

### 5.1 Introducción

- Buscan capturar la experiencia humana para resolver problemas y tomar decisiones.
- Quieren capturar las heurísticas de razonamiento de los expertos humanos.
- Usan un conocimiento declarativo, con lógica de predicados restringida a cláusulas de Horn.
- Separa conocimiento (hechos y reglas) y mecanismos de inferencia (encadenamiento).

### 5.2 Componentes básicos de los SBR

- **Base de hechos:** BH Información actual del problema. Datos iniciales y los que se van obteniendo al aplicar reglas.

- **Base de conocimientos:** BC Conocimiento declarativo experto en forma de reglas que operan sobre la BH. Reglas: Condición  $\rightarrow$  acción.

Por ejemplo: *paciente\_infantil Y manchas Y fiebre  $\rightarrow$  varicela*

- **Motor de inferencias:** MI Es un algoritmo (independiente de BC y BH) que selecciona que regla dispara (equiparación, conjunto conflicto) y determina cómo lo hace (hacia delante, hacia atrás...)

### 5.3 Inferencia

Los SBR usan encadenamiento hacia delante y encadenamiento hacia atrás. O modos mixtos

#### Encadenamiento hacia delante:

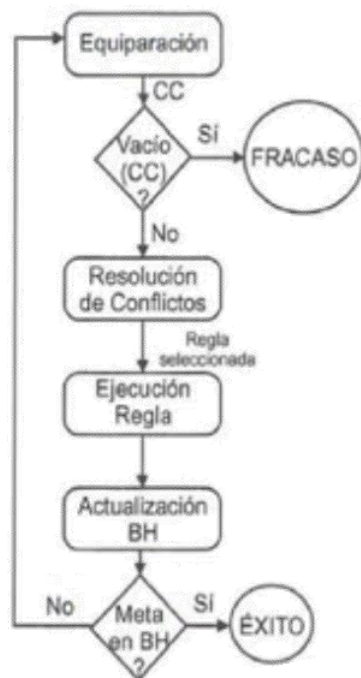
##### Ciclo de reconocimiento-acción:

**Paso 1:** Determinar a partir de la BH que reglas son aplicables (CC: **Conjunto conflicto**). Se usa algoritmo de **equiparación** (recorre toda la BC, ve los antecedentes de las reglas y se compara con la BH).

**Paso 2: Resolución de conflictos.** Se determina, de todas las instancias de reglas, cual es aplicable.

**Paso 3: Ejecución** de la regla (acciones indicadas en el consecuente o cabeza de la regla).

**Paso 4: Actualización** de la BH (regla que puede añadir o quitar hechos).



Por ejemplo: tenemos una BH con una serie de hechos:  $p, q, r, s$

Y vemos que reglas de la BC son aplicables, por ejemplo, de todas las reglas que tenemos, son aplicables estas dos, que forman el **conjunto conflicto**:

$$p, q \rightarrow r \quad p, s \rightarrow t$$

Por ejemplo: Tenemos una BC de un sistema basado en reglas con las siguientes reglas:

$$R1: h_2 \rightarrow h_4$$

$$R2: h_1 \rightarrow h_3$$

$$R3: h_3 \wedge h_8 \rightarrow h_5$$

$$R3: h_3 \wedge h_{10} \rightarrow h_5$$

$$R5: h_3 \wedge h_{11} \rightarrow h_2$$

$$R6: h_{10} \wedge h_{16} \rightarrow h_{13}$$

$$R7: h_{10} \rightarrow h_{14}$$

$$R8: h_2 \rightarrow h_{15}$$

$$R9: h_1 \rightarrow h_{13}$$

$$R10: h_4 \rightarrow h_7$$

$$R11: h_1 \wedge h_8 \rightarrow h_{10}$$

$$R12: h_{18} \rightarrow h_8$$

Donde cada  $h_i$  representa un hecho. Cada uno de ellos, se almacena en la **Base de afirmaciones** como  $h_i(t)$ , el hecho  $h_i$  fue inferido en el ciclo  $t$ . Inicialmente  $BA_0 = \{h_1, h_9, h_{11}\}$ .

- Si nuestro objetivo es obtener  $h_{13}$  indicar detalladamente cómo evoluciona la ejecución del método de encadenamiento hacia adelante, a partir de  $BA_0$ . ¿Es posible obtener  $h_{13}$ ?

¿Hasta qué ciclo llegaríamos?. Cómo mecanismo de control consideramos el criterio de refractariedad (no se puede ejecutar en el presen ciclo una regla que fue ejecutada en el ciclo anterior) y tienen preferencia las reglas de menor subíndice (la regla, no los hechos).

Buscamos en la BC las reglas que son aplicables a lo que tenemos en  $BA$ :  $h_1, h_9, h_{11}$ ,

solamente las reglas **R2** y **R9** tienen a alguna de esos hechos en el antecedente:

$$R2: h_1 \rightarrow h_3$$

$$R9: h_1 \rightarrow h_{13}$$

Estas dos reglas forman el conjunto conflicto, y ahora, para resolver el conflicto, el principio de refractariedad no aplica ya que no hay ciclos anteriores, así que, **R2** por tener menor subíndice. Aplicando **R2** se obtiene  $h_3$ . En el ciclo 1, entonces, tendríamos:

$$BA: h_1(0), h_9(0), h_{11}(0), h_3(1),$$

Pasamos al siguiente ciclo 2: El CC: **R2, R5** y **R9**.

