
Sistemas inteligentes

Representación del conocimiento

<http://www.gsi.dit.upm.es/~gfer/ssii/rcsi/>

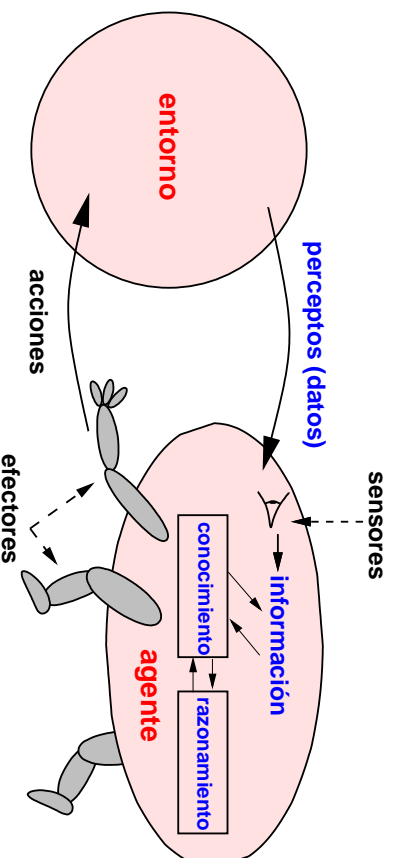
Representación del conocimiento

- El nivel del conocimiento y las ontologías
- Tipos de conocimiento
- La psicología como fuente de metamodelos
- Lenguajes formales de representación
- Lenguajes informales de representación
- Lógicas de descripciones
- Modelación del conocimiento incierto

Representación del conocimiento

- **El nivel del conocimiento y las ontologías**
 - Datos, información y conocimiento
 - La postura intencional
 - El nivel de conocimiento
 - Conceptuación
 - Ontologías
- Tipos de conocimiento
- La psicología como fuente de metamodelos
- Lenguajes formales de representación
- Lenguajes informales de representación
- Lógicas de descripciones
- Modelación del conocimiento incierto

Datos, información y conocimiento



- **Información:** Datos con *significado*
(datos interpretados por el agente)
- **Conocimiento:** Información *asimilada*

¿Qué es «conocimiento»?

R.A.E. (nueve acepciones):

«2. [m.] *Entendimiento, inteligencia, razón natural.*»

Epistemología, definición clásica:

«*Conocimiento es creencia verdadera justificada.*»

I.A., definición de Newell:

«*Lo que puede atribuirse a un agente de tal modo que su comportamiento pueda computarse de acuerdo con el principio de racionalidad.*»

Principio de racionalidad: ley de comportamiento en el nivel de conocimiento: si un agente sabe, o cree, que una de sus acciones le conduce a uno de sus objetivos seleccionará esa acción.

La postura intencional

¿El agente «sabe», «cree»...?

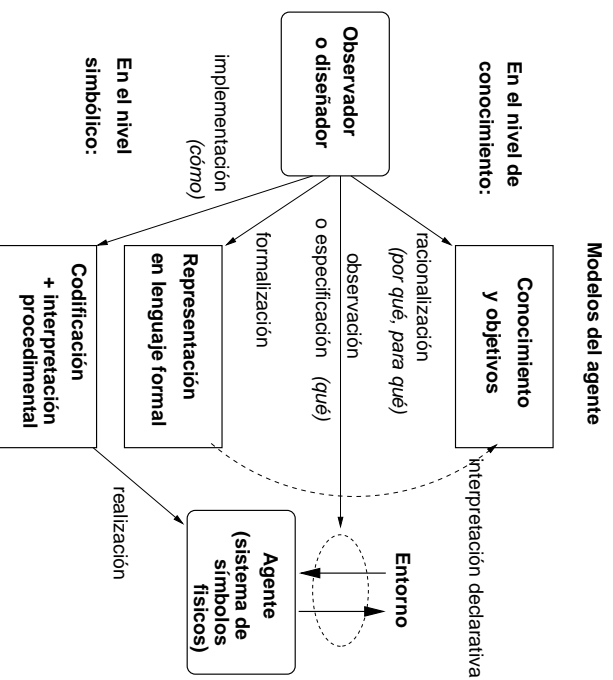
- Brentano (1874): objetos, fenómenos y estados mentales son intencionales: se dirigen a o representan algo, y a nada físico puede atribírsele esto. . .
- Dennet: «The Intentional Stance». MIT Press, 1987.
Sistema intencionado: «Su comportamiento puede predecirse mediante el método de atribuirle creencias, deseos y perspicacia racional»

¿Está justificado, o sólo es antropomorfismo retórico?

Neocibernética: atribución a los sistemas artificiales de actitudes intencionales como *herramienta de abstracción* ↔ «agentes BDI».

- actitudes epistémicas: conocimiento, sabiduría...
- actitudes doxásticas: creencia, duda...
- actitudes teleológicas (o conativas): deseo, intención...
- actitudes deónticas: obligación, compromiso...

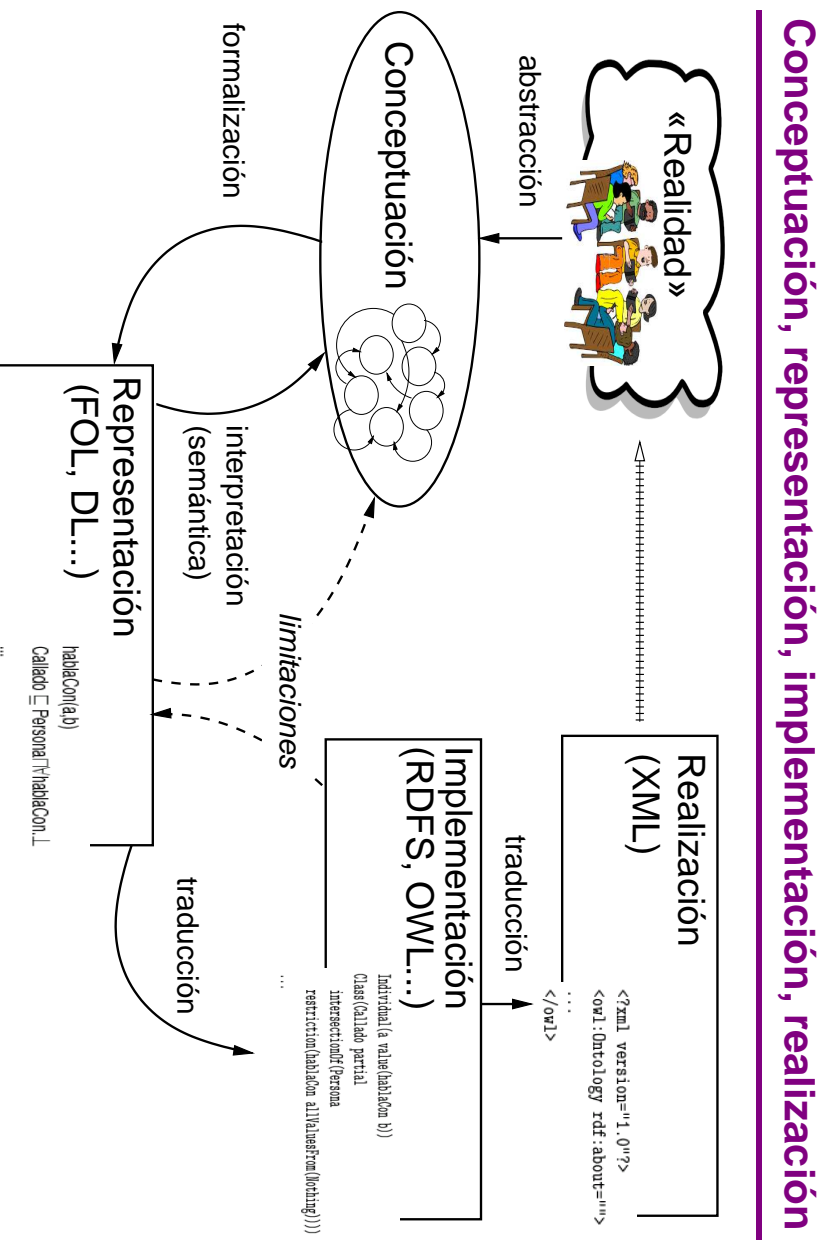
Modelación de agentes: el nivel de conocimiento



Concepción (nivel de conocimiento: Newell, 1981)

↪ **Representación** (nivel simbólico: Newell y Simon, 1976)

↪ **Codificación** (lenguajes de implementación y realización)



Conceptuación (1)

Modelo conceptual (o mental):

- Universo del discurso
- Propiedades, relaciones, restricciones, leyes...

«Formalmente, una conceptuación es una terna que consta de un universo del discurso, un conjunto de funciones de base, y un conjunto de relaciones de base en ese universo del discurso.»

Genesereth y Nilsson: Logical Foundations of Artificial Intelligence, 1987.

