



# Conocimiento

## Representación del Conocimiento





# Representaciones Estructuradas





- OBJETIVO: Representar grandes conjuntos de hechos de forma estructurada y comprimida.
  - Agrupar propiedades + representar objetos complejos.
  - Representar conocimiento taxonómico.
    - Relaciones: IS\_A, PART\_OF
    - Posibilidad de *herencia de propiedades*
  - Representar escenarios y secuencias “típicas” de acontecimientos.
- Difícil e “incómodo” de representar en lógica formal.
- Técnicas de representación:
  - Redes semánticas.
  - Frames (marcos)
  - Reglas de producción



# Redes Semánticas





- Método declarativo de representación del conocimiento.
- Estructuras gráficas (GRAFOS): codifican propiedades + conocimiento taxonómico sobre objetos
  - NODOS: Entidades del dominio (categorías u objetos)
  - ARCOS ETIQUETADOS: Relaciones entre entidades
- Enlaces (arcos etiquetados):
  - Un enlace UNIDIRECCIONAL por cq. relación/propiedad que podamos definir
  - Definen una relación binaria entre dos nodos



- Tipos de relaciones:
  - OCURRENCIA: vínculo entre un objeto particular y la clase de la que pertenece. Etiqueta:  $\in$ , PERTENECE
  - GENERALIZACIÓN: un objeto es un caso particular de otro objeto de naturaleza más general. Etiqueta: IS\_A
  - AGREGACIÓN: vínculo entre un objeto y los objetos que son parte de él. Etiqueta: PART\_OF
  - PROPIEDADES: vínculos entre objetos y características de dichos objetos
  - ACCIONES: vínculos de carácter dinámico
  - OTRAS RELACIONES ESPECÍFICAS