Por el principio de refractariedad, no podemos ejecutar **R2**, entonces, por  $<$  subíndice: **R5**.

$$BA: h_1(0), h_9(0), h_{11}(0), h_3(1), h_2(2).$$

Ciclo 3: CC: **R1, R2, R5, R8** y **R9**. No pudiendo aplicar **R5**. Se aplica **R1**

$$BA: h_1(0), h_9(0), h_{11}(0), h_3(1), h_2(2), h_4(3).$$

Ciclo 4: CC: **R1, R2, R5, R8, R9** y **R10**, No pudiendo aplicar **R1**. Se aplica de nuevo **R2**

$$BA: h_1(0), h_9(0), h_{11}(0), h_3(4), h_2(2), h_4(3).$$

Ciclo 5: CC: **R1, R2, R5, R8, R9** y **R10**, No pudiendo aplicar **R2**. Se aplica de nuevo **R1**

Esto haría entrar en bucle, y nunca se podría obtener  $h_{13}$ ; esto indica que el mecanismo de control elegido es malo.

# Fundamentos de inteligencia artificial

## Encadenamiento hacia atrás:

**Ciclo de inferencias:** Se parte de una secuencia de **objetivos M** (inicialmente, la meta que se busca inferir). Mientras **M** no será vacía:

**Paso 1:** Equiparación. Se buscan reglas cuya cabeza corresponda con **m**, el primer elemento de **M**. Se crea así el **CC**. Si **m** aparece en el **BH** se trata como regla con cuerpo vacío.

**Paso 2:** Resolución de conflictos. Se selecciona una regla del **CC**

**Paso 3:** Ejecución, se reemplaza en **M** el objetivo **m** por el cuerpo de la regla seleccionada. Este cuerpo puede ser vacío (hechos) o tener uno o varios elementos (reglas).

**Por ejemplo:** En el enunciado del ejemplo anterior:

- Si la **BA** = { $h_1, h_{11}$ } ver con encadenamiento hacia atrás si en algún momento se puede llegar a tener  $h_7$ .

Cómo nuestro objetivo es  $h_7$  buscamos una regla cuyo consecuente sea  $h_7$ , en este caso, **R10**. Obtenemos entonces cómo subobjetivo  $h_4$ . Que no está en **BA**, por lo que, por la regla **R1**, tendremos como nuevo subobjetivo el hecho  $h_2$ .

Que lo conseguimos con **R5**, y sus nuevos subobjetivos  $h_3$  y  $h_{11}$ ; cómo el segundo ya está en **BC**, el nuevo subobjetivo será  $h_3$ .

Y lo podemos encontrar en **R2**, cómo tenemos  $h_1$  en la **BA** inicial, en efecto podemos obtener  $h_7$  con la **BA** inicial y aplicando encadenamiento hacia atrás.

El encadenamiento hacia atrás tiene ventajas.

- Se limita el número de equiparaciones (la búsqueda va orientada a los consecuentes que se quieren probar)
  - Disminuye el tamaño del árbol de búsqueda.
- Pero puede llevar a bucles infinitos.

Con muchas reglas con **muchas condiciones en el antecedente**, mejor usar **encadenamiento hacia adelante**. Si hay **muchas reglas cuyo consecuente forma parte del antecedente** de otras reglas, **mejor hacia atrás**.

De forma general, si no sabemos las metas a alcanzar, mejor hacia adelante, y si las conocemos, hacia atrás.

## Encadenamiento mixto:

- Habría dos procesos de búsqueda simultáneo (hacia adelante y hacia atrás), hasta que ambos se encuentren (cómo la búsqueda bidireccional).
- Es conveniente si el espacio de búsqueda creciese exponencialmente en cada paso.
- Algunas reglas solo serán aplicables en una dirección.

## 5.4 Técnicas de equiparación

### Algoritmo RETE:

- Evita examinar todas las reglas de la BC con todos los datos de la BH
- Usa un **compilador** que **busca patrones** comunes en los antecedentes para evitar evaluarlos repetidamente.
- Crea una red RETE: Que es un árbol con la clasificación estructurada de los antecedentes
- Se guarda el resultado de la equiparación en cada ciclo para continuar en el siguiente, en vez de comenzar de nuevo.
- Es muy usado en SBR como CLIPS, por su gran eficiencia en el tiempo.
- El consumo de memoria se dispara (Problema en sistema expertos muy grandes).

## 5.5 Técnicas de resolución de conflictos

1. Se selecciona la **primera regla** que se equipara con los contenidos de la BH
2. Se elige la regla con **prioridad** más alta.
3. Se seleccionan las instanciaciones más **específicas**
4. Seleccionar **arbitrariamente** una regla del conjunto conflicto
5. Seleccionamos las instanciaciones con elementos más **recientemente** añadidos a la BH
6. Seleccionar una instancia **no ejecutada previamente** (refracción, que puede ser en un número determinado de ciclos).

## 5.6 Ventajas e inconvenientes

- Los SBR son muy modulares ya que son mantenibles si su tamaño no es muy grande.
- En sistemas grandes hay que estructurar las BC para facilitar depuración y evitar eventos colaterales.
- Al representar declarativamente el conocimiento, facilita incorporación de facilidades de auto explicación.
- Las herramientas de desarrollo de SBR integran otras técnicas convencionales.
- Los SBR tienen menor potencia expresiva que la lógica de predicado (para mejorar la eficiencia computacional), pero mejora la lógica proposicional.
- Es idónea para campos que puedan modelarse en un conjunto de múltiples estados, y que se pueda separar conocimiento y la forma en que se utiliza.