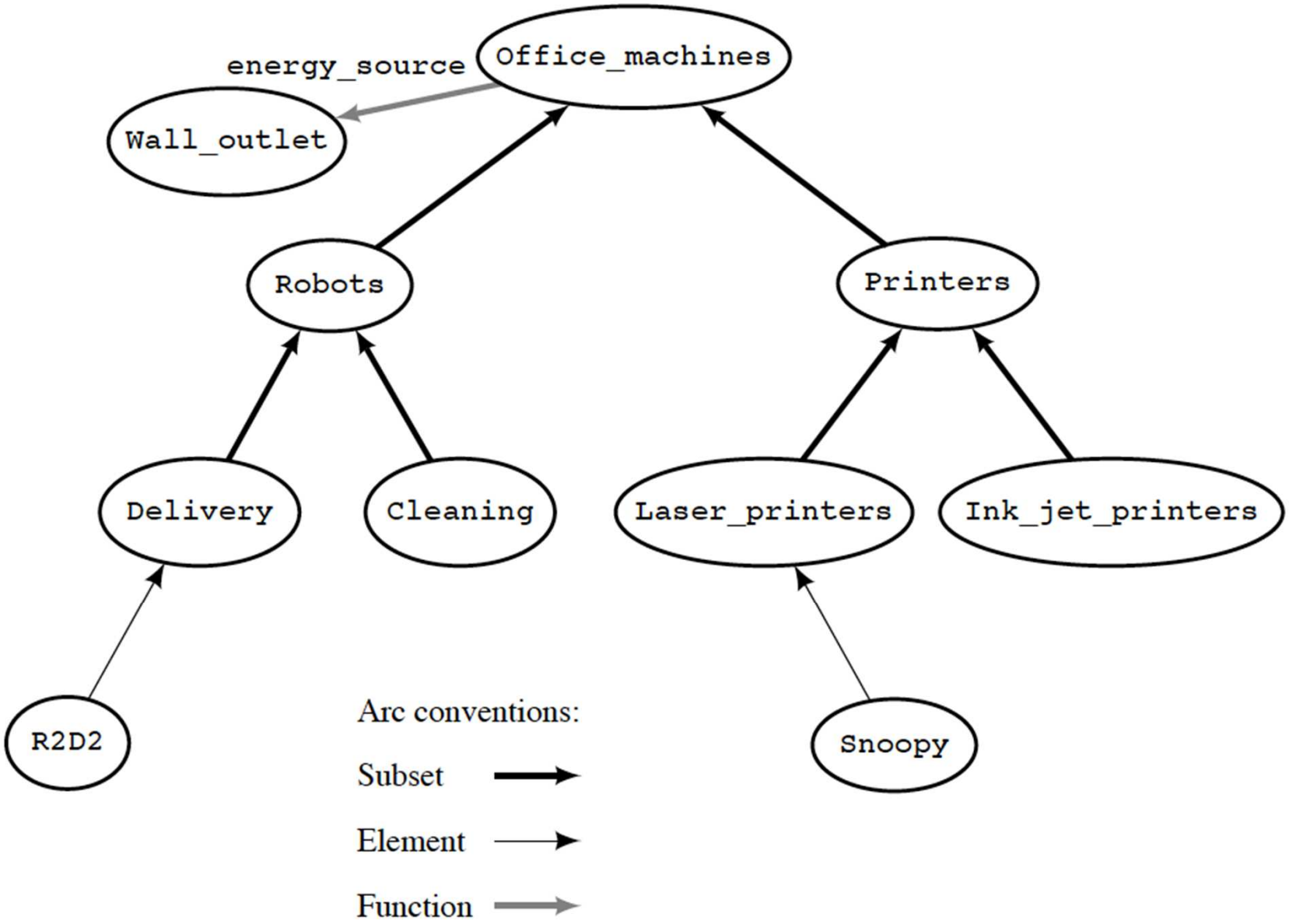
- Herencia de Propiedades

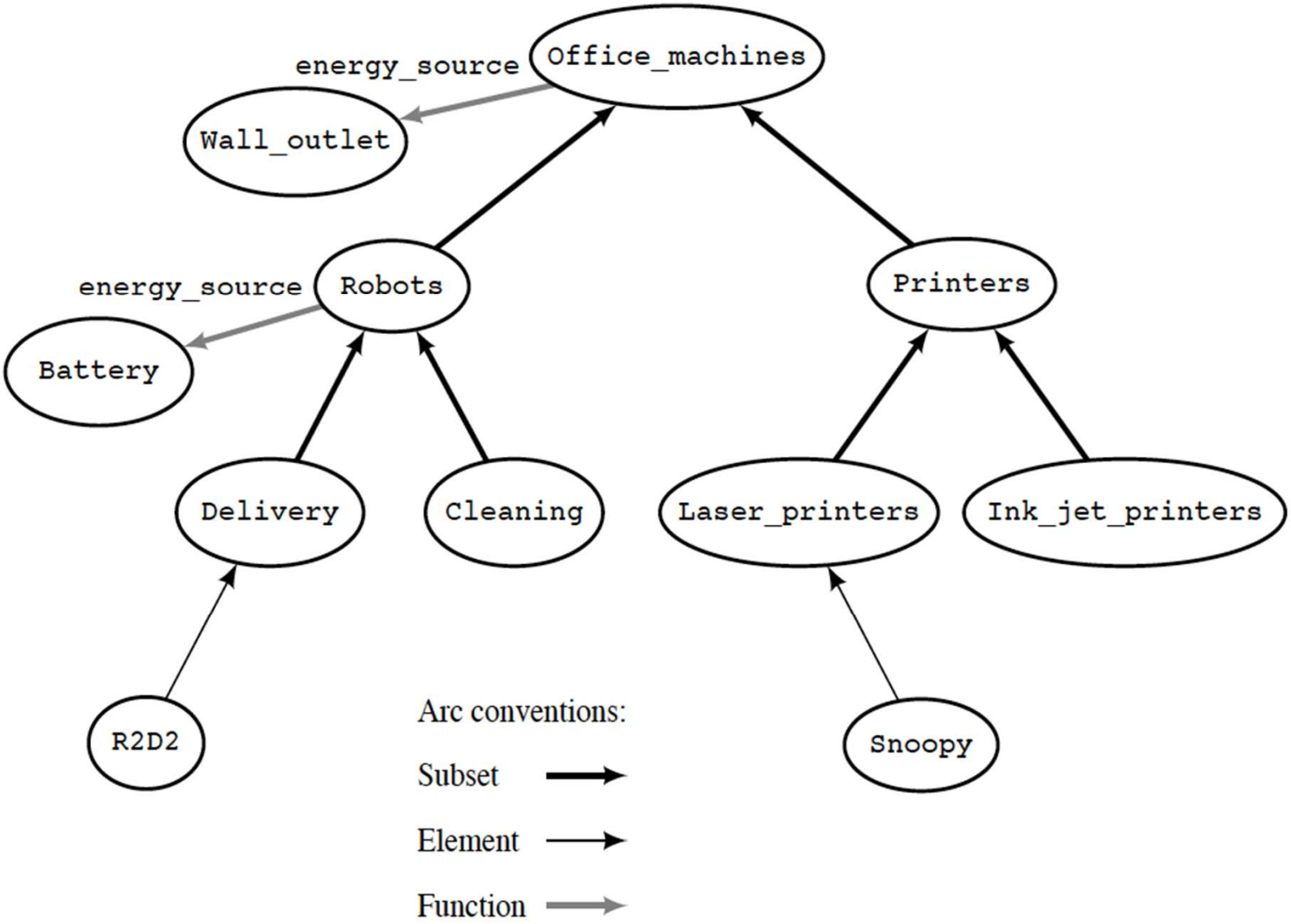
$$(\forall x)[Maquina.de.oficina(x) \supset [Fuente.de.alimentacion = Toma.de.la.pared]]$$

$$(\forall x)[Impresora.laser(x) \supset [Fuente.de.alimentacion = Toma.de.la.pared]]$$

Redes semánticas

- Las redes semánticas son estructuras gráficas que codifican el conocimiento taxonómico sobre objetos y propiedades de estos
- PROPIEDADES: nodos etiquetados con constantes de relación
- OBJETOS: nodos etiquetados con constantes de objetos
  - Arcos de jerárquica
  - Arcos de pertenencia
  - Arcos de función


energy_source
Office_machines
Wall_outlet
Robots
Printers
Delivery
Cleaning
Laser_printers
Ink_jet_printers
R2D2
Snoopy
Arc conventions:
Subset
Element
Function


energy_source
Office_machines
Wall_outlet
energy_source
Robots
Printers
Battery
Delivery
Cleaning
Laser_printers
Ink_jet_printers
R2D2
Snoopy
Arc conventions:
Subset
Element
Function

Razonamiento temporal

- Allen (1983,1984): El tiempo es algo dinámico, sobre el cual los procesos y los evento transcurren
  - E evento o suceso
  - I intervalo de tiempo

Ocurre(E,I)

- Intervalos temporales: instantes de inicio y final

$$(\forall x)[inicio(x) \leq fin(x)]$$

Razonamiento temporal

$$(\forall x, y)[Se.encuentra.con(x, y) \equiv (fin(x) = inicio(y)]$$

$$(\forall x, y)\{Antes.de(x, y) \equiv$$
$$\exists(z)[Se.encuentra.con(x, z) \wedge Se.encuentra.con(z, y)]\}$$

$$(\forall x, y)\{Antes.de(x, y) \equiv [(fin(x) < inicio(y)]\}$$

Razonamiento temporal

- Ejemplo: representación de hechos de sentido común el evento salir agua de un grifo está precedido por el de abrir una válvula, y seguido por el de cerrarla

$$(\forall y)\{Ocurre(Saleagua, y) \supset$$
$$(\exists x, z)[Ocurre(Abrir.V, x) \wedge Ocurre(Cerrar.V, z) \wedge$$
$$Se.solapa.con(x, y) \wedge Se.solapa.con(y, z)]\}$$