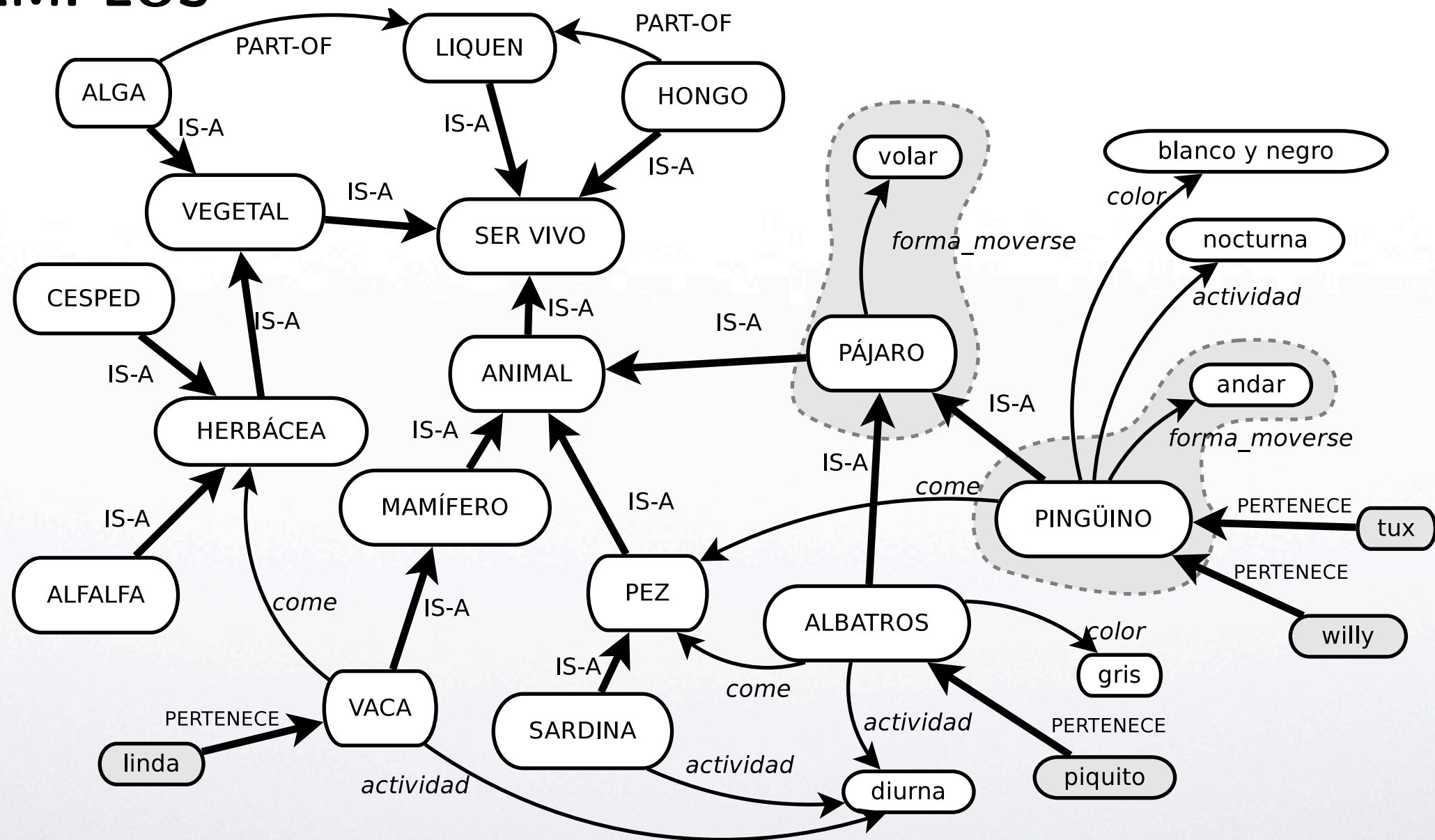




- Relación con Lógica Formal
  - Correspondencia directa Lógica de Predicado (L.P.) y Redes Semánticas (R.S.)
  - Toda R.S. puede representarse mediante fórmulas lógicas
  - Cualquier red semántica tendrá asociada un conjunto de tuplas OBJETO-RELACION-VALOR
    - $\text{Nodo1} + \text{Arco\_Etiquetado} + \text{Nodo2} \rightarrow \text{EtiquetaArco}(\text{Nodo1}, \text{Nodo2})$
  - **Ventajas R.S. respecto L.P.**
    - Notación gráfica facilita comprensión
    - Fácil especificar y manejar excepciones.
    - Modelo de ejecución más sencillo y eficiente, pero limitado
      - ◇ Inferencias y consultas en base a los enlaces



## EJEMPLOS







## TUPLAS OBJETO-ATRIBUTO-VALOR

OBJETO	ATRIBUTO	VALOR
pájaro	is_a	animal
pájaro	forma_moverse	volar
pájaro	actividad	diurna
pingüino	is_a	pajaro
pingüino	color	blanco_negro
pingüino	forma_moverse	andar
pingüino	actividad	nocturna
...	...	...
pepe	pertenece	pingüino
...	...	...



## LÓGICA PRÉDICADOS

### **OPCIÓN 1**

*is\_a(Pájaro, Animal)*

*is\_a(Pingüino, Pájaro)*

*pertenece(Pepe, Pingüino)*

*forma\_mover(Pájaro, Volar)*

*forma\_mover(Pingüino, Andar)*

...

$\forall x, s, f [is\_a(x, s) \wedge forma\_mover(s, f) \rightarrow forma\_mover(x, f)]$

...

### **OPCIÓN 2**

*pingüino(Pepe)*

*albatros(Juan)*

$\forall x \text{ pájaro}(x) \rightarrow animal(x)$

$\forall x \text{ pingüino}(x) \rightarrow \text{pájaro}(x)$

$\forall x \text{ albatros}(x) \rightarrow \text{pájaro}(x)$

$\forall x \text{ pajaro}(x) \rightarrow forma\_mover(x, Volar)$

$\forall x \text{ pajaro}(x) \rightarrow activo\_durante(x, Día)$

$\forall x \text{ pingüino}(x) \rightarrow forma\_mover(x, Andar)$

$\forall x \text{ pingüino}(x) \rightarrow color(x, Blanco\_Negro)$

...