Ejemplo: $\{\langle a, b, c, d, e \rangle, \{\text{Encima-de}\}, \{\text{Libre, Bajo, Sobre, Más-arr-que}\}\}$

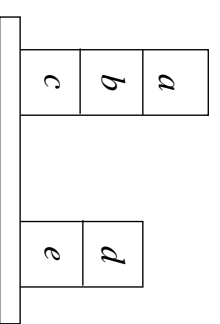
con: $\text{Encima-de} = \{b \mapsto a, c \mapsto b, e \mapsto d\}$

$\text{Libre} = \{a, d\}$

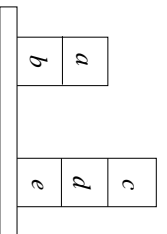
$\text{Bajo} = \{c, e\}$

$\text{Sobre} = \{\langle a, b \rangle, \langle b, c \rangle, \langle d, e \rangle\}$

$\text{Más-arr-que} = \{\langle a, b \rangle, \langle b, c \rangle, \langle a, c \rangle, \langle d, e \rangle\}$



Conceptuación (2)



¿Otra conceptuación?

La conceptuación debe reflejar un

conocimiento general sobre el dominio:

- El contenido semántico de la función *Encima-de* es el mismo que tiene la relación *Sobre*: $y = \text{Encima-de}(x)$ si y sólo si $\langle y, x \rangle \in \text{Sobre}$
- Si un bloque está libre no puede haber ninguno por encima de él, y viceversa: $x \in \text{Libre}$ si y sólo si no hay ningún y tal que $\langle y, x \rangle \in \text{Sobre}$
- Un bloque está más arriba que otro si está sobre él o está sobre un tercero que está sobre él: $\langle x, y \rangle \in \text{Más-arr-que}$ si y sólo si $\langle x, y \rangle \in \text{Sobre}$ o bien hay algún z tal que $\langle x, z \rangle \in \text{Sobre}$ y $\langle z, y \rangle \in \text{Más-arr-que}$
- Una relación básica, *Sobre* (define extensionalmente una situación)
- Definiciones de las demás *de manera intensional*

El nivel del conocimiento en SBC y en agentes

- Sistemas primitivos: conocimiento
 - ↔ representación en un lenguaje (nivel simbólico)
- Sistemas actuales: conocimiento
 - ↔ modelo (nivel de conocimiento)
 - ↔ representación en un lenguaje (nivel simbólico)
- En sistemas multiagente, necesidad de establecer lo que sabe y no sabe cada uno ↔ «ontologías»

Ontologías (sentido numerable)

«Una ontología es una especificación explícita de una concepción. Es un término adoptado de la filosofía, en la que una ontología es una explicación sistemática de la existencia. En la inteligencia artificial lo que existe es lo que puede representarse»
(Gruber, 1993)

«Una ontología es una especificación formal de una concepción compartida»
(Borst, 1997)

- En sistemas multiagente, **compromiso ontológico** (*commit*)
- Para minimizar «malentendidos» (*ontological mismatches*), lenguajes de implementación basados en lenguajes formales

Ontologías y sistemas expertos

- Sistemas expertos «de primera generación»:
 - Sistemas diseñados en el *nivel simbólico*
 - Adquisición del conocimiento = *codificación*
 - No atención a la ontología del dominio
- Sistemas expertos actuales:
 - Sistemas diseñados en el *nivel de conocimiento*
 - Adquisición del conocimiento = *modelación*
 - Ontología importante
 - Metodologías: KADS, etc.

Representación del conocimiento

- El nivel del conocimiento y las ontologías
 - **Tipos de conocimiento**
 - Declarativo y procedimental
 - Factual, normativo, táctico y estratégico
 - La psicología como fuente de metamodelos
 - Lenguajes formales de representación
 - Lenguajes informales de representación
 - Lógicas de descripciones
 - Modelación del conocimiento incierto

Conocimiento para la resolución de problemas

El conocimiento que se aplica para resolver un determinado tipo de problemas puede expresarse de dos formas:

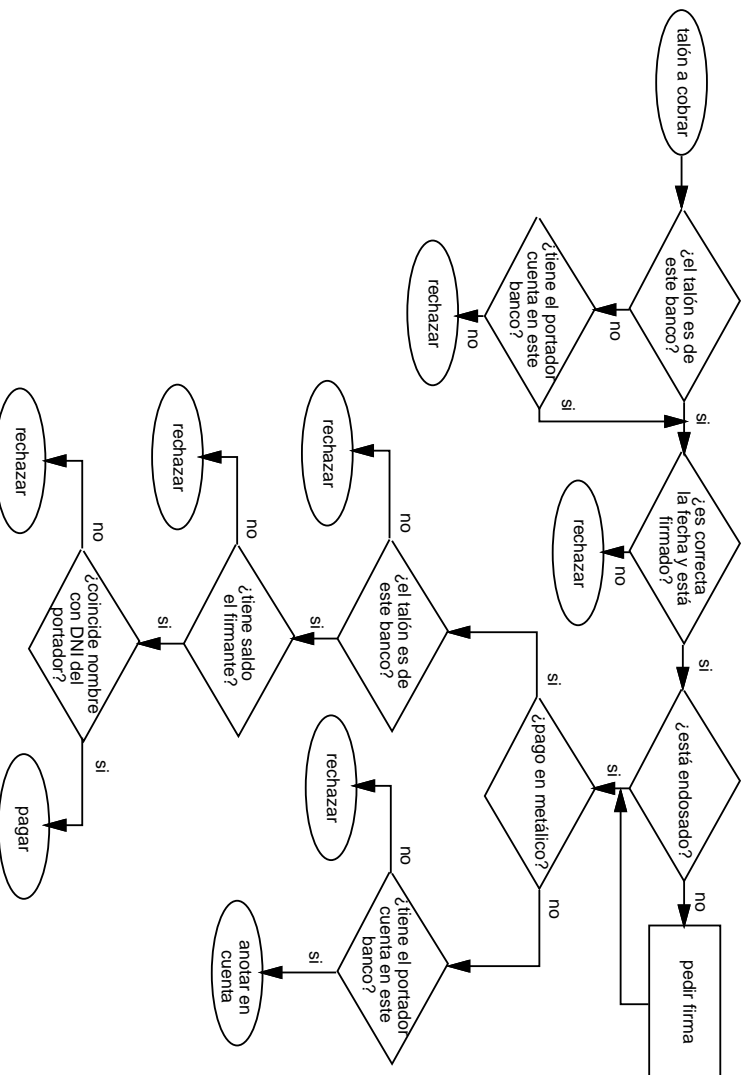
- **declarativa**: se especifican los objetos, las propiedades y las relaciones generales, y se deja al cuidado del agente que ha de resolver los problemas la aplicación de mecanismos generales de razonamiento, o
- **procedimental**: se especifica un procedimiento para resolver los problemas.

Declarativo vs. procedimental: ejemplo 1

```
superior(X,Y) :- jefe(X,Y).
superior(X,Y) :- jefe(X,Z), superior(Z,Y).
```

```
boolean superior(individuo X, individuo Y, conjunto C) {
    if (jefe(X,Y)) return true;
    else {
        C = sacar(C, X); C = sacar(C, Y); // C = C - {X,Y}
        while (!vacio(C)) {
            Z = buscar_en(C);
            if (jefe(X,Z))
                if (superior(Z,Y)) return true;
            C = sacar(C, Z); // C = C - {Z}
        }
        return false;
    }
}
```

Declarativo vs. procedimental: ejemplo 2 (1)



Declarativo vs. procedimental: ejemplo 2 (2)

- (1) SI NO talón_de_este_banco
Y NO portador_tiene_cuenta
ENTONCES rechazar
- (2) SI talón_de_este_banco
O portador_tiene_cuenta
ENTONCES talón_aceptable
- (3) SI fecha_correcta
Y talón_firmado
ENTONCES talón_cumplimentado
ENTONCES rechazar
- (4) SI NO talón_cumplimentado
ENTONCES rechazar
- (5) SI talón_cumplimentado
Y NO talón_endosado
ENTONCES pedir_firma
Y talón_endosado
- (6) SI talón_aceptable
Y talón_cumplimentado
Y talón_endosado
ENTONCES talón_completo
- (7) SI pago_metálico
Y NO talón_de_este_banco
ENTONCES rechazar

- (8) SI pago_metálico
Y talón_de_este_banco
Y NO firmante_saldo
ENTONCES rechazar
- (9) SI pago_metálico
Y NO portador_DNI
ENTONCES rechazar
- (10) SI NO pago_metálico
Y NO portador_tiene_cuenta
ENTONCES rechazar
- (11) SI talón_completo
Y NO pago_metálico
Y portador_tiene_cuenta
ENTONCES anotar
- (12) SI talón_completo
Y pago_metálico
Y talón_de_este_banco
Y firmante_saldo
Y portador_DNI
ENTONCES pagar
- (13) SI (hoy - fecha_talón) > 0
Y (hoy - fecha_talón) < 90
ENTONCES fecha_correcta

Declarativo vs. procedimental: comparación

	Declarativo	Procedimental
<i>Eficiencia</i>		≫
<i>Modularidad</i>	≈	≈
<i>Expresividad</i>	≫	
<i>Reutilización</i>	≫	
<i>Mantenimiento</i>	≫	

Tipos de conocimiento en un SBC

- **Factual**: hechos concretos
 - **Normativo**: general sobre el dominio
- Siempre declarativo

