

Ficha docente de la asignatura Inteligencia Artificial I



Tema 1. Introducción a la IA.

Principios fundamentales. Evolución histórica. Aplicaciones. Retos y características de los problemas de la IA. Agentes, visión artificial y robótica. Tipos de IA según el conocimiento y métodos de razonamiento.



Tema 2. Resolución de problemas con búsqueda.

Representación de problemas. Algoritmos de búsqueda heurística. Búsqueda local. Optimización. Algoritmos genéticos. Aprendizaje por refuerzo.



Tema 3. Representación básica del conocimiento y razonamiento.

Propiedades de la representación del conocimiento. Tipos de razonamiento (genera y prueba, análisis de medios y fines, reducción de problemas). Sistemas de producción. Sistemas basados en casos. Sistemas recomendadores.

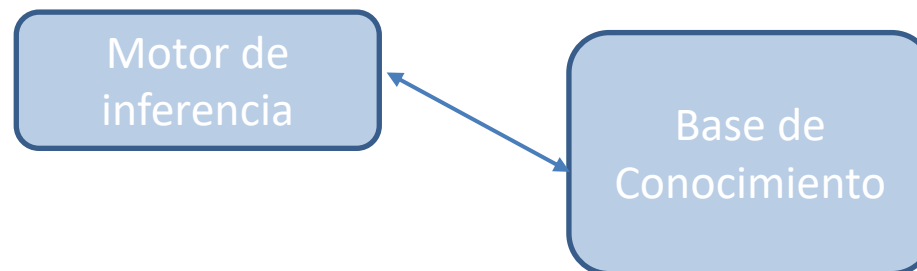
Tema 3: Sistemas basados en Conocimiento

- Representación básica del conocimiento y razonamiento
- Propiedades de la representación del conocimiento
- Sistemas basados en reglas
- Razonamiento por analogía. Sistemas CBR
- Sistemas recomendadores.

Formalizar consiste en representar simbólicamente los conocimientos de un dominio utilizando alguno de los formalismos de representación de conocimientos existentes (paso del *modelo conceptual* al *modelo formal*). Hay distintos tipos de formalismos: basados en conceptos (*marcos*), basados en relaciones (*redes semánticas*) y basado en acciones.

Sistemas Basados en Conocimiento

- ¿Qué los caracteriza? ¿Son lo mismo los SBC que los Sistemas Expertos?
- La característica principal es la separación entre:
 - Conocimiento específico del problema (Base de conocimiento)
 - reglas, jerarquías, fórmulas lógicas, heurísticas, scripts, grafos, marcos, diccionarios, casos,
 - Motor de inferencia para solucionar el problema
- Otras características:
 - **Representación explícita del conocimiento**
 - Capacidad de razonamiento es independiente del dominio
 - Capacidad para explicar sus conclusiones y el proceso de razonamiento



Información vs conocimiento

Información: Conjunto de **datos** sin interpretar

Entrada al sistema

Datos numéricos de una analítica de sangre

Resultado deportivo: 30-15

Italia 0-2 Canadá | (6-2, 3-6, 3-2)

España - Gran Bretaña: 7-6 y 7-6

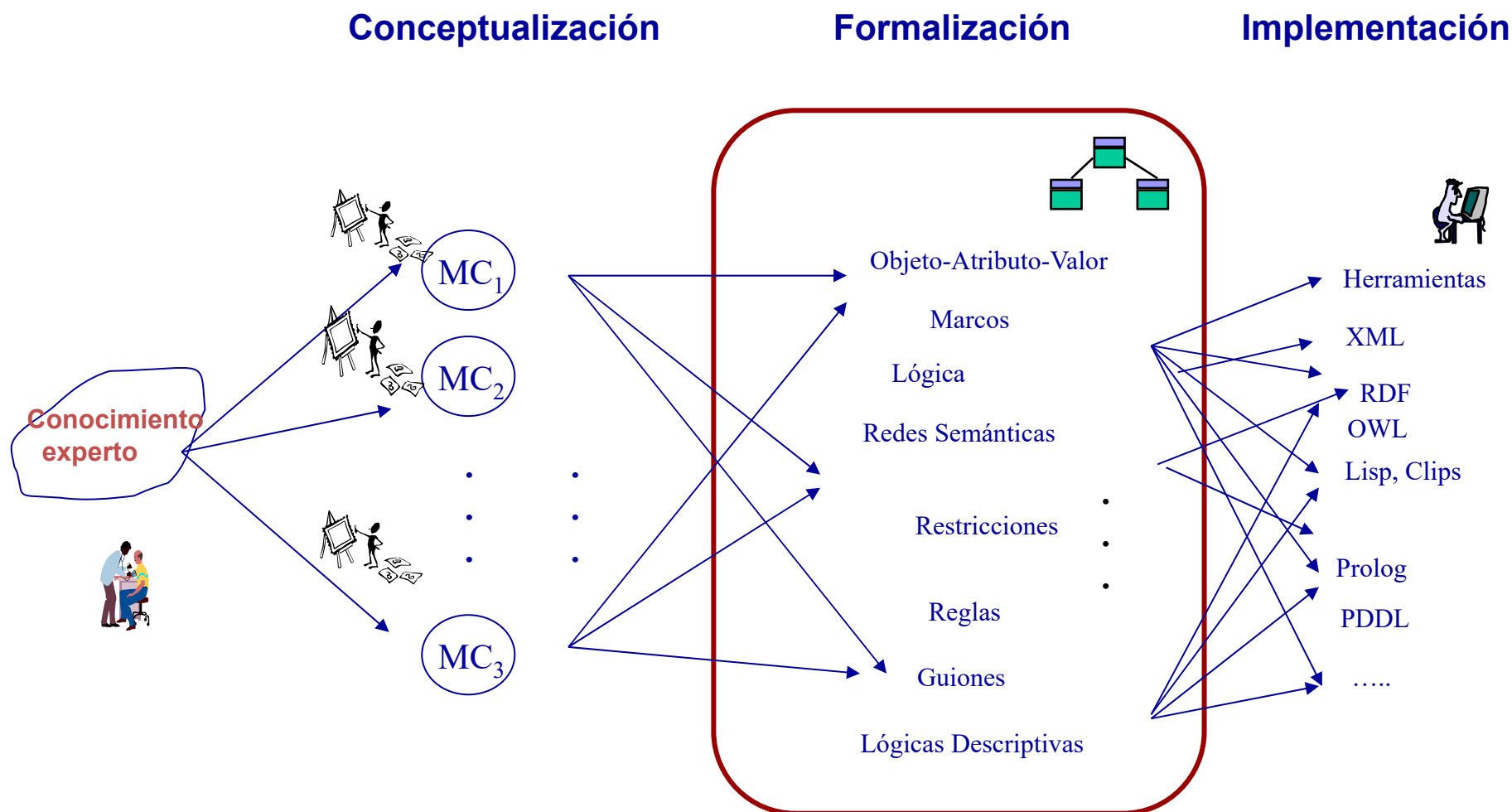
Tipo de Muestra:	Sangre Total: Aditivo EDTA		Facultad	Intervalo de Referencia
Magnitud	Resultado	Unidades		
HEMATIES	4.42	10 ⁶ /μL		4.5 - 5.9
HEMOGLOBINA	13.8	g/dL		13 - 17.4
HEMATOCRITO	39.7	%		41.5 - 50.4
VCM	89.9	fL		82 - 98
HCM	31.2	pg		27 - 32
CHCM	34.7	g/dL		30 - 34
ADE	13.4	%		12 - 15
ERITROBLASTOS #	0.02	10 ³ /μL		0.0 - 0.0
ERITROBLASTOS %	0.4	%		4 - 11
LEUCOCITOS	5.1	10 ³ /μL		40 - 75
NEUTROFILOS %	55.2	%		20 - 45
LINFOCITOS %	32.7	%		2 - 10
MONOCITOS %	8.6	%		0 - 5
EOSINOFILOS %	2.7	%		0 - 2
BASOFILOS %	0.8	%		1.6 - 7
NEUTROFILOS #	2.82	10 ³ /μL		1.2 - 4
LINFOCITOS #	1.67	10 ³ /μL		0.12 - 0.8
MONOCITOS #	0.44	10 ³ /μL		0 - 0.4
EOSINOFILOS #	0.14	10 ³ /μL		0 - 0.2
BASOFILOS #	0.04	10 ³ /μL		120 - 400
				8 - 11

