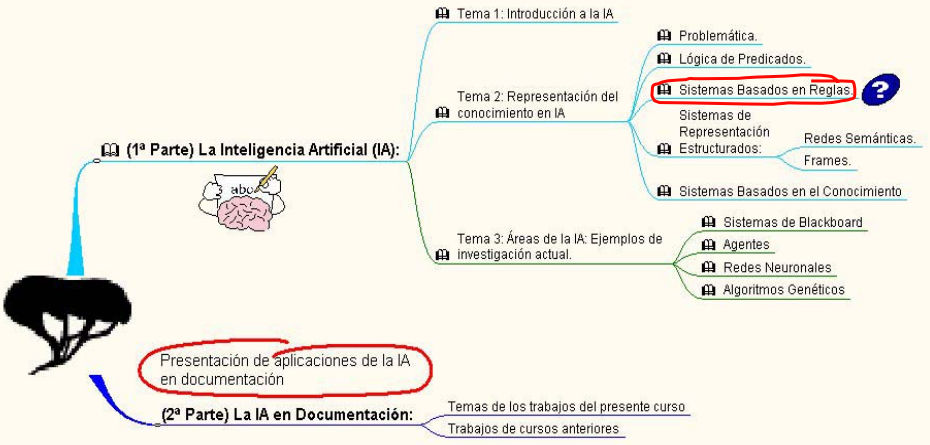
Tema 2 (II): Representación del Conocimiento en IA: Sistemas Basados en Reglas



Tema 2 (II): Representación del Conocimiento en IA: Sistemas Basados en Reglas

2. 1. Introducción.
2. 2. Representación Basada en la Lógica.
2. 3. Representación Basada en Reglas.
- 2. 3. 1. Introducción.
- 2. 3. 2. Repres. del Conocim. en SBR.
  - 2. 3. 2. 1. Hechos: Variables-Valor.
  - 2. 3. 2. 2. Reglas de Producción.
  - 2. 3. 2. 3. Hechos: Tuplas Objeto-Atributo-Valor.
- 2. 3. 3. Relación entre Reglas de Producción y la Lógica de Predicados de Primer Orden.
- 2. 3. 4. Inferencia en Sistemas Basados en Reglas.
  - 2. 3. 4. 1. Encadenamiento Inferencial.
  - 2. 3. 4. 2. Control Inferencial.
- 2. 3. 5. Patrones en Sistemas Basados en Reglas.

2. 4. Sistemas de Representación Estructurados.

- Bibliografía Básica:
  - **[Lucas 1991] Peter Lucas and Linda Van Der Gaag, "Principles of Expert Systems". Addison-Wesley. (1991). Capítulo 3.**
- Bibliografía Complementaria:
  - **[Rich 1994] E. Rich, K. Knight, "Inteligencia Artificial", McGraw Hill (1994). Capítulo 6.**
  - **[Charniak 1985] E. Charniak, D. McDermott, "Introduction to Artificial Intelligence", Addison-Wesley (1985). Capítulo 7.**
  - **[Russel 1996] S. Russel, "Inteligencia Artificial: Un enfoque moderno", Prentice Hall (1996). Capítulo 10.**
  - **[Negnevitsky 2002] M. Negnevitsky, "Artificial Intelligence: A Guide to Intelligent Systems", Addison-Wesley (2002). Capítulo 2.**

2.3.1. Introducción (I)

- **Contenido lógico de una fbf:**
  - **Independiente de la expresión en el calculo de predicados.**
- **Conocimiento experto:**
  - **Información extra-lógica o heurística implícita sobre cómo usar dicho conocimiento de forma optima.**
- **Razonamiento humano:**
  - **No es en base a 'cláusulas', sino en base a 'reglas'.**
- **Ejemplo:**

SI x,y >0 ENTONCES x*y>0

$$\forall x\, \forall y\, [>(x,0) \wedge >(y,0)] \rightarrow >(POR(x,y),0)$$

$$\equiv$$

$$\forall x\, \forall y\, [>(x,0) \wedge \neg >(POR(x,y),0)] \rightarrow \neg >(y,0)$$

2.3.1. Introducción (II)

- La implicación expresa cómo obtener una información:

  $\forall x$ vertebrado(x) $\rightarrow$ animal(x)

  $\forall x \forall y$ [trabaja-en(dep-ventas,x) ^ Edad(x,y) ^ >(y,30)] $\rightarrow$ casado(x)

  $\forall x$ [cilindro(x) ^ rojo(x)] $\rightarrow \exists y$ cubo(y)^sobre(y,x)

- Forma clausal :
  - ✓ Forma más uniforme (proceso de demostración más eficiente)
  - ✗ Pérdida de información de control

$A \vee B \vee C \equiv$

$\neg A \wedge \neg B \rightarrow C$

$\neg A \wedge \neg C \rightarrow B$

$\neg B \wedge \neg C \rightarrow A$

$\neg A \rightarrow (B \vee C)$

$\neg B \rightarrow (A \vee C)$

$\neg C \rightarrow (A \vee B)$

Cada implicación lleva su propia información heurística de control

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

- **Base de Hechos (BH)** – Información Factual

  Representación de Hechos del Dominio/Problema

  - Hechos no estructurados:
    - Listas, Strings, etc.
  - Conjunto de pares: Variable-Valor
  - Conjunto de tuplas Objeto – Atributo – Valor
  - Patrones
  - Otras representaciones:
    - Declarativas (lógica), Estructurales (redes, frames), . . .