# Mecanismos de Inferencia y Razonamiento

- **HERENCIA:** Mecanismo más importante
  - Toda propiedad de una categoría general es cierta para cq. ejemplo de sus categorías más específicas
  - Establece jerarquía taxonómica
  - Posibilidad de inferencias no monótonas
    - ◇ Herencia con excepciones ( = cancelación de la herencia)
    - ◇ Conocimiento por defecto
    - ◇ Ejemplo: *Pingüino es un Ave que no vuela*
  - PROBLEMAS:
    - ◇ Manejo herencia múltiple (por varias rutas)
    - ◇ Posibilidad de inferir conocimiento incorrecto



# Mecanismos de Inferencia y Razonamiento

- **RASTREO**

- Uso de la propiedad transitiva de algunas relaciones (PART\_OF, mayor\_que,...)
- Se infiere a partir de 2 arcos un "tercer arco" no disponible explícitamente
- PROBLEMAS: Posibilidad de inferir conocimiento incorrecto

- **EMPAREJAMIENTO**

- Construir, para un problema, un fragmento de red semántica.
- "Compararlo" con una red semántica completa (matching de grafos)
- Se deriva conocimiento implícito en la red global a partir de partes de una red, rellenando las partes "en blanco"





## VENTAJAS. REDES SEMÁNTICAS:

- Sencillas y fácil comprensión
- Representan relaciones jerárquicas de forma modular
- Eficiencia  $\Leftarrow$

## INCONV. REDES SEMÁNTICAS:

- Semántica poco clara (no "normalizada")
- Problemas al representar conocimiento no taxonómico
- Problemas al representar disyunciones, implicaciones, negaciones
  - *Una gallina no nada, Una gallina anda o salta, Las aves que no vuelan, tienen patas fuertes, ...*
- IDEM con cuantificación
- Manejo rutas conflictivas en herencia múltiple



## RAZONAMIENTO NO MONÓTONO

- IDEA BÁSICA: La adición de conocimiento nuevo puede dar lugar a que parte del conocimiento derivado que se había obtenido anteriormente sea ahora incorrecto.
- En Lógica de Predicados:
  - El razonamiento es siempre monótono
  - Añadir nuevos axiomas a la base de conocimiento no reduce el conjunto de *f.b.f.* que pueden ser demostradas.
  - Formalmente:  
Si  $\Phi \models \Psi$ , entonces, siendo  $\Delta$  un conjunto de *f.b.f.* consistente con  $\Phi$ , se verifica  $(\Phi \cup \Delta) \models \Psi$





- En ocasiones, el razonamiento es no monótono por naturaleza
  - Uso de *inferencias por defecto*
  - Se asume que algo es cierto, salvo que tengamos conocimiento contrario
  - Si aparece contradicción, debemos retractarnos de resultados obtenidos como consecuencia de las inferencias por defecto
  - En general, problema complejo
    - ◇ Necesidad mecanismos para rastrear conocimiento y razonamientos basados en creencias rebatidas
    - ◇ Util en razonamiento temporal y de sentido común



## RAZ. NO MONÓTONO EN REDES SEMANTICAS

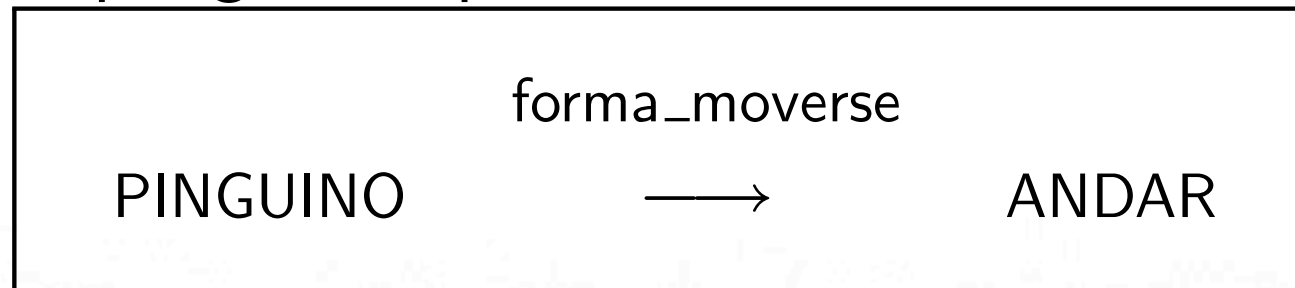
- Cancelación de la herencia
  - Implementación del mecanismo de herencia que permite incorporar un tipo limitado en razonamiento no monótono en R.S.
  - IDEA: Dar preferencia al conocimiento sobre categorías más específicas
  - Categorías generales aportan el conocimiento por defecto
  - Los niveles inferiores pueden cancelar ese conocim. por defecto
  - Análogo al *overriding* de propiedades y métodos en POO.