- **Conocimiento:** Conjunto de datos que modelan la experiencia que se tiene sobre un dominio o que surgen de interpretar la información de entrada al sistema
 - La interpretación de los valores de la analítica de sangre o los **métodos para diagnosticar enfermedades a partir de los valores de la analítica de sangre**
 - Saber como se puntúa en un deporte

Representación del conocimiento. Componentes

- Representar el conocimiento consiste en escribir con un lenguaje determinado la información de un dominio
 - **Sintaxis**: símbolos y conjunto de reglas para combinarlos
 - **Semántica**: significado de las expresiones construidas
- **Representación simbólica: las etiquetas simbólicas nos dan “poder” sobre los objetos que representan.**
 - Si tiene nombre podemos asociar con esta etiqueta todo lo que sabemos sobre este símbolo.
- Google usa búsqueda sintáctica o semántica?
- La aproximación **subsimbolica (o conexionista)** **NO** representa explícitamente el conocimiento

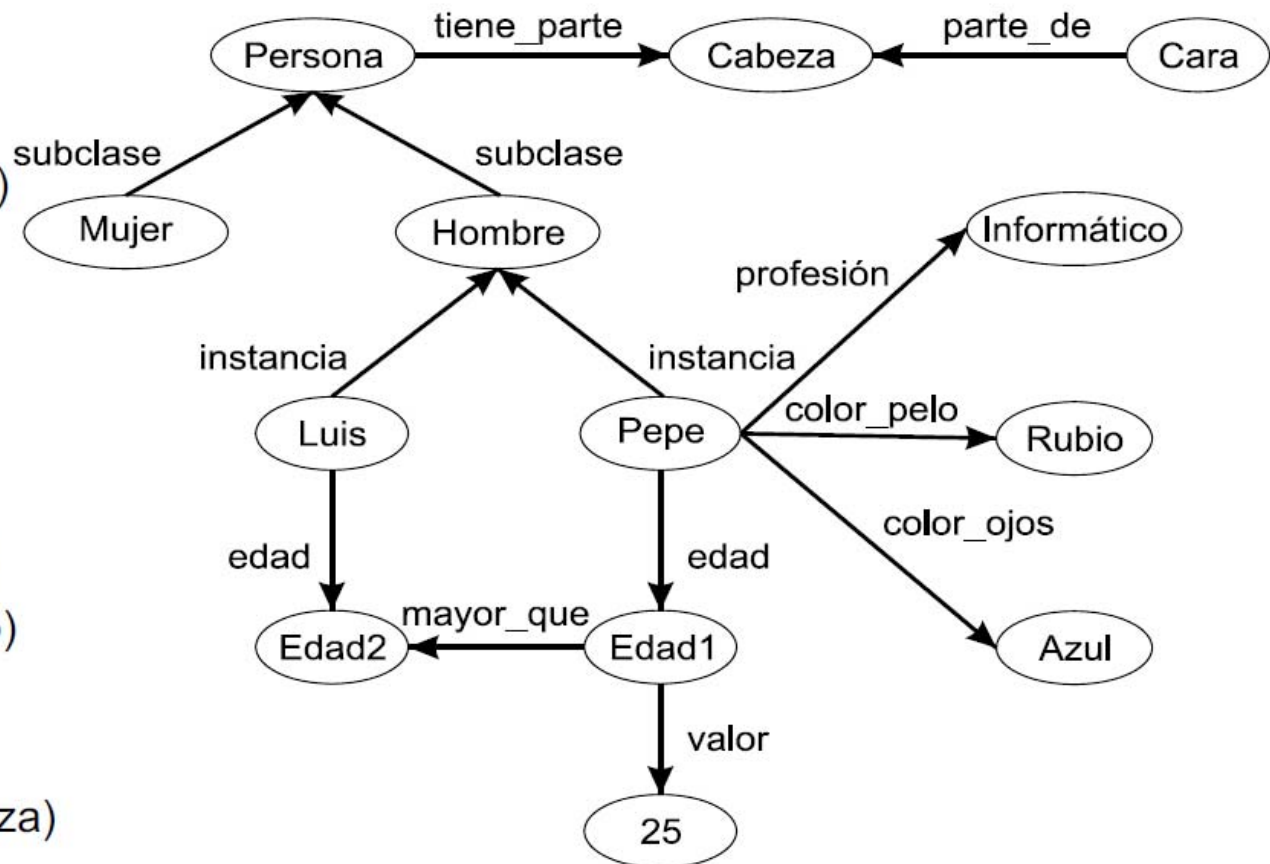
Conceptualización, Formalización e Implementación