- **Táctico**: reglas de inferencia, metarreglas...
- **Estratégico**: de control, búsqueda

En principio procedimental, pero *arquitecturas multinivel*

Representación del conocimiento

- El nivel del conocimiento y las ontologías
- Tipos de conocimiento
- **La psicología como fuente de metamodelos**
 - Paradigmas en psicología
 - Modelo del procesamiento humano
 - Memoria semántica y memoria episódica
 - Redes semánticas
- Lenguajes formales de representación
- Lenguajes informales de representación
- Lógicas de descripciones
- Modelación del conocimiento incierto

Evolución de la psicología

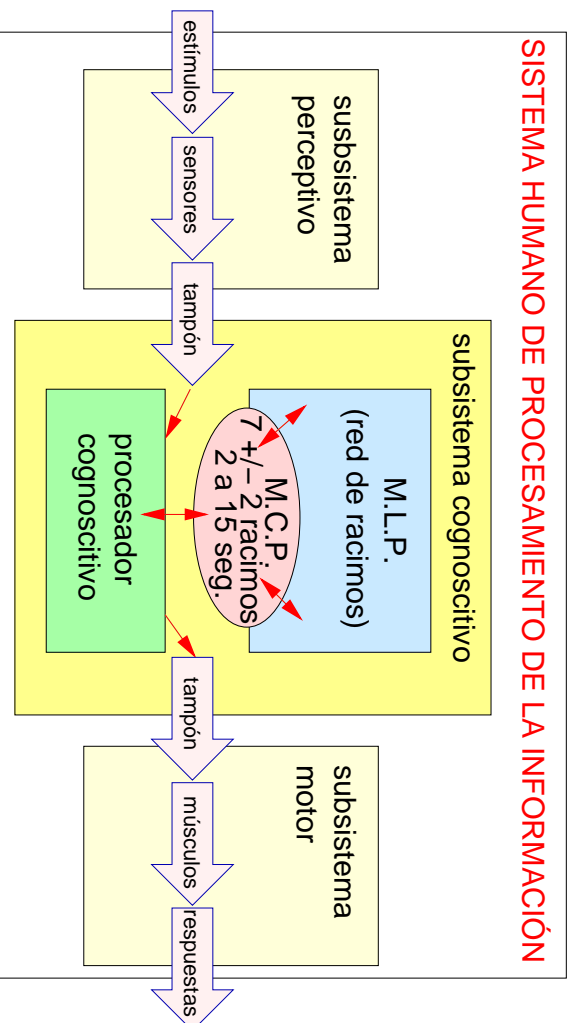
- **Subjetivismo** (introspección):
Hermann von Helmholtz, William James (\approx 1880)
- **Conductismo** (mediciones objetivas):
John Watson, Edward Thorndike (\approx 1910)
Burrhus Skinner (1957)
Crisis (cambio de paradigma) en los años 60
- **Cognitivismo** (modelos de concepciones):
Jean Piaget (1926), Kenneth Craik (1943),...
George Miller (1956): racimos («chunks»)
Ross Quillian (1968): memoria semántica
Allan Newell y Herbert Simon (1972):
modelo del procesamiento de la información

Modelo del procesamiento de la información

ENTORNO

(Newell y Simon, 1972)

SISTEMA HUMANO DE PROCESAMIENTO DE LA INFORMACIÓN



El procesador cognoscitivo selecciona perceptos y accede a la M.C.P. en ciclos *reconocimiento*—*acción* de 70 mseg.

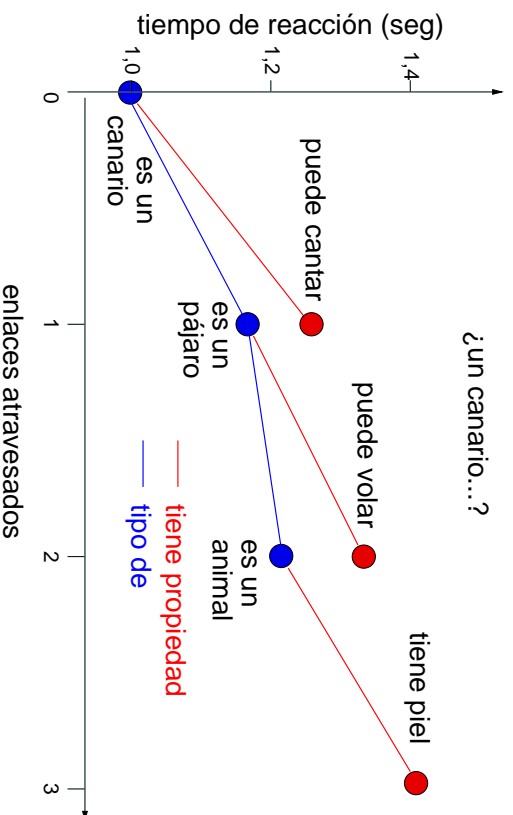
Memoria semántica vs. memoria episódica

- **Memoria episódica:** (Quillian, 1968)
recuerdo de eventos (episodios) experimentados:
ayer me mordió un perro, dónde estuve el domingo,
cómo fui a Tokio el año pasado, ...
(↪ Conocimiento factual)

- **Memoria semántica:**
guarda vocabulario, hechos, relaciones,
procedimientos. . . , sin referencia a cómo, dónde o
cuándo se han adquirido esos *conocimientos*:
los perros muerden, las discotecas aturden,
para viajes largos es mejor el avión. . .
Procede de *procesos de abstracción y generalización*.
(↪ Conocimiento normativo)

Memoria semántica: tiempo de acceso y estructura

(Collins y Quillian, 1969)



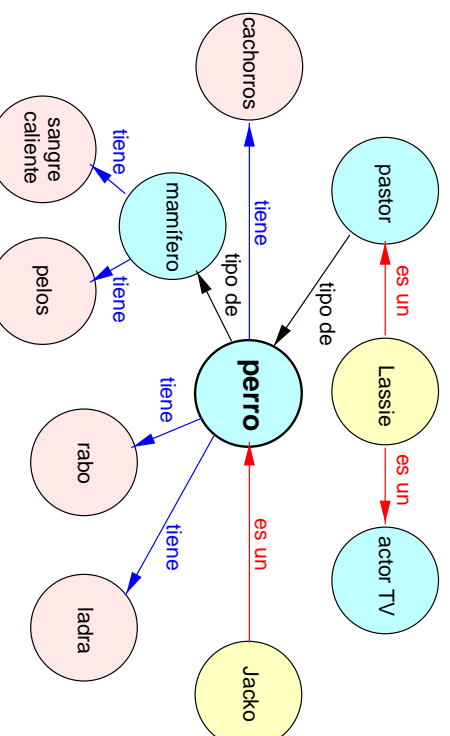
Hipótesis: los conceptos se almacenan asociándose entre sí directa o indirectamente, y para la recuperación hay que «recorrer» las asociaciones

Almacenamiento en la memoria semántica

(Lindsay y Norman, 1977)

Racimo (chunk): Red de conceptos interrelacionados que el procesador cognoscitivo almacena en o recupera de la memoria en un ciclo de reconocimiento—acción

Ejemplo:



↔ Redes semánticas ↔ Modelos de orientación a objetos

Representación del conocimiento

- El nivel del conocimiento y las ontologías
- Tipos de conocimiento
- La psicología como fuente de metamodelos
- **Lenguajes formales de representación**
 - Lenguajes lógicos
 - Lógica de proposiciones y lógica de predicados (FOL)
 - Extensiones de FOL
- Lenguajes informales de representación
- Lógicas de descripciones
- Modelación del conocimiento incierto

Lenguajes de representación: requisitos

- **Sintaxis**: interpretable por un procesador (el que implementa el proceso de razonamiento)
- **Semántica**: no ambigua; interpretación eficiente de algoritmos de razonamiento
- **Pragmática**: expresividad suficiente

En la práctica, elección lenguaje = elección herramienta

La **lógica formal** nos proporciona:

- Lenguajes para representación del conocimiento
- Modelos (funcionales) para implementar procesos de razonamiento

Lógicas de base: de proposiciones,

de predicados de primer orden (FOL)

Extensiones:

- multivaloradas, borrosa...
- de descripciones
- modales
- ...

Distintos tipos para **representar** distintas **conceptuaciones**

Representación en lógica de proposiciones: ejemplo

En el ejemplo del cajero, todas las reglas, salvo la (13), se pueden formalizar en lógica de proposiciones:

Variables proposicionales $t, p, r, c \dots$ con las **interpretaciones:**

$I(t)$ = «talón de este banco»; $I(p)$ = «portador tiene cuenta»

$I(r)$ = «rechazar»; $I(c)$ = «aceptar» ...

Sentencias:

R1: $\neg t \wedge \neg p \Rightarrow r$

R2: $I(t) \vee I(p) \Rightarrow c$

...