Ejemplo (sobre red semántica anterior)

- Supongamos que no existe el enlace:



- Un *pingüino* es un *pájaro*.  
Con los que sabemos ahora, por ser un *pájaro* (y mientras no se demuestre lo contrario) asumimos que *vuela* por aplicación de la herencia de propiedades.
- Si se añade el enlace anterior  
Aplicando cancelación de la herencia, el razonamiento por defecto "*los pingüinos vuelan*" es falso.
- Debemos retractarnos de los razonamientos basados en ese conocimiento por defecto al saber que los pingüinos "*andan*"



# Marcos





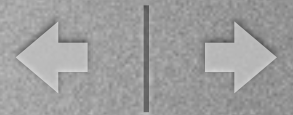
Método declarativo de representación del conocimiento.

OBJETIVO: Suministrar mecanismos de razonamiento por semejanza

Representación estructurada de conocimiento estereotipado

Un FRAME representa una entidad del mundo real

- Los hechos se agrupan en objetos
- FRAME = red semántica compleja



## ESTRUCTURA

- CABECERA: Etiqueta lingüística que da nombre al frame
  - Es representativa de la clase de objetos que describen
- SLOTS (ranuras): Contienen la info. relativa a la cabecera del FRAME
  - Conjunto de pares atributo-valor
  - Representan: propiedades de la clase, objetos, propiedades de los objetos, subobjetos, relaciones entre frames, etc...
  - Slots estructurados en niveles: cada indentación especializa nivel superior
  - Pueden ser propios o heredados





Pueden incorporar:

- Metaconocimiento: info. sobre el frame, ayuda a manipular el conocimiento
- Información procedimental (DEMONS)
  - se activan cuando se accede al slot
  - permiten: uniones procedimentales entre frames, interacción mundo exterior
  - aportan carácter dinámico
  - ej: IF\_NEEDED, IF\_ADDED, IF\_REMOVED, etc

RAZONAMIENTO e INFERENCIA:

- HERENCIA: Mediante inclusión de slots IS\_A
- EMPAREJAMIENTO: Relleno de slots vacíos.



## VENTAJAS

- Permiten trabajar con info. incompleta
- Fácil de implementar y ampliar
- Herencia de propiedades ( “infieren” conocimiento no representado explícitamente)
- Interacción mundo exterior y cooperación esquemas de representación procedim

## INCONV.

- Problemas para representar conocimiento no taxonómico
- IDEM con disyunciones e implicaciones





## RED SEMÁNTICA ANTERIOR

```
FRAME: pájaro
  is_a: animal
  forma_moverse: volar
  actividade: diurna

FRAME: pingüino
  is_a: pájaro
  color: blanco_y_negro
  forma_moverse: andar
  actividad: nocturna
  tamaño: mediano

FRAME: willy
  is_a: pingüino
  tamaño: grande
...
```

## INDENTACIÓN DE SLOTS

```
FRAME: casa
  elementos_básicos:
    cimientos:
      composición: hormigón
      localización: bajo_tierra
    paredes:
      composición: ladrillo
      elementos_complementarios:
        puertas:
          tipo: convencional
          material: madera
        ventanas:
          ....
  tejado:
    material: pizarra
    color: negro
    localización: arriba
```



## METACONOCIMIENTO y DEMONS

FRAME: sintomas\_065

info:

formato: reducido

uso: informativo

autor: Dr. Pepe

version: 0.99

paciente: Sr. Pérez

temperatura: alta

sudoración: presente

dolor\_muscular: ausente

descompos\_organos: ausente

FRAME: juan

is\_a: empleado

IF\_NEEDED: cargar\_base\_datos

IF\_REMOVED: print\_borrado\_base\_datos





# Sistemas Basados en Reglas



Reglas  $\approx$  Operadores en búsquedas en espacio de estados

Inferencia similar al *MODUS PONENS* (con restricciones)