- **Base de I**

- **Control** – Motor de Inferencia

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

- **Base de Hechos (BH)** – Información Factual
  - Representación de Hechos del Dominio/Problema
  - Declaración del Dominio de la BH: ONTOLOGÍA
- **Base de Reglas (BR)** – Conocimiento Normativo
  - Específico a un dominio de problemas
  - Reglas de Producción
- **Control** – Motor de Inferencia
  - Representación conocimiento general de resolución de problemas

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.1. Hechos: Variable – Valor

- **Hecho: Var. cuyo valor es una cte. o cjto. de ctes.**
  - Declaración del Dominio de la BH: ONTOLOGÍA
    - Conjunto de todas las variables, junto con sus posibles valores.
    - Descripción de la información que es relevante en el dominio modelado por el sistema.
  - Información del problema
    - Conjunto de pares varoable – valor.
- **Tipos de Variables**
  - Variables uni – valor
    - Información que es única. Sexo, Edad, Temperatura, ...
  - Variables multi – valor
    - Valor puede ser un cjto. de ctes. Enfermedad, Síntomas, ...

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.1. Hechos: Variable – Valor

- **Declaración del Dominio de una BH: ONTOLOGÍA (I)**
  - **Declaración de Variables: Puede ser:**
    - **No – Tipificada: Sólo se define el nombre**
    - **Tipificada:**
      - **Variable mono-valor (x^s) = $x^s : \tau$**
      - **Variable multi-valor (x^m) = $x^m : 2^\tau$**
        - **τ: Tipo que denota un cjto. no vacío de ctes. (enteros, reales, {blanco, rojo, azul})**
  - **Declaración del Dominio de la BH**
    - **Conjunto D de declaraciones de variables (restringe el valor que una variable puede tomar).**

      D = { sexo: {mujer, hombre}, edad: int, nombre: string,
      dolencia: $2^{\{\text{fiebre, dolor-abdominal, dolor-cabeza}\}}$,
      trastorno: $2^{\{\text{aneurisma-aortico, estenosis-arterial}\}}$ }

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.1. Hechos: Variable – Valor

- **Declaración del Dominio de una BH: ONTOLOGÍA (II)**
  - **Hecho: Una variable junto con el valor(es) que adopta:**
    - **$x^s = c$, donde $c \in t$. (x^s es declarada como $x^s:t$)**
    - **$x^m = C$, donde $C \subseteq t$. (x^m es declarada como $x^m:2^t$)**
  - **Conjunto de Hechos**

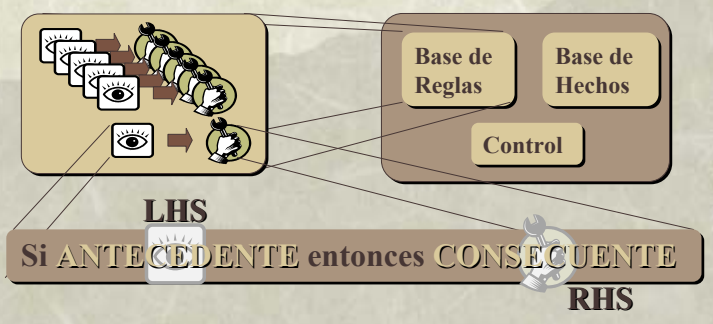
    $\{ x_1^s = c_1, \ldots\ldots, x_p^s = c_p, x_1^m = C_1, \ldots\ldots, x_q^m = C_q \}$
    donde c_i son constantes y C_j conjuntos de constantes.

    - **Una var. puede ocurrir una sola vez en un cjto. de hechos.**

{ sexo = hombre, edad = 27,
dolencia = { fiebre, dolor – abdominal },
enfermedad = { aneurisma – aórtico }
}

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción

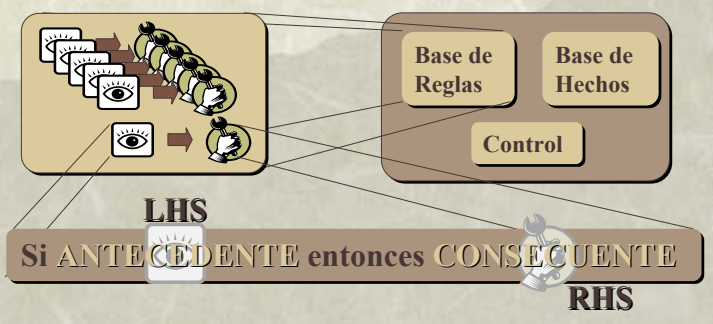


- **Representan:**
  - Conocimiento de resolución de problemas de un dominio específico (ejemplo: reglas físicas)
  - Conocimiento heurístico o reglas de experiencia (analogías informales en la vida real).