La (13), requiere lógica de predicados de primer orden:

$fecha(talón, X) \wedge fecha(hoy, Y) \wedge mayor(Y-X, 0) \wedge mayor(90, Y-X)$

$\Rightarrow fecha_corr(talón, sí)$

Lógica de predicados de primer orden («FOL»)

- ♣ **Limitaciones de la lógica de proposiciones: no permite representar**
 - razonamientos basados en propiedades o relaciones de tipo *general*
 - razonamientos basados en conocimientos *generales*
- ♣ En lugar de variables proposicionales, **predicados** que representan propiedades o relaciones
- ♣ Variables para representar objetos o individuos
- ♣ Cuantificadores

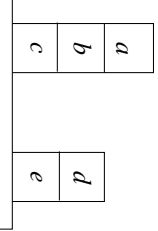
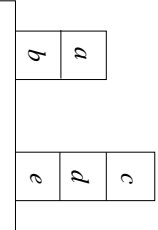
Lógica de predicados: concepciones

Para representación en FOL, una concepción debe estar formada por:

- Universo del discurso, \mathcal{U} : conjunto de *objetos*
- Conjunto de relaciones, \mathcal{R} ($R \subseteq \mathcal{U}^n$)
- Conjunto de funciones, \mathcal{F} ($F : \mathcal{U}^n \rightarrow \mathcal{U}$)
- Conjunto de conocimientos sobre el dominio:
 - Conocimientos *factuales*: relaciones y funciones definidas extensionalmente
 - Conocimientos *normativos*: relaciones y funciones definidas intensionalmente (enunciados compuestos con «y», «o», «no» y restricciones sobre los valores de enunciados: «si... entonces...»)

Ejemplo: mundo de los bloques.

Representación en FOL del mundo de los bloques

Símbolo	Interpretación		Conocimiento factual:	Conocimiento factual:
A	bloque a		$e(B) = A; e(C) = D; e(E) = D$	$e(B) = A; e(E) = D; e(D) = C$
B	bloque b		$l(A); i(D)$	$l(A); l(C)$
...	...		$b(C); b(E)$	$b(B); b(E)$
s	función Sobre		$s(A, B); s(B, C); s(D, E)$	$s(A, B); s(C, D); s(D, E)$
e	relación Encima-de		$m(A, B); m(B, C)$	$m(A, B); m(C, D)$
b	relación Bajo		$m(A, C); m(D, E)$	$m(D, E); m(C, E)$
l	relación Libre			
m	relación Más arriba			
...	...			

Conocimiento normativo:

$$y = e(x) \Leftrightarrow s(y, x)$$

$$l(x) \Leftrightarrow (\nexists y)(s(y, x))$$

$$m(x, y) \Leftrightarrow s(x, y) \vee (\exists z)(s(z, y) \wedge m(z, y))$$

Traducción de FOL a un lenguaje de implementación (Prolog)

Conocimiento factual:

$e(B) = A$: las funciones deben expresarse como relaciones

$s(A, B) \rightsquigarrow$ sobre(a, b).

$l(A) \rightsquigarrow$ Libre(a).

...

Conocimiento normativo:

$$l(x) \Leftrightarrow (\nexists y)(s(y, x)) \rightsquigarrow$$

Libre(X) :- not(sobre(Y, X)).

$$m(x, y) \Leftrightarrow s(x, y) \vee (\exists z)(s(z, y) \wedge m(z, y)) \rightsquigarrow$$

mas-arriba-que(X, Y) :- sobre(X, Y).

mas-arriba-que(X, Y) :- sobre(X, Z), mas-arriba-que(Z, Y).

Lógica de predicados: cosificación

Relaciones/propiedades sobre relaciones/propiedades

↪ *lógica de predicados de segundo orden*

Artificio para evitarla: **cosificación** (*reification*)

Ejemplo en el mundo de los bloques:

$\langle \{a, b, c, d\}, \{\}, \{\text{Rojo}, \text{Verde}, \text{Azul}, \text{Blanco}\} \rangle$

Para introducir *propiedades de los colores* conceptuamos a éstos como objetos e introducimos una función parcial, *Color*:

$\langle \{a, b, c, d, \text{rojo}, \text{verde}, \text{azul}, \text{blanco}\}, \{\text{Color}\}, \{\text{Oscuro}, \text{Claro}\} \rangle$

Color = $\{a \mapsto \text{blanco}, b \mapsto \text{rojo} \dots\}$

Claro = $\{\text{blanco}, \text{azul}\}$

etc.

Lógica de predicados: aspectos pragmáticos

- **Poder expresivo grande**, pero sólo concepciones asertóricas:
 - Verdadero o falso
 - Sin matices (« posible », « mañana », « el agente cree » .. .)
- Semidecidible (Gödel, 1930):
 - si $\{\varphi_1, \varphi_2, \dots\} \models \varphi$ existe una demostración de φ
 - si $\{\varphi_1, \varphi_2, \dots\} \not\models \varphi$, en general, no hay procedimiento finito
- Complejidad: problemas NP-completos
- Compromiso entre compleción y algoritmos eficientes

Limitaciones de FOL como lenguaje de representación

- **Declarativas:**
 - Carencia de estructura en \mathcal{U}
 - Expresión poco natural de conocimientos taxonómicos/mereológicos
 - Asunción mundo cerrado supone conocimiento completo
- **Procedimentales:**
 - Semidecidible \rightsquigarrow mundo cerrado
 - Algoritmos de unificación complejos
 - Lenguaje diseñado sin considerar aspectos procesales

Extensiones de la lógica de predicados

- **de orden superior:** se puede eludir con «cosificación»
- **no monótona:** razonamiento por defecto, mantenimiento de la verdad
- **multivaloradas y borrosa:** imprecisión e incertidumbre
- **modales:** creencias, deseos, intenciones...
abducción:

$(\forall X)(\text{hepatitis}(X) \Rightarrow \text{ictericia}(X))$

ictericia(*a*)

M(*hepatitis*(*a*))

$\diamond(\text{hepatitis}(a))$

Representación del conocimiento

- El nivel del conocimiento y las ontologías
- Tipos de conocimiento
- La psicología como fuente de metamodelos
- Lenguajes formales de representación
- **Lenguajes informales de representación**
 - Reglas
 - Redes semánticas y mapas conceptuales
 - Marcos
- Lógicas de descripciones
- Modelación del conocimiento incierto

Sistemas de reglas de producción

Aplicación de los « sistemas canónicos » de Emil Post (1943)
<base de datos, {producciones}, sistema de control>

Producciones (o reglas de producción):

$$(A, B) \text{ (o bien: } A \Rightarrow B)$$

- de tipo «implicación»:
SI antecedentes ENTONCES consecuente
- de tipo «activación»:
SI condiciones ENTONCES acción

$$A_1 \wedge A_2 \wedge \dots \wedge A_n \Rightarrow C$$

- A_1, A_2, \dots : literales (o sentencias, en general)
- C : un literal o una conjunción (no disyunción)

Reglas causales y reglas de diagnóstico

- **Reglas causales**: causa \Rightarrow efecto(s)
 - razonamiento basado en modelo (deductivo)
 - puede haber incertidumbre (aproximado: no deductivo)
 - «si hay un ataque el sistema *puede* caerse»
- **Reglas de diagnóstico**: efecto \Rightarrow posible causa (hipótesis)
 - razonamiento basado en hipótesis (no deductivo)
 - siempre hay incertidumbre (salvo excepciones)
 - «si el sistema se ha caído *puede* haber sido por un ataque»

Triplas objeto–atributo–valor

Forma restringida de FOL:

- Restricción sintáctica: relaciones binarias, $r(X, Y)$.
- Restricción semántica: *atributo(Objeto, Valor)*.
 - Mucho más expresiva que la lógica de proposiciones.
 - Sistema inferencial mucho más eficiente que el de FOL:
 - Implementación más estructurada.
 - Algoritmo de unificación más sencillo.

Triplas OAV: ejemplo de conceptualización

Objetos	Atributos	Valores
talón	banco	este_banco, otro_banco
	aceptable	si, no
	cumplimentado	si, no
	endosado	si, no
	completo	si, no
	fecha_correcta	si, no
	fecha	M/D/A
	acción	pagar, rechazar, anotar
portador	tiene_cuenta	si, no
	DNI	si, no
firmante	saldo	si, no
pago	modo	metalico, en_cuenta

Triplas OAV, ejemplo: codificación

```
IF (talon, banco, otro_banco)
  AND (portador, tiene_cuenta, no)
  THEN (talon, accion, rechazar)
  . . . . .
  IF (talon, fecha, X)
    AND ($system, $date, Y)
    AND ((Y-X) > 0)
    AND ((Y-X) < 90)
  THEN (talon, fecha_correcta, si)
```

(Sintaxis depende de la herramienta)