- Sintaxis relajada
- Se permiten acciones en los consecuentes
- Es el mecanismo de control quien determina que inferencias se llegarán a realizar realmente

TIPOS de SISTEMAS

- En función de la sintaxis de las reglas y del mecanismo de control ( $\approx$ búsqueda)





# Sistemas Inteligentes



***Juan Carlos González Moreno***

**Curso 2012-13**



- **SIST. ENCADENAM. HACIA ADELANTE**(dirigidos por los datos)
  - Una regla está **ACTIVADA** si sus antecedentes emparejan con algunos hechos del sistema
  - En IF ALL, todos. En IF ANY, al menos uno.
  - Se parte de los hechos ya confirmados en el sistema
  - Se razona hacía adelante buscando antecedentes que emparejen
- **SIST. ENCADENAM. HACIA ATRÁS** (dirigido por los objetivos)
  - Una regla está **ACTIVADA** si sus consecuentes emparejan con algunos hechos del sistema
  - Se comienza con una hipótesis
  - Se razona hacía atrás buscando consecuentes que emparejen





- Una misma regla puede ser activada varias veces por distintos conjuntos de hechos (ó de hipótesis)
- Cada una de esas activaciones constituye una **instancia** de la regla activada
- El motor de inferencias tratará esas instancias por separado, manteniendo siempre traza de cuales fueron los hechos (ó hipótesis) que provocaron su activación