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción



if el paciente sufre dolor abdominal, y
un murmullo abdominal es percibido por auscultación, y
una masa pulsante es palpada en el abdomen
then el paciente padece una aneurisma aórtico

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción

- **Antecedente (LHS):** Condiciones sobre la BH
  - ¿$x_i^s = c$?, donde $c \in \tau$. $(x^s : \tau)$

```
( test ( = ?a 7 ) )
```

  - ¿$x_j^m = C$?, donde $C \subseteq \tau$. $(x^m : 2^\tau)$

```
( test ( member ?enfermedad gripe ) )
```

- **Consecuente (RHS):**
  - Asertar o eliminar hechos:
    - Añadir ($x_i^s = c \mid x_j^m = C$)

```
( bind ?a 6 ) )
```

```
( bind ?enfermedad ( insert ?enfermedad faringitis ) )
```

    - Eliminar ($x_i^s = c \mid x_j^m = C$)

```
( bind ?a 7 )
```

```
( bind ?enfermedad ( delete ?enfermedad faringitis ) )
```

  - Acciones externas

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción

- **Sintaxis**
  - <name> :: = string
  - <conditional-element> ::= <and-CE> | <or-CE> | <not-CE>| <test-CE>| <ordered-pattern> | <assigned-pattern>
  - <and-CE> ::= (and <conditional-element>$^+$)
  - <or-CE> ::= (or <conditional-element>$^+$)
  - <not-CE> ::= (not <conditional-element>)
  - <test-CE> ::= (test (<function-call> <term> <term>))
  - <function-call> ::= = | > | >= | < | <= | <> | eq | neq | <member>
  - <term> ::= <variable> | <constant>
  - <variable> ::= ? <name>
  - <constant> ::= <number> | <name>
  - <member> ::= member <variable> <term>
  - <assigned-pattern> ::= <variable> <- <ordered-pattern>
  - <ordered-pattern> ::= (<name> <constraint>*)
  - <constraint> ::= $ <variable> | <term>

```
( defrule <name>
    <conditional-element>+
    = >
    <action>*
)
```

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción

- **Sintaxis**
  - <action> ::= (bind <variable> <term>) | (<action-multi-value>) | (retract <term>) | (assert <ordered-pattern>)
  - <action-multi-value> ::= (create | <replace> | <action-name> <variable> <term>$^+$)
  - <action-name> ::= insert | delete
  - <replace> ::= replace <variable> <term> <term>$^+$

```
( defrule <name>
    <conditional-element>+
    =>
    <action>*
)
```

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción

- **Antecedente**
  - **Predicados** (se evalúan a Verdadero o Falso)
    - Variables uni-valuadas

```
( test ( = ?a 7 ) )
( test ( eq ?a casa ) )
( test ( neq ?a casa ) )  =  ( not ( test ( eq ?a casa ) ) )
( or ( test ( >= ?a 4 ) )  ( test ( = ?a 3 ) ) )
( and ( test ( >= ?a 7 ) ) ( test ( < ?a 20 ) )  ( test ( = ?b 9 ) ) )  =
    ( and ( and ( test ( >= ?a 7 ) ) ( test ( < ?a 20 ) ) ) ( test ( = ?b 9 ) ) )
```

    - Variables multi-valuadas

(member <variable> <term>)
<variable>: var. multivaluada
<term> : elemento

```
( test ( member ?lista 3 ) )
( and ( test ( member ?lista 3 ) ) ( not ( test ( member ?lista 7 ) ) ) )
```

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción

- **Consecuente**
  - **Asignación de Valores**
    - Variables uni-valuadas

```
( bind ?a 3 )
( bind ?a casa )
( bind ?b ?a )
```

    - Variables multi-valuadas

```
( bind ?a (create) )
( bind ?a (create ( 1 2 )) )
( bind ?a (insert ?a 4 5 6 7 ) )
( bind ?a (insert ?a casa agua ) )
```

?a

```
( )
( 1 2 )
( 4 5 6 7 1 2 )
( casa agua 4 5 6 7 1 2 )
```

(create): crea una variable multicampo vacía.
(insert <variable> <term>+)
<variable>: var. multicampo donde se quiere insertar un valor
<term>$^{+}$: uno o más elementos, pudiendo ser éstos vars. o ctes.

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción

- **Consecuente**
  - **Modificación de Valores**
    - Variables uni-valuadas

```
( bind ?a 1 )
```

    - Variables multi-valuadas

```
( bind ?a (replace ?a 4 10 ) )
( bind ?a (replace ?a casa hotel apartamento ) )
```

?a

```
( casa agua 10 5 6 7 1 2 )
( hotel apartamento agua 10 5 6 7 1 2 )
```

(replace <variable> <term> <term>+)
<variable>: var. multicampo donde se quiere reemplazar un valor
<term>: elemento que se quiere reemplazar
<term>$^{+}$: elemento o elementos por los que se va a sustituir el elemento a reemplazar.

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción

- **Consecuente**
  - **Eliminación de Valores**
    - Variables uni-valuadas

```
( bind ?a 1 )
```

    - Variables multi-valuadas

?a

```
( bind ?a (delete ?a 10 5 6 ) )
```

```
( hotel apartamento agua 7 1 2 )
```

(delete <variable> <term>+)
<variable>: var. multicampo donde se quiere borrar un valor
<term>$^{+}$: se puede borrar uno o más elementos de la var. multicampo (vars. o ctes.).