Ejemplo base de conocimiento: red semántica (se verán en el 2º cuatrimestre)

- Una red semántica es la forma natural de representar relaciones correspondientes a instancias cerradas de **predicados binarios** en lógica

```
subclase(Mujer, Persona)
subclase(Hombre, Persona)
instancia(Pepe, Hombre)
instancia(Luis, Hombre)
edad(Pepe, Edad1)
edad(Luis, Edad2)
valor(Edad1, 25)
mayor_que(Edad1, Edad2)
profesión(Pepe, Informático)
color_pelo(Pepe, Rubio)
color_ojos (Pepe, Azul)
tiene_Parte(Persona, Cabeza)
parte_de(Cara, Cabeza)
```



Sistemas de reglas

IF the infection is meningitis
AND patient has evidence of serious skin or soft tissue
infection
AND organisms were not seen on the stain of the culture
AND type of infection is bacterial
THEN There is evidence that
those seen on cultures or
is *Staphylococcus coagulans*

MYCIN Sample Rule

Human-Readable Format

IF the stain of the organism is gram negative
AND the morphology of the organism is rod
AND the aerobiocity of the organism is gram anaerobic
THEN there is strong evidence (0.8)
that the class of the organism is enterobacteriaceae

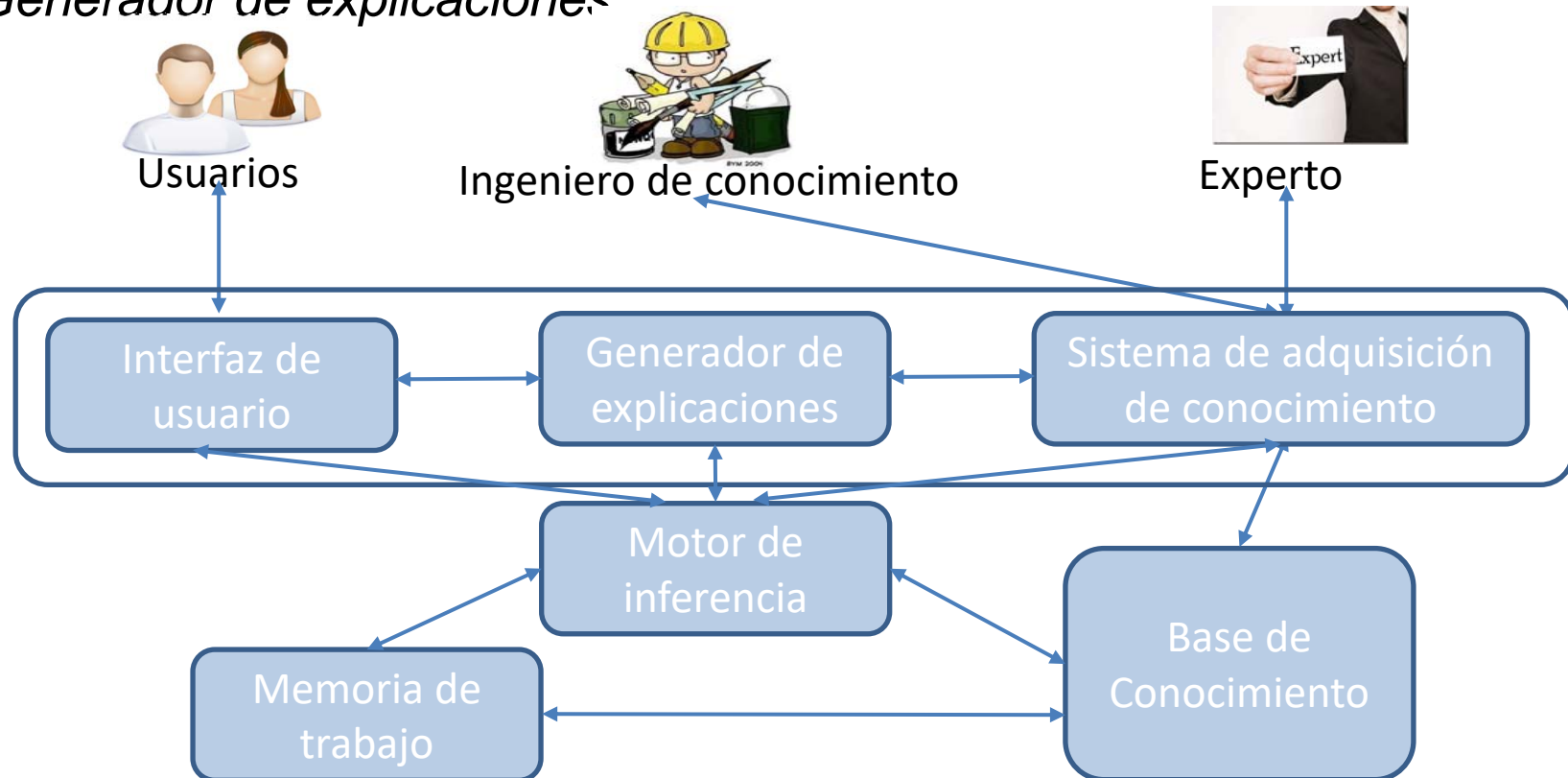
MYCIN Format

```
IF (AND (SAME CNTEXT GRAM GRAMNEG)
        (SAME CNTEXT MORPH ROD)
        (SAME CNTEXT AIR AEROBIC))
THEN (CONCLUDE CNTEXT CLASS ENTEROBACTERIACEAE
      TALLY .8)
```