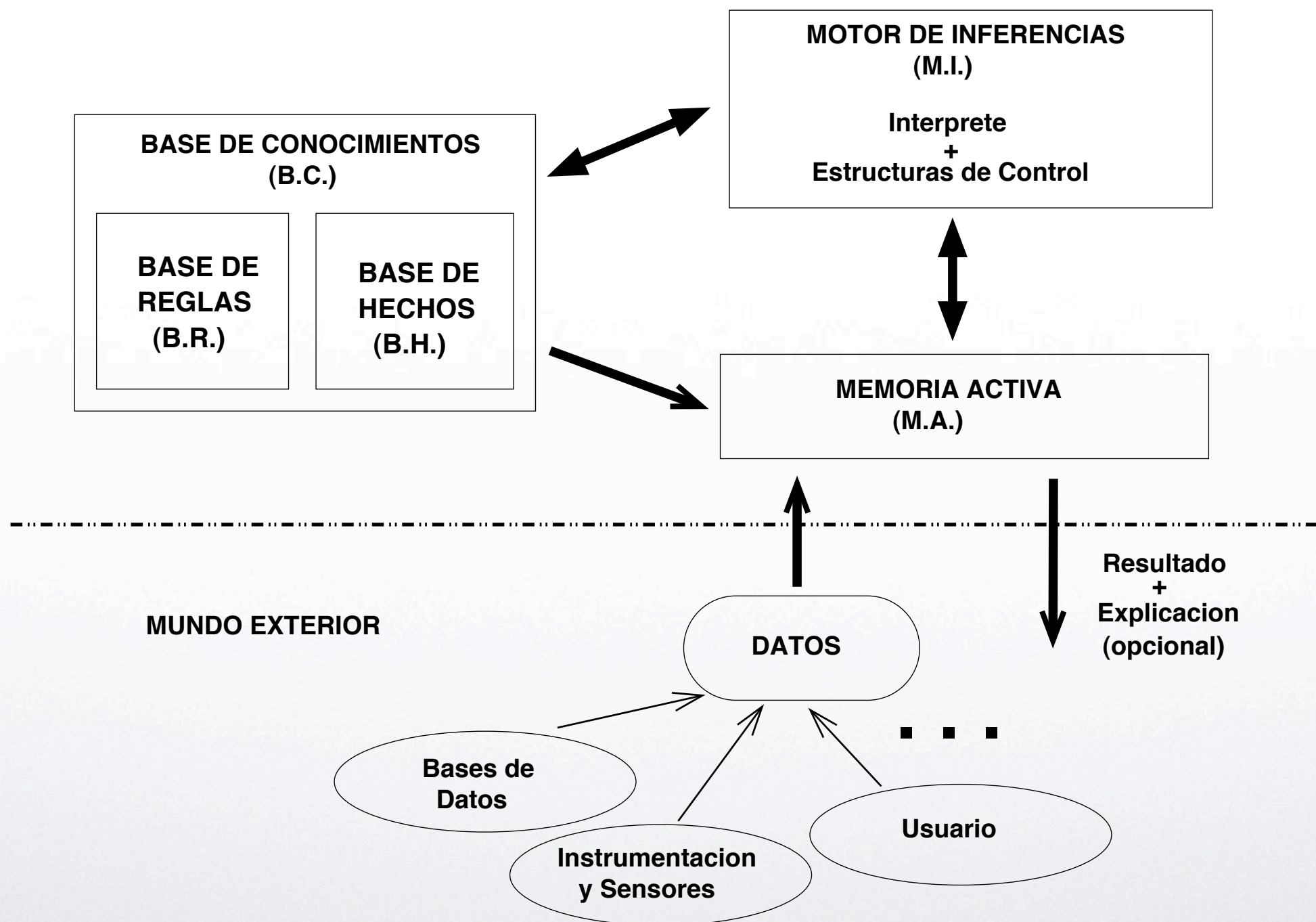
## CARACTERÍSTICAS

- Modularidad: reglas = pequeñas cantidades de conocimiento (relativamente) independiente
- Incrementalidad/Modificabilidad: posible añadir/cambiar reglas con relativa independencia
- Naturalidad y Transparencia: representación del conocimiento próxima y comprensible por personas
- Capacidad de generar explicaciones

## GENERACIÓN de EXPLICACIONES

- Posibilidad de “explicar” el porque de un resultado
- Devolver a usuario la cadena de reglas empleadas
  - Combinar reglas y hechos del árbol de búsqueda según las conectivas
- Incrementan la “aceptación” del resultado ofrecido (dominios críticos)







## COMPONENTES

1. BASE DE CONOCIMIENTOS (BC)
  - Reúne todo el conocimiento del sistema
  - Formada por base de reglas(**BR**) + base de hechos(**BH**)
2. MEMORIA ACTIVA (MA)
  - Colección de hechos, representando el estado actual del problema
  - Actúa como “disparador” del motor de inferencias
  - Refleja los cambios en el conocimiento del sistema
  - Interactúa con el mundo exterior (usuario, bases de datos, etc...)