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción

- **Ejemplos (I):**
  - **If** el paciente sufre dolor abdominal, y
    un murmullo abdominal es percibido por auscultación, y
    una masa pulsante es palpada en el abdomen
    then el paciente padece una aneurisma aórtico

```
( defrule aneurisma
        ( and ( test ( eq ?dolencia dolor-abdominal ) )
              ( test ( eq ?auscultación murmullo-abdominal ) )
              ( test ( eq ?palpacion masa-pulsante ) )
        )
        =>
        ( bind ?enfermedad
                ( insert ?enfermedad aneurisma-aórtico ) )
)
```

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción

- **Ejemplos (II):**
  - **If** el paciente experimenta un dolor en la pantorrilla cuando anda, que desaparece gradualmente en reposo
    then una estenosis de una de las arterias de la pierna, posiblemente debido a una arteriosclerosis, es concebible

```
( defrule estenosis
        ( and ( test ( eq ?dolencia dolor-de-pantorrilla ) )
              ( test ( eq ?presencia andando ) )
              ( test ( eq ?ausencia en-reposo ) )
        )
        =>
        ( bind ?causa estenosis-arterial )
        ( bind ?enfermedad ( insert ?enfermedad arteriosclerosis ) )
)
```

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción

- **Ejemplos (III):**
  - **If** la presión asistólica excede los 140 mmHg, y
    la presión del pulso excede los 50 mmHg, y
    es percibido un murmullo diastólico, o
    es observado un corazón agrandado
    then el paciente sufre regurgitación aórtica

```
( defrule regurgitación
        ( and ( test ( > ?presión-asistólica 140 ) )
              ( test ( > ?presión-puslo 50 ) )
              ( or ( test ( eq ?auscultación murmullo-diastólico ) )
                   ( test ( eq ?percusión corazón-agrandado ) ) )
        )
        =>
        ( bind ?enfermedad ( insert ?enfermedad regurgitación-aórtica ) )
)
```

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.2. Reglas de Producción

Predicado o Acción	vars. uni-valor	vars. multi-valor
(test (eq ?x c))	$x^s = c$	----------
(test (member ?x c))	------	$c \in x^m$
(test (not (eq ?x c))	$x^s \neq c$	----------
(test (neq ?x c))	$x^s \neq c$	----------
(test (not (member ?x c))	-------	$c \notin x^m$
(test (< ?x c))	$x^s < c$	----------
(test (> ?x c))	$x^s > c$	----------
(bind ?x c)	$x^s \leftarrow c$	
(bind ?x (insert ?x c)	-------	$x^m \leftarrow \{c\} \cup x^m$
(bind ?x (delete ?x c)	-------	si $x^m = \{c\}$ entonces $x^m \leftarrow$ vacía sino $x^m \leftarrow x^m - \{c\}$
(bind ?x (replace ?x c d))	-------	$x^m \leftarrow (x^m \cup \{d\}) - \{c\}$
(bind ?x (create))	-------	$x^m \leftarrow$ vacía

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.3. Hechos: Objeto – Atributo – Valor

- **Hecho: Var. – Valor**
  - **No permite 'relacionar' variables entre sí.**
- **Al modelar dominios de problemas, es frecuente establecer vars. (atribs.) que caracterizan un mismo objeto, y subdominios (objetos/subobjetos) inter-relacionados.**
- **Frecuentemente se usan objetos para agrupar las propiedades mencionadas en las reglas heurísticas o de experiencia.**

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.3. Hechos: Objeto – Atributo – Valor

- **Objetos:**
  - **Estruct. de inform. que modela un subdominio/concepto.**
  - **Obj.: descrito por props. o atribs. específicos de tal obj.**

objeto dolor

atributos localización, carácter

  - **Tupla objeto – atributo – valor (o – a – v):**
    - **Permite expresar hechos asociados a atribs. de un obj.**
    - **Par ob obj. – atributo se comporta igual que las vars.**
    - **Atributos de tipo uni-valor ($o.a^s$) y multi-valor ($o.a^m$)**
    - **Distintas tuplas cuyo 1er elemento sea el mismo obj. denotan distintas características de un mismo obj.**
    - **Un par obj. – atrib. indica que el atrib. pertenece al obj.**

(dolor localización vientre)

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.3. Hechos: Objeto – Atributo – Valor

- **Ejemplos:**
  - **Conocimiento ➡ Objetos**

Conceptos/características:	atributo / valor
Relaciones:	atributo/valor

Concepto mesa:

Forma (rectangular, circular, etc.)

Tamaño (grande, mediano, etc.)

Nº patas (3, 4, 6, 12, etc.)

Color(blanca, negra, etc.)

➡

BH

(mesa forma rectangular)

(mesa tamaño grande)

(mesa número-patas 4)

(mesa color negra)

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.3. Hechos: Objeto – Atributo – Valor

- **Antecedente Reglas de Producción**

```
<conditional-element> ::= <and-CE> | <or-CE> | <not-CE>| <test-CE> |
                          <ordered-pattern> | <assigned-pattern>
    <assigned-pattern> ::= <variable> < –  <ordered-pattern>
    <ordered-pattern> ::= ( <name> <constraint>* )
    <constraint> ::= <term>
    <term> ::= <constant> | <variable>     (variable sólo en el elemento valor)
```

```
( and ( mesa forma rectangular )
        ( mesa color rojo ) )
        ( sangre presion-sistolica 140 )
        ?y < – ( sangre presion-pulso ?pp )
        ( test ( < ?pp 50 ) )
        ( or ( paciente auscultacion murmullo-diastolico )
                ?z  < – ( paciente percusion corazon-agrandado ) ) )
```

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.3. Hechos: Objeto – Atributo – Valor

- **Consecuente Reglas de Producción**

```
<action> ::= (bind <variable> <term> | <action-multi-value>) | (retract <term>) |
                (assert <ordered-pattern>)
    <constraint> ::= <term>
    <term> ::= <variable> | <constant>
```

– **Creación de Valores**

```
( bind ?a  rectangular )                          ;;; vble. uni-valuada
( assert ( mesa forma ?a ) )
( bind ?lista ( create ( forma rectangular ) ) )  ;;; vble. Multivaluada
(assert (mesa ?lista))
```