Factor de certidumbre 0.8

SBC. Arquitectura general

- **Base de conocimiento (BC):** Conocimiento **estático** sobre el dominio
- **Memoria de trabajo (MT):** Conocimiento **dinámico** sobre la situación actual
- **Motor de inferencia**
- *Generador de explicaciones*



Ingeniería del conocimiento

- El proceso de construir una **BC inicial** junto con un motor de inferencia se llama **ingeniería del conocimiento**
- El ingeniero del conocimiento se encarga de:
 - Construir **la base de conocimiento**:
 - Investiga el dominio concreto y dialoga con el experto humano
 - Conceptualiza el dominio
 - Crea una representación formal de los objetos y relaciones del dominio
 - Implementación
 - Motor de inferencia → razona con el conocimiento adquirido

Reglas, casos, taxonomías
conceptuales, atributos, rangos,
terminología, restricciones,
dependencias, procedimientos,...

Propiedades del conocimiento

- Voluminoso
- Impreciso
- Inseguro
- **Heurístico (experto/experiencia)**
- Incompleto
- Evoluciona (puede cambiar)
- **Hay que adquirirlo y representarlo**

Conocimiento heurístico

- ¿Qué caracteriza el conocimiento experto? La experiencia, por ejemplo, un médico o un técnico
- La parte del conocimiento más difícil es el **conocimiento informal o atajos** que permiten a un experto alcanzar rápidamente la solución de un problema sin tener que llevar a cabo un análisis detallado del problema
- Parte del conocimiento en los sistemas basados en conocimiento
 - Es **conocimiento heurístico**, es decir, no representa un análisis en profundidad del problema, sino una solución “aceptable”
 - Es el conocimiento específico aplicable a cada situación específica del dominio → “sabiduría heurística”
 - **Relaciones de causa y efecto que surgen a partir de la experiencia del experto**
- Pueden ser reglas de actuación, el orden de aplicación de inferencias, **reglas causa-efecto, “recetas”, casos prototípicos, valoraciones de un estado,.....**
- *Un conductor descubre que el tubo de escape de su coche hace un ruido extraño estando parado en un semáforo. ... en una ocasión llevó el coche al taller y el mecánico le dijo que estaba mal conectado un tubo, y que entraba aire en el motor y producía esos ruidos. El mecánico le enseñó qué tubo era. Así que para el coche, abre el capó, mira el mismo tubo (que está otra vez suelto) y lo ajusta. Al arrancar de nuevo el tubo de escape ya no hace ruidos.*

Conocimiento incompleto

- En general es imposible representar todo el conocimiento de un dominio
- Causas:
 - Falta de conocimiento en la BC
 - Mecanismos de inferencia insuficientes
- ¿Cómo actúa el sistema frente a este hecho?:
 - ***Hipótesis de mundo cerrado*** →
Todo lo que no se puede obtener de la BC es **falso**
 - ***Hipótesis de mundo abierto*** →
Todo lo que no se puede obtener de la BC **no se sabe**

Hipótesis de mundo cerrado

- Los asertos verdaderos están incluidos en la BC o pueden ser derivados de ella
- **Todo lo que no se puede obtener de la BC es falso**
- La respuesta para preguntas de las que no tengo información es siempre FALSO
- **Se niega la incompletitud** → Esto puede ser engañoso porque el usuario puede pensar que el sistema tiene evidencias de que realmente la respuesta es FALSO
- Fácil de implementar y se utiliza mucho pero hay que ser consciente de que puede no ser válido para algunas situaciones

JUAN

24 años

Soltero

Médico

¿Juan es moreno? NO

JUAN

24 años

Soltero

Rubio

¿Juan es moreno? NO

Hipótesis de mundo abierto

- Todo lo que no se puede obtener de la BC **no se sabe**
- La respuesta para preguntas de las que no tengo información es siempre **NO SÉ**
- Distingue entre lo que es **falso** basado en evidencias de la BC y lo que **no se sabe**

JUAN

24 años

Soltero

Médico

¿Juan es moreno? NO SÉ

JUAN

24 años

Soltero

Médico

Rubio

¿Juan es moreno? NO

Conocimiento por omisión

- **Conocimiento general:** las aves vuelan
- Conocimiento que se asume implícitamente mientras no se niegue explícitamente
- Las **excepciones** se establecen a posteriori
- **Ejemplo:** mecanismos de herencia con excepciones
 - Por ejemplo, la propiedad poder volar es habitual en las aves, es una propiedad habitual pero no esencial.
 - Si un individuo x es un ave puedo inferir que vuela, pero si me dicen que el individuo x es un pingüino → diré que no vuela y hay que borrar todas las inferencias que se hayan hecho teniendo en cuenta que x volaba

Conocimiento por omisión

- Sistemas **monótonos**: Lo verdadero no puede dejar de serlo, no puedo retractarme de algo ya inferido
 - Son más fáciles de implementar pero más limitados
- Sistemas **no monótonos**:
 - Permiten usar conocimiento por defecto que puede modificarse después
 - Ricky es un pájaro → Ricky vuela;
 - Ricky es un pingüino → Ricky NO vuela;
 - ***Las conclusiones establecidas en un cierto momento pueden dejar de ser ciertas si llega nueva información***
 - Requieren garantizar la consistencia → TMS (*True Maintenance Systems*)
 - Permite saber qué inferencias se han hecho a partir de otras y cuáles hay que deshacer si algo deja de ser verdad.