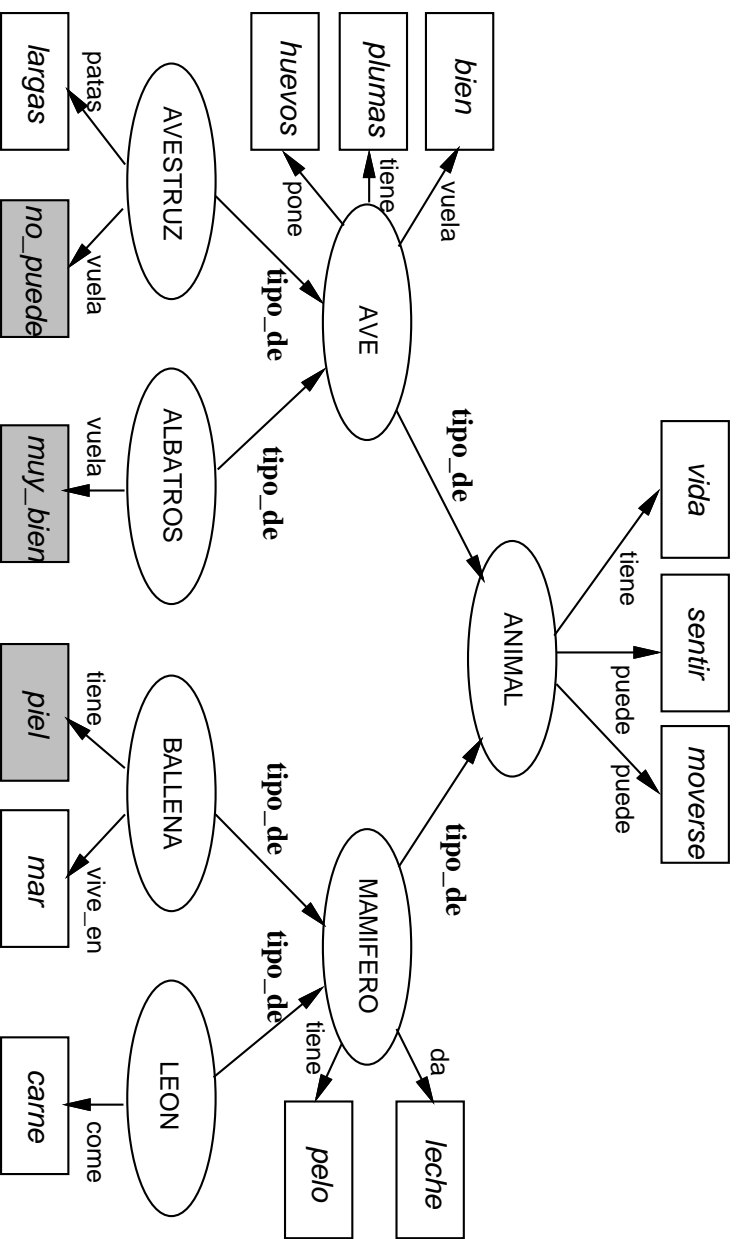
Redes semánticas

Conocimiento taxonómico

(herencia de propiedades)

- **nodos** (objetos, clases y propiedades)
 - de objeto, o de clase
 - de atributo (valor de propiedad)
 - normales
 - excepcionales
- **arcos orientados** (relaciones binarias)
 - de nodo objeto/clase a nodo clase
 - de nodo objeto/clase a atributo

Redes semánticas: ejemplo



De redes semánticas a lógica (1)

<u>sentencia en la red</u>	<u>sentencia lógica</u>
arco obj $\xrightarrow{\text{es-un}}$ clase	es_un(obj, clase) (pertenencia)
arco clase1 $\xrightarrow{\text{tipo-de}}$ clase2	tipo_de(clase1, clase2) (inclusión)
	$(\forall X)(\text{es_un}(X, \text{clase1})$ $\Rightarrow \text{es_un}(X, \text{clase2}))$
arco obj $\xrightarrow{\text{atrib}}$ valor	propiedad(obj, atrib, valor)
arco clase $\xrightarrow{\text{atrib}}$ valor	propiedad(clase, atrib, valor)
arco obj $\xrightarrow{\text{atrib}}$ excep	excep(obj, atrib, excep)
arco clase $\xrightarrow{\text{atrib}}$ excep	excep(clase, atrib, excep)

De redes semánticas a lógica (2)

Herencia como relación transitiva:

$$\begin{aligned} (\forall X)(\forall Y) & ((\text{es_un}(X, Y) \Rightarrow \text{hereda_de}(X, Y)) \wedge \\ & (\text{tipo_de}(X, Y) \Rightarrow \text{hereda_de}(X, Y)) \wedge \\ & ((\exists Z)(\text{es_un}(X, Z) \wedge \text{hereda_de}(Z, Y)) \wedge \\ & \Rightarrow \text{hereda_de}(X, Y))) \end{aligned}$$

Si no hubiese excepciones,

$$\begin{aligned} (\forall O)(\forall C)(\forall A)(\forall V) \\ & (\text{hereda_de}(O, C) \wedge \text{propiedad}(C, A, V) \\ & \Rightarrow \text{propiedad}(O, A, V)) \\ (\forall C1)(\forall C2)(\forall A)(\forall V) \\ & (\text{hereda_de}(C2, C1) \wedge \text{propiedad}(C1, A, V) \\ & \Rightarrow \text{propiedad}(C2, A, V)) \end{aligned}$$

De redes semánticas a lógica (3)

Herencia con excepciones:

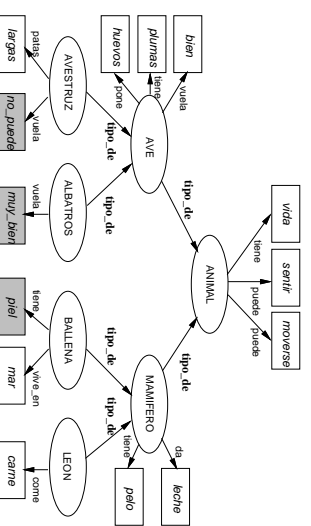
$$\begin{aligned} & (VO)(VC)(VA)(VV) \\ & \text{(hereda_de}(O, C) \wedge \text{propiedad}(C, A, V) \\ & \quad \wedge \neg \text{excep}(O, A, V)) \\ & \Rightarrow \text{propiedad}(O, A, V)) \end{aligned}$$

$$(VO)(VA)(VV)(\text{excep}(O, A, V) \Rightarrow \text{propiedad}(O, A, V))$$

(Y lo mismo para $C1, C2$)

Ejemplo de RS en Prolog: conocimiento factual

propiedad(animal, tiene, vida).
propiedad(animal, puede, sentir).
propiedad(animal, puede, moverse).
propiedad(ave, vuela, bien).
propiedad(ave, tiene, plumas).
propiedad(ave, pone, huevos).
propiedad(mamifero, da, leche).
propiedad(mamifero, tiene, pelo).
propiedad(avestruz, patas, largas).
propiedad(ballena, vive_en, mar).
propiedad(tigre, come, carne).



excepcion(avestruz, vuela, no_puede).
excepcion(albatros, vuela, muy_bien).
excepcion(ballena, tiene, piel).

Ejemplo de RS en Prolog: conocimiento normativo

Del dominio:

```
tipo_de(ave, animal).          tipo_de(albatros, ave).
tipo_de(mamifero, animal).     tipo_de(ballena, mamifero).
tipo_de(avestruz, ave).       tipo_de(tigre, mamifero).
```

General sobre herencia:

```
hereda_de(X,Y) :- tipo_de(X,Y).
hereda_de(X,Y) :- tipo_de(X,Z), hereda_de(Z,Y).

propiedad(Tipo, Atributo, Valor) :-
    hereda_de(Tipo, TipoPadre),
    propiedad(TipoPadre, Atributo, Valor),
    not excepcion(Tipo, Atributo, Valor).
propiedad(Tipo, Atributo, Valor) :-
    excepcion(Tipo, Atributo, Valor).
```

Ejemplo de RS en Prolog: consultas

■ «¿Tiene pelo el avestruz?»

```
?- propiedad(avestruz, tiene, pelo).
NO
```

■ «¿Quiénes pueden volar?»

```
?- propiedad(Quien, vuela, Como).
Quien = Avestruz      Valor = no_puede.
Quien = Albatros     Valor = muy_bien.
```

■ Mala concepción:

```
?- propiedad(ballena, tiene, vida).
NO
```

Marcos (*frames*)

Nodo de objeto o clase en la RS → marco: nombre

ranura1

ranura2

```
<ranura> ::= <nombre de relación>:<objeto relacionado> |
           <nombre de relación>:<clase relacionada> |
           <nombre de propiedad>:<valor de la propiedad> |
           <nombre de propiedad>: if_needed <procedimiento>
           <nombre de propiedad>: if_added <procedimiento>
<ranura> ::= <metaatributo>:<valor>
<nombre de relación> ::= es_un | tipo_de | <relacion específica>
«if_needed», «if_added»: procedimientos adjuntos, o adosados (procedural
attachment)
```

Ejemplo de marco

```
marco_avestruz
descripcion: caracterización básica de un avestruz
autor      : Prof. Ornito
fecha     : 31/12/2000
tipo_de   : ave
patas     : largas
vuela    : (exc) no_puede
detalles : if_needed {buscar en enciclopedia}
```


Evolución en la representación del conocimiento

Años 70 y 80:

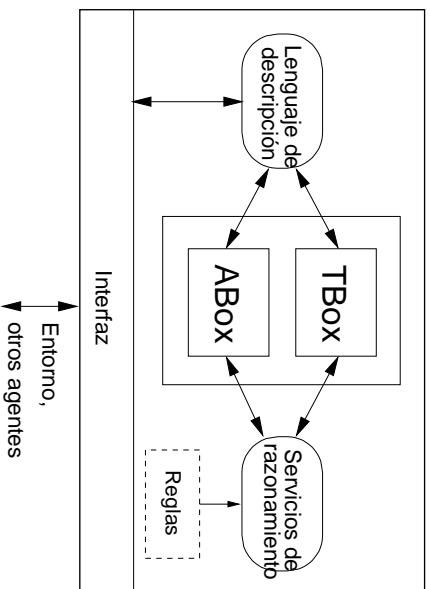
1. *Formalismos basados en lógica (especialmente, FOL y OAV):*
 - propósito general, muy expresivos, métodos de razonamiento sólidos..., pero
 - no decidibles (salvo con asunción de mundo cerrado), complejidad «incontrolada» ...
2. *Formalismos basados en redes semánticas y marcos:*
 - naturalidad de representación, métodos de razonamiento eficientes..., pero
 - estructuras y métodos *ad hoc*, carencia de caracterización semántica precisa...