– **Eliminación de Valores**

```
                ?z < – ( paciente percusion corazon-agrandado )
( retract ?z )
( retract 6 )          ;;; 6 es el índice del hecho que se quiere eliminar
( bind ?a 6 )          ;;; se puede asignar a una vble. el índice
( retract ?a )
```

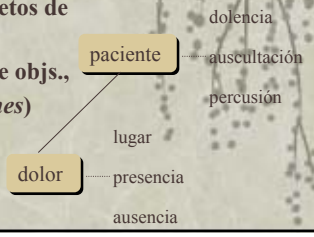
2.3.2. Representación del Conocimiento en Sistemas Basados en Reglas

2.3.2.3. Hechos: Objeto – Atributo – Valor

- **Ejemplo:**

```
( defrule estenosis
        ( paciente dolencia dolor-de-pantorrilla )
        ( dolor presencia andando )
        ( dolor ausencia en-reposo )
        =>
        ( assert ( dolor causa estenosis-arterial ) )
        ( assert (paciente enfermedad regurgitación-aortica ) ) )
```

- **Limitaciones del paradigma obj. – atrib. – valor**
  - No permite expresar relaciones explícitas entre objetos de una forma natural (redes semánticas / *frames*).
  - Un esquema de objs. define las interrelaciones entre objs., y entre objs. y sus atribs. (redes semánticas / *frames*)



2.3.3. Relación entre Reglas de Producción y Lógica de Predicados de Primer Orden

- **En general, las RP pueden ser traducidas a la lógica de predicados (≈ implicaciones).**
- **Semántica de un SBR se describe mediante una semántica procedural, como un método de inferencia específico de aplicación de las RP.**
- **Esta relación permite que una BR sea desarrollada sin necesidad de un conocim. preciso sobre el funcionamiento de su método de inferencia, basado en sus formas lógicas equivalentes.**
- **Ventajas de un razonamiento basado en reglas:**
  - Base formal en el cálculo de predicados.
  - Permite una BH más clara y estructurada:
    - Objeto – Atributo – Valor, redes, *frames*

2.3.3. Relación entre Reglas de Producción y Lógica de Predicados de Primer Orden

- **Relación entre predicados de RP y fbf**

Pred. o Acción	Repr. Lóg. de Atrib. Uni-	Repr. Lóg. de Atrib. Multi-valor
(o a v)	a(o) = v	a(o,v)
(o a ?x) (test (neq ?x v))	¬ a(o) = v	¬ a(o,v)
(not (o a v))	¬ a(o) = v	¬ a(o,v)
(o a ?x) (test (eq ?x v))	a(o) = v	--
(o a ?x) (test (< ?x v))	(o) < v	--
(o a ?x) (test (> ?x v))	a(o) > v	--
(assert (o a v))	a(o) = v	a(o,v)
(retract <term>)	--	--

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.3. Relación entre Reglas de Producción y Lógica de Predicados de Primer Orden

- **Relación entre predicados de RP y fbf**

(defrule nombre
(or $c_{1,1}$ $c_{1,2}$ ..... $c_{1,m}$)
•
•
•
(or $c_{b,1}$ $c_{n,2}$ ..... $c_{n,p}$)
=>
a_1 a_2 ..... a_q)

$$((c'_{1,1} \vee c'_{1,2} \vee \ldots\ldots \vee c'_{1,m}) \wedge$$
$$\vdots$$
$$((c'_{n,1} \vee c'_{n,2} \vee \ldots\ldots \vee c'_{n,p}))$$
$$\rightarrow$$
$$(a'_1 \wedge a'_2 \wedge \ldots\ldots\ldots \wedge a'_q)$$

2.3.3. Relación entre Reglas de Producción y Lógica de Predicados de Primer Orden

- **Ejemplo**

```
( defrule regurgitación
        ( sangre  presión-asistólica ?ps )
        ( test ( > ?ps 140 ) )
        ( sangre  presión-pulso  ?pp )
        ( test ( > ?pp 50 ) )
        ( or  ( paciente  auscultación  murmullo-diastólico )
              ( paciente  percusión  corazón-agrandado ) )
        =>
        ( assert ( paciente  enfermedad  regurgitación-aórtica ) ) )
```

(presión-asistólica(sangre) > 140 ^ presión-pulso(sangre) > 50 ^
(auscultación(paciente, murmullo-diastólico) v
percusión(paciente, corazón-agrandado)))
→
enfermedad(paciente, regurgitación-aórtica)

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.3. Relación entre Reglas de Producción y Lógica de Predicados de Primer Orden

- **Hechos vs. fbf**
  - Atributos uni-valor
    - **Predicado de igualdad de la lógica de primer orden**
      - **(o a^s v) es traducido en la cláusula unidad a(o) =v**
      - **Los axiomas de igualdad garantizan la unicidad de los atributos monovalor,**

        a(o) = v y a(o) = w / v ≠ w inconsistencia
  - Atributos multi-valor
    - **(o a^m v) es traducido en un cjto. de cláusulas unidad $a(o,v_i)$:**
      - **Una para cada v_i del cjto. de valores ctes. adoptado por el atributo: $a(o,v_i)$**

        a(o, v) y ¬ a(o, v) inconsistentes
      - **Una por cada una de las restantes ctes. v_i que ocurren en el tipo de atributo (negación por ausencia): $a(o,v_i)$**
        - ***Se asume que los valores de atributos multi-valor no introducidos explícitamente se consideran como no verdaderos (suposición del mundo cerrado)***