Conocimiento inseguro

- El conocimiento añadido al sistema se acompaña de un factor de certeza que se usa en las inferencias
 - Por ejemplo: “Si el paciente presenta la piel amarilla entonces el diagnostico en el 70% de los casos es hepatitis y en el 30% de los casos ictericia”
 - Factor de certeza → nivel de seguridad

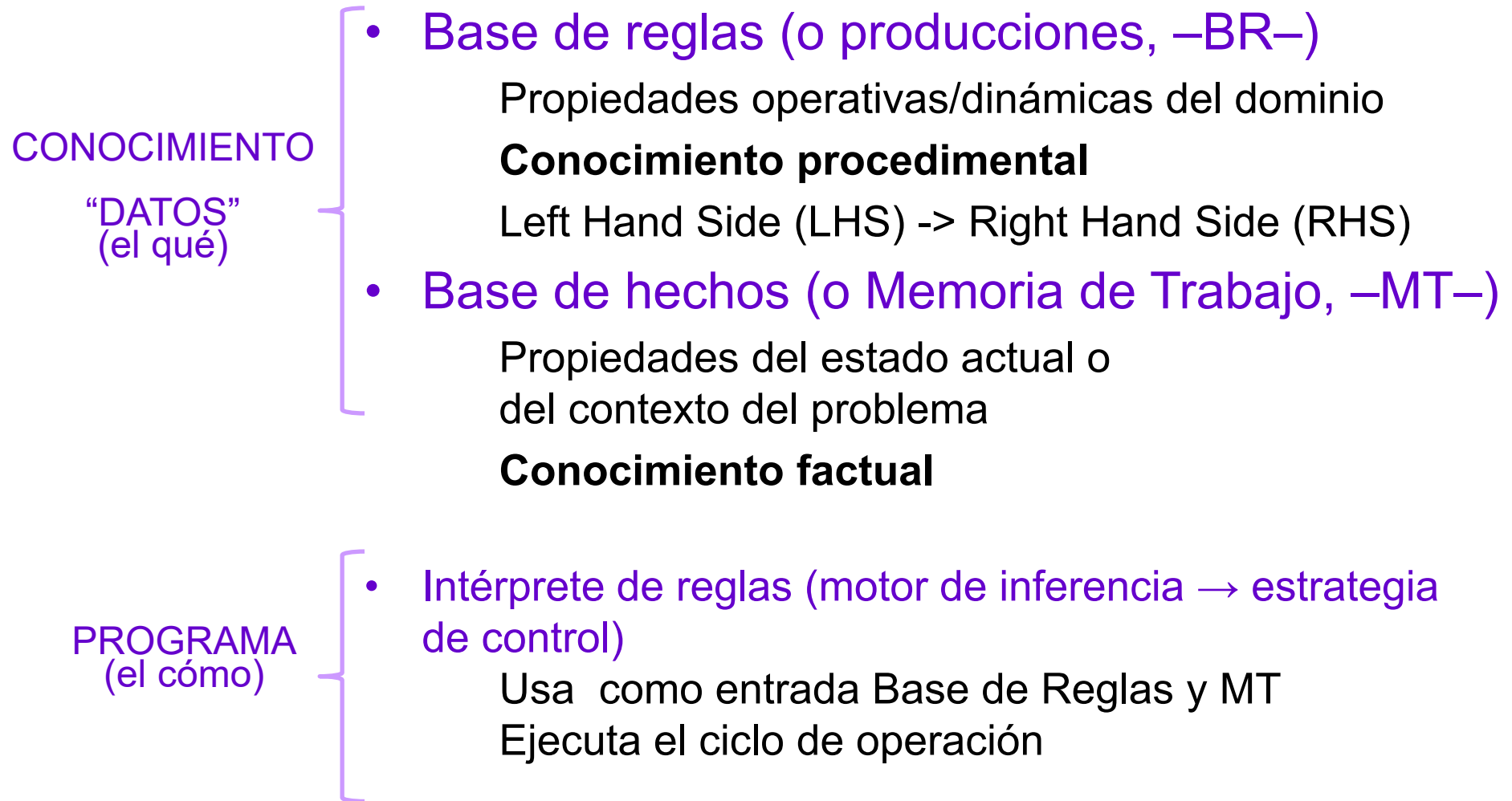
Conocimiento impreciso

- El conocimiento puede usar predicados o cuantificadores vagos (no precisos)
 - “Si la humedad es **alta**, la presión es baja y está muy nublado, entonces lloverá”
 - “Si eres **alto** puedes jugar en el equipo”
- **El factor de certeza es 1** pero es conocimiento **impreciso**
 - ¿cuánto es alta?, ¿cuánto es baja?, ¿cuánto es muy nublado?
- **Lógica difusa**, cuantificadores especiales...
 - una persona es alta si mide mas de 1.80mts, pero de igual forma se considera a una persona como alta si mide 1.7999mts
 - En lógica tradicional hay demarcaciones estrictas
 $A = \{ x \mid x > 1.8 \}$ Una persona que mide 1.799999mts es baja!