Años 90 y 00: Fusión de 1 y 2 \rightsquigarrow lógicas de descripciones

Lógicas de descripciones: principios básicos

- En lugar de predicados unarios, **conceptos (o clases)**: El concepto «Abogado» representa a $\{x \mid \text{abogado}(x)\}$
- FOL orientada a razonamientos y a verdad o falsedad; DL orientadas a objetos, clases, pertenencia, roles...
- En lugar de predicados diádicos, **roles (o propiedades)**: « \exists tieneHijo.Abogado» representa a $\{x \mid (\exists y)(\text{tieneHijo}(x, y) \wedge \text{Abogado}(y))\}$
- **Constructores** de conceptos para formar *descripciones* de conceptos: «Abogado $\sqcap \exists$ tieneHijo.Abogado»
- **Familia de lenguajes** con diversos equilibrios entre expresividad (constructores) y complejidad del razonamiento, pero *todos decidibles con asunción de mundo abierto*

Agente basado en DL



TBox: «terminología» de un dominio, u «ontología» (conocimiento normativo):

Definiciones intensionales de conceptos y roles

ABox: Aserciones sobre objetos (conocimiento factual)

Agente basado en DL vs. KBS

El «agente basado en DL» es una reformulación de la idea de «sistema basado en conocimiento»:

- TBox = base de conocimientos (en Prolog, «reglas»)
- ABox = base de hechos (en Prolog, «hechos»)

Lo nuevo es el diseño de *lenguajes de descripción*

- adecuados para la *comunicación con otros agentes* (artificiales o humanos)
- que permitan una formulación de problemas de razonamiento *decidibles con asunción de mundo abierto*
- que conduzcan a *algoritmos de razonamiento eficientes*



Varias formas de expresar la sintaxis:

- **RDF/XML:** «Normativa» del W3C para intercambio entre agentes. Excesivamente verbosa para humanos.
- **Sintaxis abstracta:** De alto nivel y más legible, pero aún incómoda por excesivo anidamiento de paréntesis. Usada por el W3C para definir la semántica de OWL 1.0.
- **Sintaxis funcional:** Resuelve algunos problemas de la anterior para su procesamiento. Introducida para OWL 1.1.
- **Sintaxis de Manchester:** Más legible por humanos.
- **Otras:** OBO, SOF, Sydney...

OWL 2: sintaxis funcional y sintaxis de Manchester, ejemplo

- En DL: Alumno \equiv Persona $\sqcap \geq 1$ matriculadoEn
- En sintaxis funcional:

```
Namespace(a=<>)
Ontology(<#>
  Declaration(Class(a:Persona))
  Declaration(ObjectProperty(matriculadoEn))
  EquivalentClasses(a:Alumno
    ObjectIntersectionOf(a:Persona ObjectMinCardinality(1 a:matriculadoEn)))
  )
O bien:
EquivalentClasses(a:Alumno
  ObjectIntersectionOf(a:Persona ObjectSomeValuesFrom(a:matriculadoEn)))
```

- En sintaxis de Manchester:

```
Namespace: a <>
Ontology: <#>
Class: a:Persona
ObjectProperty: a:matriculadoEn
Class: a:Alumno
EquivalentTo:
  a:Persona
  and (a:matriculadoEn min 1 owl:Thing)
```

OWL 2: traducción del ejemplo a RDF/XML

```
<?xml version="1.0"??>
<!DOCTYPE rdf:RDF [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
] >
<rdf:RDF xmlns=""#""
  xml:base=""#"
  xmlns:owl=""http://www.w3.org/2002/07/owl#"
  xmlns:rdf=""http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <owl:Ontology rdf:about="""/>
  <owl:Class rdf:about=""Persona"/>
  <owl:ObjectProperty rdf:about=""matriculadoEn"/>
  <owl:Class rdf:about=""Alumno">
    <owl:equivalentClass>
      <owl:IntersectionOf rdf:parseType=""Collection">
        <rdf:Description rdf:about=""Persona"/>
        <owl:Restriction>
          <owl:onProperty rdf:resource=""matriculadoEn"/>
          <owl:minCardinality rdf:datatype=""&xsd;nonNegativeInteger">
            1
          </owl:minCardinality>
        </owl:Restriction>
      </owl:IntersectionOf>
    </owl:equivalentClass>
  </owl:Class>
</rdf:RDF>
```

Representación del conocimiento

- El nivel del conocimiento y las ontologías
- Tipos de conocimiento
- La psicología como fuente de metamodelos
- Lenguajes formales de representación
- Lenguajes informales de representación
- Lógicas de descripciones
- **Modelación del conocimiento incierto**
 - Incertidumbre e imprecisión
 - Lógica multivalorada
 - Factores de incertidumbre
 - Inferencia bayesiana
 - Lógica borrosa

Incertidumbre e imprecisión

Proposición incierta:

No se conoce, o no puede determinarse su verdad o falsedad

Si el paciente tiene fiebre y tose
y tiene dolores musculares
entonces padece gripe o padece bronquitis
o padece tuberculosis o ...

Proposición imprecisa:

Se refiere a una variable cuyo valor no se conoce o no puede determinarse con precisión

"¿Tose mucho el paciente?"

Grado de creencia y grado de verdad

- Proposiciones verdaderas o falsas, variables con un valor, pero el agente no lo sabe:
 - Incertidumbre: « Si A entonces B o C »
 - Imprecisión: « ¿Qué temperatura tiene el paciente? »
- Grado de verdad subjetivo:
 - « ¿Es muy mayor? »
Subjetivo, aunque el agente sepa que tiene 60 años
 - « ¿Tose mucho? »
¿Cuantificación de «tose»?

Incertidumbre e imprecisión: modelos

Modelos teóricos:

- Lógica multivalorada
- Lógica probabilística
- Lógica borrosa
- Teoría de la evidencia («credibilidad», «plausibilidad»)
- Teoría de la posibilidad («necesidad», «posibilidad»)

Modelos heurísticos más usados en ingeniería del conocimiento:

- Factores de incertidumbre (MYCIN)
- Inferencia bayesiana (PROSPECTOR)

Lógica multivalorada (de primer orden)

- **Sintaxis:** igual que en la binaria
- **Conceptuación:** como la binaria, pero las tuplas de las relaciones tienen *grados de pertenencia* $\in [0, 1]$
- **Interpretación:** igual que en la binaria
- **Satisfacción:** en lugar de $\models_{IA} S, g_{IA}(S) \in [0, 1]$:
 $g_{IA}(p(t_1, t_2, \dots, t_n)) = g_p$ sii
 $(A(t_1), A(t_2), \dots, A(t_n), g_p) \in I(p)$
- **Semántica:** convenios para calcular $g_{IA}(S)$ a partir de los grados de satisfacción de sus componentes

Leyes de Lukasiewicz (1930)

- $g(\neg S) = 1 - g(S)$
- $g(S1 \wedge S2) = \min(g(S1), g(S2))$
- $g(S1 \vee S2) = \max(g(S1), g(S2))$
- $g(S1 \Rightarrow S2) = 1$ si $g(S1) \leq g(S2)$
 $= 1 - g(S1) + g(S2)$ si $g(S1) > g(S2)$
- $g((\forall X)(S1(X))) = \min_X(g(S1(X)))$
- $g((\exists X)(S1(X))) = \max_X(g(S1(X)))$

Para el condicional se usa más:

$$g(S1 \Rightarrow S2) = \max(1 - g(S1), \min(g(S1), g(S2)))$$

Factores de incertidumbre (1)

- Escala, por ejemplo: $-1 \leq c \leq +1$
- Hechos: $A(c_A), B(c_B) \dots$
- Reglas: $E_i \Rightarrow H_j(c_{ij})$ (regla de diagnóstico)

Extensión heurística del modus ponens:

- $A(c_A)$
 $(A \Rightarrow B)(c_R)$

si $c_A > 0$ entonces $B(c_B)$, con $c_B = c_A \times c_R$

- $c(A_1 \vee A_2) = \max(c_{A_1}, c_{A_2})$
- $c(A_1 \wedge A_2) = \min(c_{A_1}, c_{A_2})$
- $c(\neg A) = -c_A$

Factores de incertidumbre (2)

Combinación de reglas:

(R_1 da la conclusión $H(c_1)$ y R_2 da $H(c_2)$)

- $c = c_1 + c_2 - c_1 \cdot c_2$ si $c_1 \cdot c_2 > 0$
- $c = \frac{c_1 + c_2}{1 - \min(|c_1|, |c_2|)}$ si $c_1 \cdot c_2 < 0$

Problemas con este enfoque:

- Malos resultados para dominios complejos, con cadenas causales largas
- Resultados catastróficos si se mezclan reglas causales con reglas de diagnóstico

Inferencia bayesiana y abducción

Abducción:

$(\forall X) (\text{hipótesis}_j(X) \Rightarrow \text{evidencia}_i(X))$
*evidencia*_{*i*}(*a*)

M(hipótesis_{*j*}(*a*))

◇(hipótesis_{*j*}(*a*))

Teorema de Bayes:

$$P(H_j|E_i) = \frac{P(H_j)P(E_i|H_j)}{\sum_j P(H_j)P(E_i|H_j)}$$

Pero las H_j deberían ser mutuamente excluyentes...