2.3.3. Relación entre Reglas de Producción y Lógica de Predicados de Primer Orden

- **Ejemplo:** D = { enfermedad^m:2$^{\{ \text{regurgitación-aortica, arteriosclerosis} \}}$, edad^s: int }
  - **Hechos**

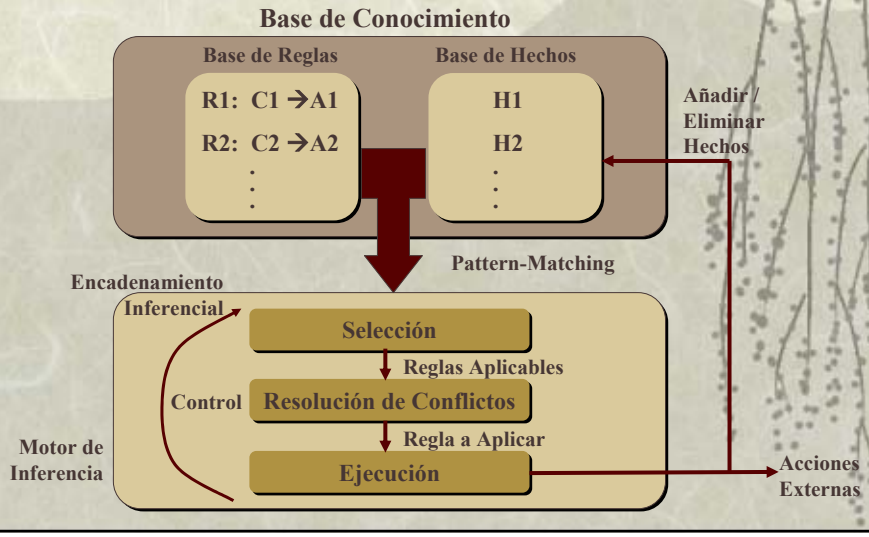
```
{ enfermedad^m = { arteriosclerosis }, edad^s = 70 }

( bind ?enfermedad ( insert ?enfermedad arteriosclerosis ) )
( bind ?edad 70 )
( paciente enfermedad arteriosclerosis )
( paciente edad 70 )
```

  - **Cláusulas**

```
{ enfermedad( paciente, arteriosclerosis ),
  ¬ enfermedad( paciente, regurgitación-aórtica ),
  edad( paciente ) = 70 }
```

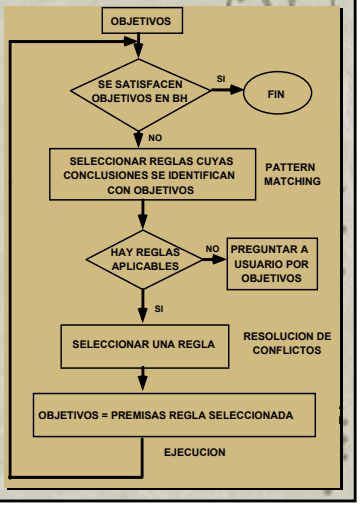

2.3.4. Inferencia en Sistemas Basados en Reglas



2.3.4. Inferencia en Sistemas Basados en Reglas

2.3.4.1. Encadenamiento Inferencial

- **Inferencia dirigida por el objetivo *(backward)***
  - *fbf implicación usadas como B-reglas:*
    - *Partiendo de la fbf objetivo, se obtienen BH objetivos parciales hasta que se satisface la condición de terminación (BH inicial).*
  - *Se deduce sólo lo que es necesario en la demostración.*
  - *Características:*
    - **Más parecido a la conducta humana (eficacia).**
    - **Obtiene respuestas a preguntas.**
    - **Permiten preguntar al usuario.**
    - **Necesita objetivos explícitos.**

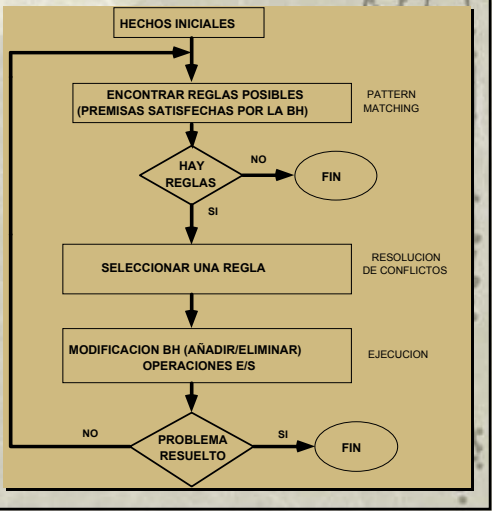


Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.4. Inferencia en Sistemas Basados en Reglas

2.3.4.1. Encadenamiento Inferencial

- **Inferencia dirigida por los datos *(forward)***
  - *fbf implicación usadas como F-reglas:*
    - *Operan sobre la BH inicial hasta satisfacer las fbf objetivos.*
  - *Se deduce todo lo que se puede deducir.*
  - *Características:*
    - **Sistemas generativos.**
    - **Obtiene todos los nuevos hechos deducibles.**

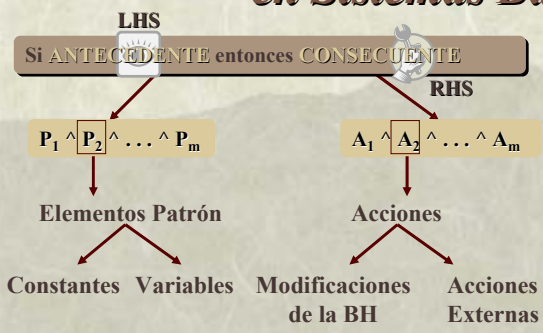


2.3.4. Inferencia en Sistemas Basados en Reglas

2.3.4.2. Control Inferencial

- En amplitud
- En profundidad
- Control inferencial (resolución de conflictos):
  - Complejidad de premisas
  - Peso de las reglas
  - Combinación: peso, complejidad
  - Mayor frecuencia
  - Última utilizada
  - Metarreglas:
    - Aprendizaje.
    - Utilización de reglas.
    - Exclusión de reglas.
    - Asignación dinámica de pesos.