Técnicas de representación

- Representaciones basadas en relaciones
 - Lógica
 - Redes semánticas
 - Grafos de datos enlazados (linked data)
- Representaciones estructurada (basadas en objetos)
 - Marcos
 - Objeto-Atributo-Valor
- Representaciones basadas en acciones
 - **Sistemas de reglas (o de producción)**
 - Scripts (guiones)
- Todos los sistemas anteriores se pueden combinar o variar dando lugar a nuevas técnicas

Arquitectura de un sistema de Reglas



Base de reglas (BR)

- Conocimiento **procedimental** sobre cómo solucionar problemas
- Formato uniforme: **SI condiciones-LHS ENTONCES acciones-RHS**
 - **Condiciones** (**antecedente** de la regla): verificar en la MT (patrones)
 - **Acciones** (**consecuente** de la regla):
 - Cambios a realizar sobre la MT
 - Actividades externas al sistema (imprimir, órdenes dispositivos)
 - **No** son reglas del tipo *if-then-else*
- Las reglas **son independientes** unas de otras
 - Una regla **no** puede hacer referencia a otra (no están acopladas)
 - **La única comunicación entre reglas es a través de la MT**
 - Flexibilidad/modularidad versus ineficiencia
 - Una regla resuelve una parte del problema: un paso

Reglas

- Las reglas de producción tienen dos partes:
 - **Izquierda (LHS):** Patrón o condición que determina la aplicabilidad de la regla
 - **Derecha (RHS):** Descripción de la operación o acción a llevar a cabo al aplicar el operador (o disparar la regla)
- Ejemplo:

```
(defrule fallo2 "Fallo temperatura"  
  (temp ?x)  
  (temp-ext ?y)  
  (test (>= (abs (- ?y ?x)) 10))
```

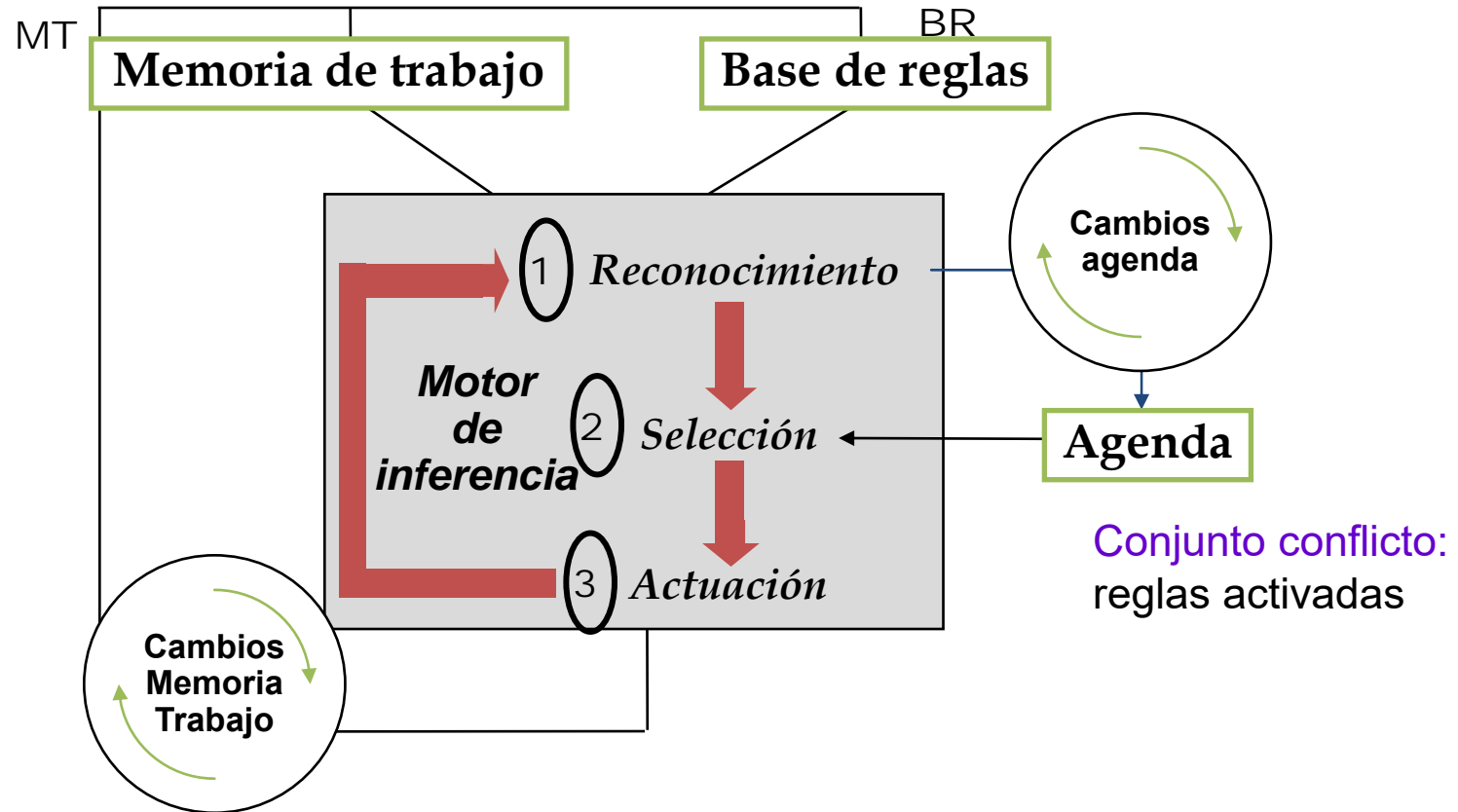
} LHS (left-hand side)

=>

```
(printout t "El frigo ha perdido gas" crlf) )
```

} RHS (right-hand side)