Inferencia bayesiana: exclusión de hipótesis

$$P(H_j|E_i) = \frac{P(H_j)P(E_i|H_j)}{P(H_j)P(E_i|H_j) + P(\bar{H}_j)P(E_i|\bar{H}_j)}$$

$$P(H_j|\bar{E}_i) = \frac{P(H_j)P(\bar{E}_i|H_j)}{P(H_j)P(\bar{E}_i|H_j) + P(\bar{H}_j)P(\bar{E}_i|\bar{H}_j)}$$

Con:

$$P(\bar{H}_j) = 1 - P(H_j)$$

$$P(\bar{E}_i|H_j) = 1 - P(E_i|H_j)$$

$$P(\bar{E}_i|\bar{H}_j) = 1 - P(E_i|\bar{H}_j)$$

Inferencia bayesiana: conoc. factual y normativo

Conocimiento factual: E_1, E_2, \dots

Conocimiento normativo:

- $P(H_1), P(H_2) \dots$
- $P(E_i|H_j) \equiv$ Si H_j es verdadera entonces aparece E_i con probabilidad $P(E_i|H_j)$
- $P(E_i|\bar{H}_j) \equiv$ Si H_j es falsa entonces aparece E_i con probabilidad $P(E_i|\bar{H}_j)$

Base de conocimientos:

$$\{P(E_i|H_j), P(E_i|\bar{H}_j), P(H_j)\}$$

Regla de inferencia (conocimiento tático): Bayes

Inferencia bayesiana: conocimiento táctico

$$P(H_j|E_i) = \frac{P(H_j)P(E_i|H_j)}{P(H_j)P(E_i|H_j) + P(H_j)P(E_i|\bar{H}_j)}$$

$$P(H_j|\bar{E}_i) = \frac{P(H_j)P(\bar{E}_i|H_j)}{P(H_j)P(\bar{E}_i|H_j) + P(\bar{H}_j)P(\bar{E}_i|\bar{H}_j)}$$

¿Conocimiento estratégico?

- ¿evidencias múltiples?
- ¿imprecisión/incertidumbre en las respuestas del usuario?

Inferencia bayesiana: combinación de evidencias

Si E_2 «viene después» de E_1 , probabilidad de H_j condicionada a E_2 en el contexto de E_1 :

$$P(H_j|E_1 \cap E_2) = \frac{P(H_j|E_1)P(E_2|H_j \cap E_1)}{P(H_j|E_1)P(E_2|H_j \cap E_1) + P(\bar{H}_j|E_1)P(E_2|\bar{H}_j \cap E_1)}$$

Suposición: H_j causas directas de E_i :

$P(E_i|H_j \cap E_k) = P(E_i|H_j)$, etc.

$$P(H_j|E_1 \cap E_2) = \frac{P(H_j|E_1)P(E_2|H_j)}{P(H_j|E_1)P(E_2|H_j) + P(\bar{H}_j|E_1)P(E_2|\bar{H}_j)}$$

Es decir, Bayes aplicado al resultado de tener en cuenta E_1

↪ **algoritmo de actualización sucesiva de las $P(H_j)$**

Inferencia bayesiana: suficiencia y necesidad

Medida de suficiencia: $S_{ji} = \frac{P(E_i|H_j)}{P(E_i|H_j)}$

- entre 1 (E_i indiferente) y ∞ (E_i suficiente)

Medida de necesidad: $N_{ji} = \frac{P(\bar{E}_i|H_j)}{P(\bar{E}_i|H_j)}$

- entre 1 (E_i indiferente) y 0 (E_i necesaria)

Possibilidad, o verosimilitud («odds»): $V = \frac{P}{1-P}$

Reformulación del teorema de Bayes:

$$V(H_j|E_i) = V(H_j) \cdot S_{ji}; \quad V(H_j|\bar{E}_i) = V(H_j) \cdot N_{ji}$$

Base de conocimientos: $\{H_j \Rightarrow E_i(S_{ji}, N_{ji})\}$

Inf. bayesiana: tratamiento de la incertidumbre

$$(1) V(H_j|E_i) = V(H_j) \cdot S_{ji} \quad (2) V(H_j|\bar{E}_i) = V(H_j) \cdot N_{ji}$$

- *del experto:* se traduce en S y N (y en $P(H_i)$)
 - *del usuario:* ante una pregunta sobre E_i , la evalúa en alguna escala convenida, por ejemplo, $[-1, +1]$
 - si $R(E_i) = 1$ se aplica (1)
 - si $R(E_i) = -1$ se aplica (2)
 - en otro caso se interpola mediante una función $M_{ji} = f(R(E_i))$ tal que:
 - $M_{ji}(+1) = S_{ji}$
 - $M_{ji}(-1) = N_{ji}$
 - $M_{ji}(0) = 1$
-

Inferencia bayesiana: modelo procesal

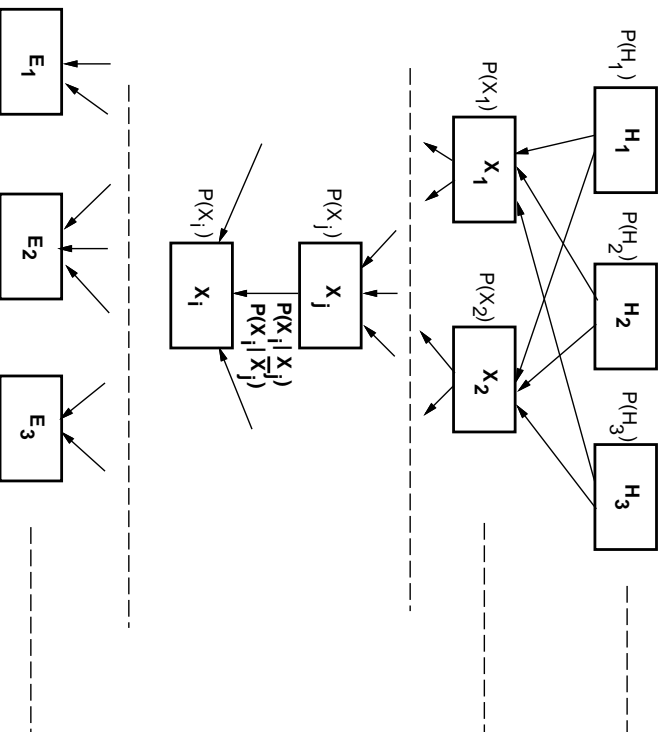
- el motor de inferencias evalúa en cada momento la evidencia más determinante, E_i
- si las respuestas extremas a esa evidencia no pueden «desbancar» a la hipótesis actualmente más probable, termina
- en otro caso, pregunta al usuario
- actualiza todas las $V(H_j)$:

$$V(H_j) \leftarrow V(H_j) \cdot M_{ji}$$

Redes bayesianas (1)

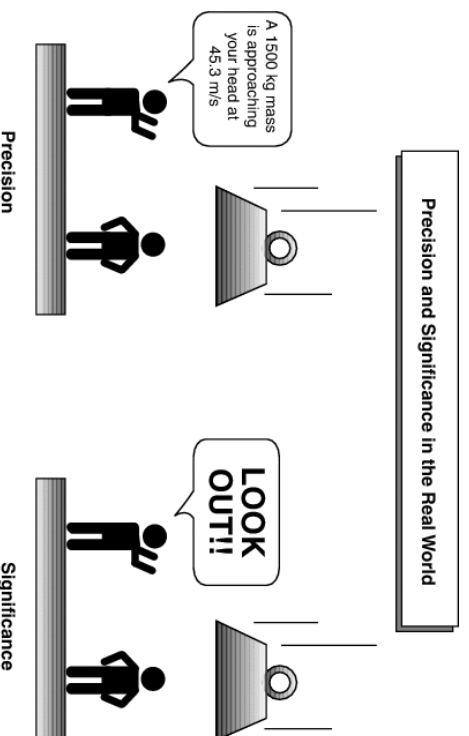
- El algoritmo de actualización sucesiva de $P(H_j|E_1 \cap E_2 \cap E_3 \dots)$ sólo es válido si H_j es *causa directa* de E_1, E_2, \dots
- Si se pueden identificar cadenas causales con dependencias directas se puede modelar el conocimiento como un grafo acíclico con
 - nodos: variables aleatorias X_i , que representan a las causas o las evidencias
 - arco de X_j a X_i : representa la relación causal directa de X_j sobre X_i , con etiquetas $P(X_i|X_j)$ y $P(X_i|\bar{X}_j)$ (o medidas de suficiencia y necesidad)