2.3.5. Patrones en Sistemas Basados en Reglas



- El leng. de repr. del conoc. descrito hasta ahora nos permite utilizar:
  - Pares variable – valor
  - Tuplas obj. – atr. – valor
  - Reglas de Producción

- Vamos a enriquecer la expresividad de dicho lenguaje, mediante patrones, para la representación de:
  - Hechos de la BH (constantes)
  - Condiciones y Conclusiones de las RP (constantes y variables)

2.3.5. Patrones en Sistemas Basados en Reglas

- **Hechos como patrones**
  - Patrón (*pattern*): **secuencia finita ordenada de elementos:**
    (e_1 e_2 e_3..... e_n) donde cada elemento patrón e_i ($1 \leq i \leq n$) es:
    - **una constante: elemento patrón constante**
    - **una variable: elemento patrón variable.**
  - Variable – Valor: **puede ser representado por un patrón constituido por dos constantes:** x = 10 ≡ (x 10)
  - Objeto – atributo – valor: **puede ser representado por un patrón constituido por tres constantes:**

    (horno, temperatura, 100) ≡ (horno temperatura 100)
  - Hecho **(extensión a patrones): secuencia finita ordenada de elementos:**
    (f_1 . . . f_n) donde cada elemento patrón f_i ($1 \leq i \leq n$) es una cte.

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.5. Patrones en Sistemas Basados en Reglas

- **Ejemplos**

(paciente nombre Juan edad 18 enfermedad aneurisma)

(paciente nombre Luis edad 25 enfermedad gripe arteriosclerosis)

(paciente nombre Pedro edad 21 enfermedad neumonía)

(médico nombre Juan especialidad cardiología)

(manzana tamaño grande color verde)

(lista a b c d e)

Un patrón extiende la representación de hechos

2.3.5. Patrones en Sistemas Basados en Reglas

- **Patrones en las RP**
  - **Las condiciones y conclusiones de las reglas pueden referir a condiciones y conclusiones sobre patrones (con vars. y ctes.).**

```
( defrule regla-1
        ( e1  e2  e3 ..... en )
        =>
        ( assert ( a1  a2  a3 ..... am ) )
```

  - **Un elemento patrón e_i puede ser:**
    - Constante Patrón**: No contiene variables**
      (hambriento Pedro)
    - Variable Patrón Uni-Valor**: su nombre empieza con el signo?:**
      ?x, ?sexo, ?edad, ...
    - Variable Patrón Multi-Valor**: su nombre empieza con $?:**
      $?enfermedad, $?ocupación, ...
    - Variables sin enlace**: Su nombre solo consta del símbolo ? o $?. Son vars. especiales que pueden estar ligadas a cualquier cte. o secuencia de ctes., pero no preservan sus enlaces.**
      (paciente nombre ? edad ? enfermedad $?z)

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.5. Patrones en Sistemas Basados en Reglas

- **Distintas ocurrencias de una misma variable patrón en una regla denotan el mismo elemento.**

```
( defrule regla-1
        ( paciente nombre Juan edad ?y enfermedad $?z )
        =>
        ( assert  ( paciente nombre Juan edad ?y enfermedad gripe ) ) )
```

- **Mediante los patrones se generaliza la sintaxis de las RP, permitiendo el uso de variables (+ *pattern-matching*).**

2.3.5. Patrones en Sistemas Basados en Reglas

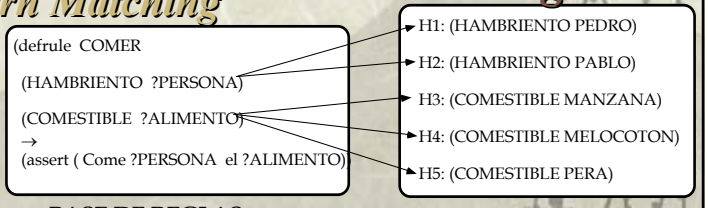
2.3.5.1. *Pattern Matching*

- ***Pattern Matching*: Hacer sintácticamente iguales un patrón de una regla de producción y un hecho asociando variables y hechos.**

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.5. Patrones en Sistemas Basados en Reglas

2.3.5.1. *Pattern Matching*



BASE DE REGLAS

BASE DE HECHOS

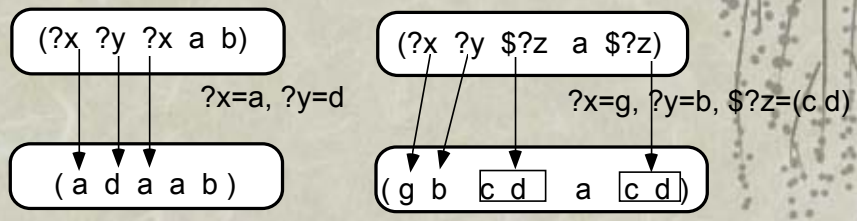
(p1 ? p2 ..... ?pn)
Asumible por defecto
Match: listas con variables instanciadas.
No match: lista vacía (≈ false).
not (p1 ? p2 .....? pn):
Asumible como la negación
Complementario a Igual.