Diagrama de Funcionamiento



Motor de Inferencia: Ciclo de Operación

1. **Reconocimiento o equiparación** (*matching*): LHS
 - Qué reglas son aplicables: crea el **conjunto conflicto** o **agenda**
 2. **Selección o resolución de conflictos**: ¿qué regla se ejecuta?
 - Lo decide de acuerdo a la **estrategia de control**
 3. **Actuación**: se ejecuta la regla (ejecutar las acciones del RHS)
 - Añadir/quitar hechos de MT y operaciones externas al sistema de reglas
 4. **Repite el ciclo**: Inicializa la agenda y vuelve al punto 1.
 - Hasta que se alcanza el objetivo
-
- Se puede cambiar la **estrategia de control** que resuelve conflictos cuando hay varias reglas en la agenda
 - ¿Qué regla debo ejecutar de todas las de la agenda?
 - La estrategia se define al diseñar el sistema
 - Ej: **(set-strategy depth)**: selecciona la más recientemente activada

Ejemplo

- Si la edad del paciente es inferior a 10 años, y tiene manchas rojas y fiebre entonces tiene varicela

SI (Paciente ?p ?edad) and (?edad < 10) and
(Síntomas ?p fiebre) and (Síntomas ?p manchas-rojas)
ENTONCES **Añadir (Enfermedad ?p varicela)**

Variables que tomaran valor para un paciente concreto **?p ?edad**

El resto son símbolos constantes

Ejemplo: ordenación de cadenas

- Sistema que ordena las letras de una cadena de $\{a, b, c\}^*$
 - Estrategia de **resolución de conflictos**:
primera regla **según el orden de definición**
 - Conjunto de reglas
 - 1) $ba \rightarrow ab$
 - 2) $ca \rightarrow ac$
 - 3) $cb \rightarrow bc$

Iteración	MT	Agenda	R. disparada
0	cbaca	1(cb aca), 2(cb ac a), 3(cb aca)	1(cb aca)
1	cabca	2(cab ca), 2(cab ca)	2(cab ca)
2	acbca	2(cab ca), 3(ac bca)	2(cab ca)
3	acbac	1(ac ba c), 3(ac ba c)	1(ac ba c)
4	acabc	2(ac abc)	2(ac abc)
5	aacbc	3(aa cb c)	3(aa cb c)
6	aabcc	-----	Halt

Tipos de Razonamiento : Encadenamientos

- Encadenamiento progresivo (hacia adelante, forward chaining)
 - Dirigido por datos (antecedentes-LHS)
 - Desde los datos (hechos) conocidos busca la conclusión
 - Si los datos verifican las condiciones (LHS) de una regla,
 - Entonces la regla se puede aplicar/activar/disparar/ejecutar
- Encadenamiento regresivo (hacia atrás, backward chaining)
 - Dirigido por objetivos (consecuentes-conclusiones-RHS):
 1. Selecciona conclusiones (RHS) posibles:
 - De reglas cuyos RHS equiparen a la consulta
 2. Intenta probar su validez buscando evidencias (LHS) que la soporten:
 - Un antecedente-LHS es cierto si:
 - a) Sus hechos están en la MT del sistema
 - b) Si no, se busca si es consecuente (RHS) de alguna regla R:
 - Se prueban recursivamente los antecedentes de dicha regla R
 - c) Si no son aplicables a) y b), se asume la hipótesis del mundo cerrado (o se pregunta al usuario)

Palíndromos. Razonamiento progresivo y regresivo

- Las reglas pueden utilizarse de dos formas:

(R1) $\$ \rightarrow a\a

(R2) $\$ \rightarrow b\b

(R3) $\$ \rightarrow c\c

“\$” encaja con cualquier cadena

– Encadenamiento hacia adelante (razonamiento progresivo):

- Utilizar las reglas para **generar** palíndromos
- Dado un símbolo inicial como c , la secuencia de reglas $(R1)$, $(R2)$, $(R3)$, $(R2)$, $(R3)$ generará la siguiente secuencia de cadenas:

$aca \quad bacab \quad cbacabc \quad bcbacabcb \quad cbcbacabcbc$

– Encadenamiento hacia atrás (razonamiento regresivo):

- Utilizar las reglas para **reconocer** palíndromos
- Dado un palíndromo como $bacab$, podemos trazar la secuencia de reglas que llevan a su construcción
- $bacab$ satisface la parte derecha de $(R2)$, cuya parte izquierda da aca
- aca satisface la parte derecha de $(R1)$, cuya parte izquierda da c
 - Respuesta : Sí, es un palíndromo

Criterios de elección de dirección del encadenamiento

- Dado el problema, ir de menos a más:
 - Complejidad del LHS y del RHS de las reglas
 - Empezar por el menor número de hechos (inicial u objetivo)
 - Aspecto del espacio de estados
 - Naturaleza y disponibilidad de los datos del problema
 - Factor de ramificación en ambas direcciones
 - Atrás: Si hay muchos datos iniciales (hechos) y/o pocas posibles conclusiones
 - Adelante: Si hay pocos datos iniciales y/o muchas posibles conclusiones
- Si el proceso de resolución es interactivo
 - Utilizar el mismo que el experto usa de forma natural
- Sistemas de contestación de preguntas
 - Utilizar hacia atrás (*p.ej. sistemas de diagnóstico*)
- **Sistemas que reaccionan a datos que llegan periódicamente**
 - Utilizar hacia delante
 - Ej: datos de sensores recibidos cada minuto en una central térmica

Implementación: fase de reconocimiento

- El encaje uno a uno es muy ineficiente
- En cada ciclo los cambios producidos en la base de hechos son los únicos que influyen en la modificación del conjunto conflicto (*REdundancia TEmporal*)
 - Los nuevos hechos que son introducidos en la memoria de trabajo son los que determinan las nuevas reglas aplicables
 - Idea **del algoritmo RETE**: en lugar de buscar qué reglas satisfacen los hechos existentes en cada momento, son los nuevos hechos generados en cada ciclo los que buscan o determinan qué nuevas reglas se seleccionan