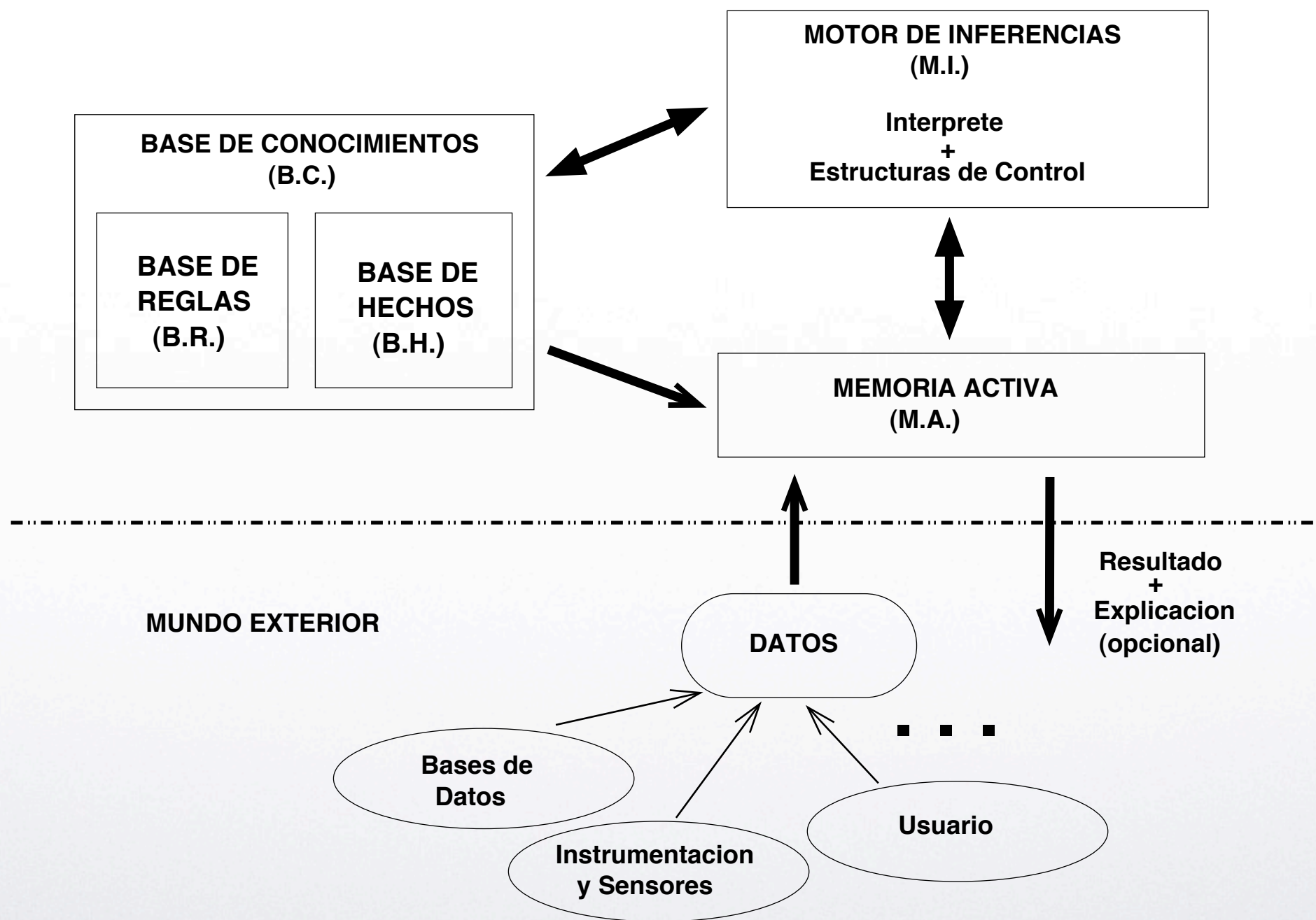
- Contiene:
  - Datos iniciales del problema + datos incorporados con posterioridad
  - Hechos establecidos durante procesos de inferencia
  - Hipótesis de trabajo, metas y submetas aún no confirmadas
- Determina las REGLAS ACTIVADAS
  - Conjunto de las *instancias* de reglas que están en condiciones de ser ejecutadas
  - Encadenam. hacia adelante: antecedentes representados en la **MA**
  - Encadenam. hacia atrás: consecuentes representados en la **MA**
  - El **MI** decide cual o cuales de las reglas ACTIVAS se ejecutará



## MOTOR DE INFERENCIAS (MI)

- Controla el funcionamiento del sistema
  - procesos de emparejamiento
  - selección de reglas
  - ejecución de reglas
  - rutinas externas
- Compuesto por : intérprete + estructuras de control
- Separado e independiente de la **BC**
- Funcionamiento análogo a búsqueda en espacio de estados
  - ESTADO: Representado por conj. hechos de **MA**
  - OPERADORES: Reglas de la **BR**
  - Con encadenam. hacía atrás : búsqueda en grafos AND-OR (búsqueda por subobjetivos)







## TAREAS MI:

- Ciclo básico:
  1. Examen de la **MA** y selección de reglas activas (*emparejamiento*)
    - depende del tipo de encadenamiento
  2. Selección reglas a ejecutar (*resolución conflictos*), en función de:
    - estrategia de exploración
    - modelos de resolución de conflictos
  3. Ejecución reglas y actualización de la **MA**
  4. Mantenimiento del autoconocimiento del sistema
    - control de instancias de reglas activadas (agenda)
    - control de reglas ejecutadas
    - control del orden de activación y del orden de ejecución
    - mantener orden de los hechos en la **MA**