Redes bayesianas (2)



Precisión y significación

En sistemas complejos la *precisión da resultados poco significativos* (principio de incompatibilidad de Zadeh)



(De <http://www.mathworks.com/help/toolbox/fuzzy/>)

Modelación de lo impreciso

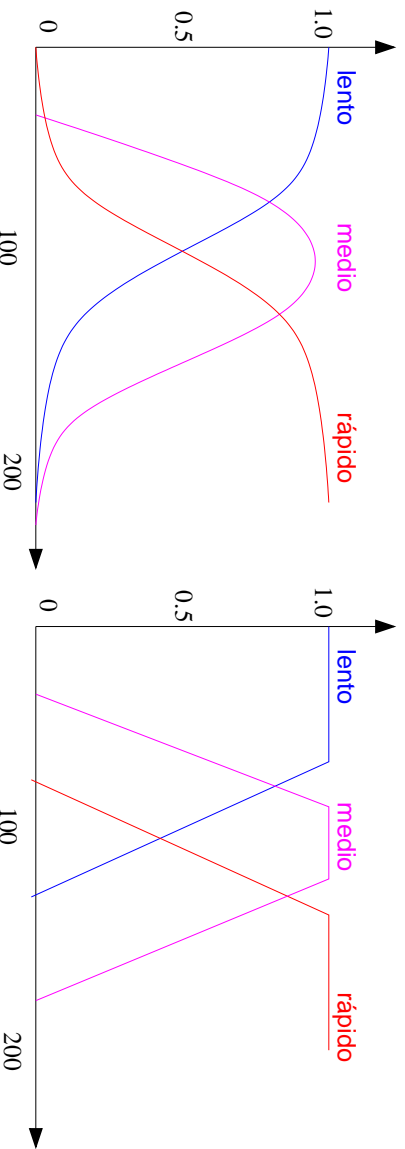
- «Levantarse suavemente el pie del embrague»
- «Sacar del horno cuando empieza a estar dorado»
- «Cuando hace algo de frío Ana suele ponerse el abrigo negro»
- «Hoy hace bastante frío»
- «Es casi seguro que llevará el abrigo negro»

«Si la temperatura del vapor vivo es alta y el caudal de inyección de vapor es bajo hay que bajar algo el caudal de inyección de vapor»

Conjuntos borrosos

Subconjunto borroso de $\mathcal{U} = \{x\}$: $C = \{(x|\mu_C(x))\}$
 $0 \leq \mu_C(x) \leq 1$: grado de pertenencia de x a C

Ejemplo:



Álgebra de conjuntos borrosos

- Complementación: $C = \bar{A} = \{x | \mu_C(x) = 1 - \mu_A(x)\}$
- Intersección: $C = A \cap B = \{x | \mu_C(x) = \min(\mu_A(x), \mu_B(x))\}$
- Unión: $C = A \cup B = \{x | \mu_C(x) = \max(\mu_A(x), \mu_B(x))\}$
- Producto: $C = AB = \{x | \mu_C(x) = \mu_A(x) \cdot \mu_B(x)\}$
- Potenciación: $C = A^\alpha = \{x | \mu_C(x) = \mu_A^\alpha(x)\}$
- Producto cartesiano:
 $C = A \times B = \{(x, y) | \mu_C(x, y) = \min(\mu_A(x), \mu_B(y))\}$
- Relación borrosa entre los conjuntos borrosos A y B : cualquier subconjunto de $A \times B$

Lógica borrosa

- **Sintaxis:** la clásica
+ partículas lingüísticas: *muy*, *poco*...
- **Conceptuación:** \mathcal{I} , \mathcal{R} (borrosas), \mathcal{F} (borrosas)
- **Interpretación:** como en lógica binaria, más:
 $I(\text{muy } p(X)) = I^2(p(X))$,
 $I(\text{poco } p(X)) = I^{1/2}(p(X))$
...

Lógica borrosa: satisfacción

- a) **Satisfacción graduada:** $g_{IA}(S)$, como en lógica multivalorada, en función de los componentes y para valores *concretos* de las variables
- b) **Satisfacción borrosa:** *significado*(S) = conjunto (o relación) borroso función de la variable (o las variables) de S:
- $S(\neg A) = \overline{S(A)}$
 - $S(A \wedge B) = S(A) \cap S(B)$
 - $S(A \vee B) = S(A) \cup S(B)$
 - $S(A \Rightarrow B) = \overline{S(A)} \cup S(B)$

Lógica borrosa: inferencia

Distintas extensiones del *modus ponens*:

- a) **Inferencia graduada:** Dados $p(X)$ y $p(X) \Rightarrow q(Y)$ y un valor *concreto* de X , obtener $q(Y)$ (o, en su caso, un valor concreto de Y)
- b) **Inferencia borrosa:** Dados $p(X)$ y $p(X) \Rightarrow q(Y)$, obtener la expresión lingüística del conjunto inducido, $q_p(Y)$

Sistemas expertos borrosos

- Aplicaciones numéricas (no simbólicas)
- Reglas de la forma «bajo(X) y no_muy_alto(Y) \Rightarrow abrir_poco(Z)»
- Proceso inferencial (inferencia graduada):
 1. Borrosificación (*fuzzification*): dados valores de las variables, grados de satisfacción de las premisas
 2. Inferencia MIN-MAX, o bien PROD-SUM:
 - a) para cada regla R_i , grado de satisfacción del antecedente, g_A^i
 - b) transformación de los conjuntos borrosos del consecuente:
 $\mu_C^i(Z) = \min(g_A^i, \mu_C(Z))$, o bien $\mu_C^i(Z) = g_A^i \cdot \mu_C(Z)$
 - c) composición: $\mu_C(Z) = \max(\mu_C^i(Z))$, o bien $\mu_C(Z) = \Sigma(\mu_C^i(Z))$
 3. Desborrosificación (valor de la variable del consecuente):
 - Centroide: el correspondiente al centro de gravedad de $\mu_C(Z)$
 - Promedio de los máximos
 - ...

Sistemas expertos borrosos: ejemplo (1)

bajo(t) = $1 - t/10$

alto(t) = $t/10$

($0 \leq t \leq 10$)

R1: si X es bajo e Y es bajo entonces Z es alto

R2: si X es bajo e Y es alto entonces Z es bajo

R3: si X es alto e Y es bajo entonces Z es bajo

R4: si X es alto e Y es alto entonces Z es alto

Borrosificación para algunos valores de X e Y:

X	Y	bajo(X)	alto(X)	bajo(Y)	alto(Y)	g_A^1	g_A^2	g_A^3	g_A^4
0,0	3,2	1,0	0,0	0,68	0,32	0,68	0,32	0,0	0,0
0,0	6,1	1,0	0,0	0,39	0,61	0,39	0,61	0,0	0,0
0,0	10,0	1,0	0,0	0,0	1,0	0,0	1,0	0,0	0,0
3,2	0,0	0,68	0,32	1,0	0,0	0,68	0,0	0,32	0,0
6,1	0,0	0,39	0,61	1,0	0,0	0,39	0,0	0,61	0,0
10,0	0,0	0,0	1,0	1,0	0,0	0,0	0,0	1,0	0,0
3,2	3,1	0,68	0,32	0,69	0,31	0,68	0,31	0,32	0,31

Sistemas expertos borrosos: ejemplo (2)

Inferencia para $X = 0, Y = 3, 2$:

a) $g_A^1 = 0, 68; g_A^2 = 0, 32; g_A^3 = 0; g_A^4 = 0$

b) Con MIN: $R1(Z) = \{Z/10 \text{ si } Z \leq 6, 8; 0, 68 \text{ si } Z \geq 6, 8\}$

$$R2(Z) = \{0, 32 \text{ si } Z \leq 6, 8; 1 - Z/10 \text{ si } Z \geq 6, 8\}$$

$$R3(Z) = 0, 0; R4(Z) = 0, 0$$

Con PROD: $R1(Z) = 0, 068 \cdot Z$

$$R2(Z) = 0, 32 - 0, 032 \cdot Z$$

$$R3(Z) = 0, 0; R4(Z) = 0, 0$$

c) Composición con MAX:

$$\text{Result}(Z) = \{0, 32 \text{ si } Z \leq 3, 2; Z/10 \text{ si } 3, 2 \leq Z \leq 6, 8 \text{ si } Z \geq 6, 8\}$$

Composición con SUM: $\text{Result}(Z) = 0, 32 + 0, 036 \cdot Z$

Desborrosificación:

- MIN-MAX y promedio de máximos: $Z = 8, 4$
- PROD-SUM y centroide: $Z = 6, 7$