Estrategias de control o resolución de conflictos - I

- Estrategias de resolución del conjunto conflicto (agenda)
 - **Qué regla del conjunto conflicto será disparada en cada ciclo**
- Dos enfoques de estrategias
 - **Control global**: control independiente del dominio de aplicación
 - Estrategias implementadas en el intérprete (optimización)
 - No son modificables por el programador
 - **Control local**: control dependiente del dominio de aplicación
 - El programador da instrucciones explícitas: **establecer prioridades**
- Objetivos, que el sistema se comporte de forma
 - **Sensible**: que reaccione a cambios producidos en la memoria de trabajo
 - **Estable**: que haya continuidad en la línea de razonamiento
- **Principio de refracción**: una regla sólo se dispara una vez con los mismos hechos
- La estrategia ha de tener un equilibrio entre sistemática y eficiente

Estrategias de control o resolución de conflictos – I I

1. **Orden** en la base de reglas
 - Se selecciona la primera regla aplicable,
 - Asume orden lineal explícito en la BR (lo cual puede ser peligroso)
2. **Prioridades** asociadas a las reglas (*conocimiento heurístico*)
 - Se selecciona la regla de **mayor prioridad**
 - La prioridad la establece el experto del dominio - *control local*
3. **Especificidad**: más prioridad a las reglas con más condiciones
4. **Novedad**: prioridad de las reglas según
 - a) **Antigüedad (en MT) de sus hechos**
 - b) Momento de activación de la regla
 - (set-strategy depth)** : dispara la más recientemente activada
 - (set-strategy breadth)** : dispara la que lleva más tiempo activada
5. **Arbitrariedad**
 - Cuando no existen criterios adicionales disponibles (*no sistemática*)

Aprender reglas a partir de ejemplos

If Allergies = Yes and Smoker = Yes and Irregular-Heartbeat = Yes, then Asthma

If Allergies = Yes and Past-Respiratory-Illness = Yes and Avg-Body-Temperature ≥ 0.1 , then Asthma

If Smoker = Yes and BMI ≥ 0.2 and Age ≥ 60 , then Diabetes

If Family-Risk-Diabetes = Yes and BMI ≥ 0.4 and Frequency-Infections ≥ 0.2 , then Diabetes

If Frequency-Doctor-Visits ≥ 0.4 and Childhood-Obesity = Yes and Past-Respiratory-Illness = Yes, then Diabetes

If Family-Risk-Depression = Yes and Past-Depression = Yes and Gender = Female, then Depression

If BMI ≥ 0.3 and Insurance-Coverage = None and Avg-Blood-Pressure ≥ 0.2 , then Depression

If Past-Respiratory-Illness = Yes and Age > 50 and Smoker = Yes, then Lung Cancer

If Family-Risk-LungCancer = Yes and Avg-Body-Temperature ≥ 0.1 , then Lung Cancer

If Disposition-Tiredness = Yes and Frequency-Doctor-Visits ≥ 0.4 , then Lung Cancer

If Family-Risk-Leukemia = Yes and Frequency-Doctor-Visits ≥ 0.4 , then Leukemia

If Disposition-Tiredness = Yes and Frequency-Doctor-Visits ≥ 0.4 , then Leukemia

Is the person fit?

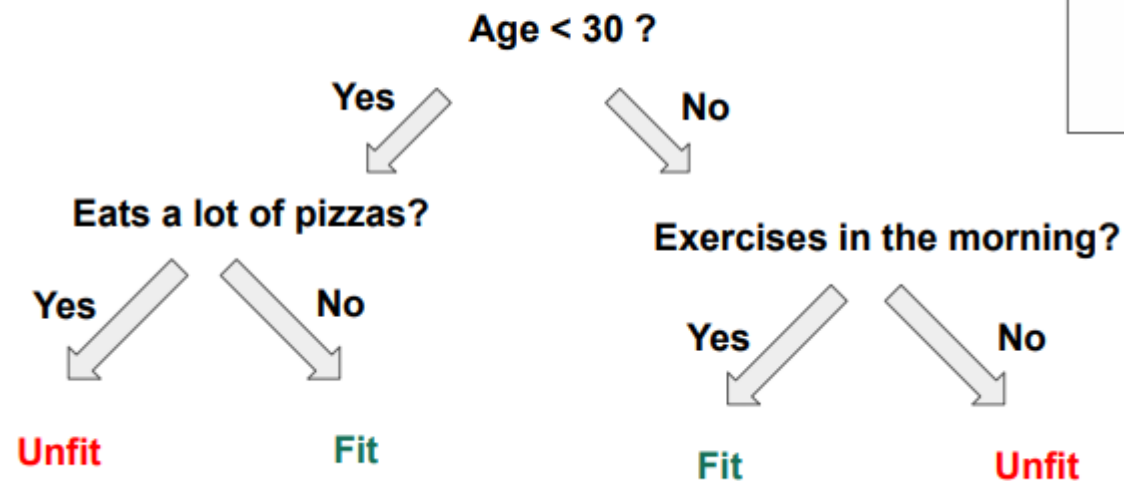


Figure credit: Interpretation
Leskovec

Ventajas de los Sistemas de Reglas

- Separación de conocimiento y control
 - Simplifica el desarrollo de sistemas expertos
- Carácter totalmente **modular** de la base de conocimiento
- Conocimiento representado con una sintaxis simple y homogénea
- Naturalidad para representar cierto tipo de conocimiento
- Control dirigido por patrones
 - Los programas en IA necesitan flexibilidad
 - Las reglas pueden “dispararse” en cualquier secuencia
 - Estado → reglas aplicables → camino a la solución
- Independencia del lenguaje
- Modelo plausible del mecanismo humano de resolución de problemas
 - Razonamiento basado en reglas: decidir qué hacer o qué se puede deducir cuando se dan una serie de condiciones
- Trazabilidad del proceso de razonamiento y explicaciones
 - Se puede comunicar qué condiciones se conocían en un momento determinado, por qué se ha escogido una regla y qué se ha deducido tras aplicarla