I1: {?PERSONA = PEDRO, ?ALIMENTO = MANZANA} {H1, H3}
I2: {?PERSONA = PEDRO, ?ALIMENTO = MELOCOTON} {H1, H4}
I3: {?PERSONA = PEDRO, ?ALIMENTO = PERA} {H1, H5}
I4: {?PERSONA = PABLO, ?ALIMENTO = MANZANA} {H2, H3}
I5: {?PERSONA = PABLO, ?ALIMENTO = MELOCOTON} {H2, H4}
I6: {?PERSONA = PABLO, ?ALIMENTO = PERA} {H2, H5}

2.3.5. Patrones en Sistemas Basados en Reglas

2.3.5.1. *Pattern Matching*

- **Un Enlace (*binding*) de una var. es una cte. o una secuencia de ctes. que instancian la var.**
  - **?x = d : la variable mono-valor ?x está ligada a la constante d.**
  - **$?y = ($c_1$ ..... c_n) : la variable multi-valor $?y está ligada a la secuencia de constantes (c_1 ..... c_n)**

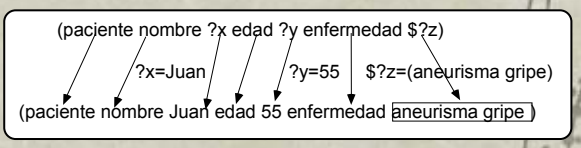


Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.5. Patrones en Sistemas Basados en Reglas

2.3.5.1. *Pattern Matching*

- **Un patrón (P) y un hecho (F) hacen *match* si existe un enlace de vars. de P tal que hace sintácticamente iguales P y F.**
- **Ejemplo: Enlace: ?x=Juan ?y=55 $?z=(aneurisma gripe)**



- **Las vars. patrón de un patrón son reemplazadas por una o más ctes. dependiendo del tipo (mono- o multi-valor).**
  - **Var. patrón mono-valor: sólo se instancia por una única cte.**
  - **Var. patrón multi-valor: se instancia por una secuencia de ctes.**
- **La múltiple ocurrencia de una var. en un patrón denota un mismo objeto.**

2.3.5. Patrones en Sistemas Basados en Reglas

2.3.5.1. *Pattern Matching*

- **Otros predicados de las Reglas de Producción:**
  - **= > < : se aplican a variables instanciadas.**
  - ***assert*** **: añadirá 'patrón' a la base de hechos. Las variables deben estar instanciadas.**
  - ***retract*** **: elimina el hecho de la base de hechos. Las variables deben estar instanciadas.**

2.3.5. Patrones en Sistemas Basados en Reglas

2.3.5.2. Ejemplo - 1

BH= {(Persona nombre Juan edad 10)}

```
( defrule regla-1
      ( persona nombre ?x edad ?y )
      ( test ( > ?y 9 ))
      =>
      ( assert  ( ?x etapa joven ) ) )
```

?x = Juan, ?y = 10

El hecho (Juan etapa joven) será añadido a la BH.

2.3.5. Patrones en Sistemas Basados en Reglas

2.3.5.2. Ejemplo - 2

BH ={(lista a b f g h) , (elemento g)}

```
( defrule R1
    ?f <- (lista ?x $?y)
    (elemento ?e)
    (test (neq ?e ?x))
     =>
    (retract ?f)
    (assert  (lista $?y)))
```

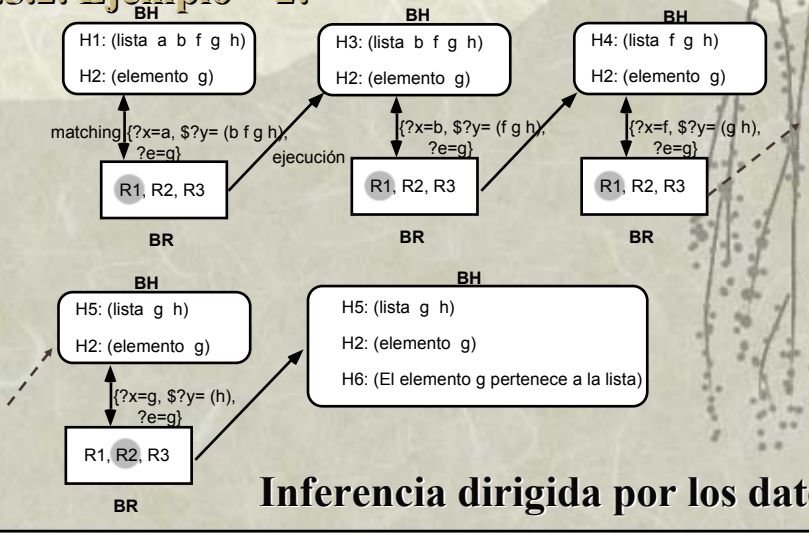
```
( defrule R2
   (lista ?x $?)
   (elemento ?x)
    =>
   (assert (El elemento ?x pertenece a la lista))
   (printout t "El elemento " ?x " pertenece a la lista"))
```

```
( defrule R3
   (lista)
   (elemento ?x)
    =>
   (assert (El elemento ?x no pertenece a la lista))
   (printout t "El elemento " ?x " no pertenece a la lista"))
```

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.5. Patrones en Sistemas Basados en Reglas

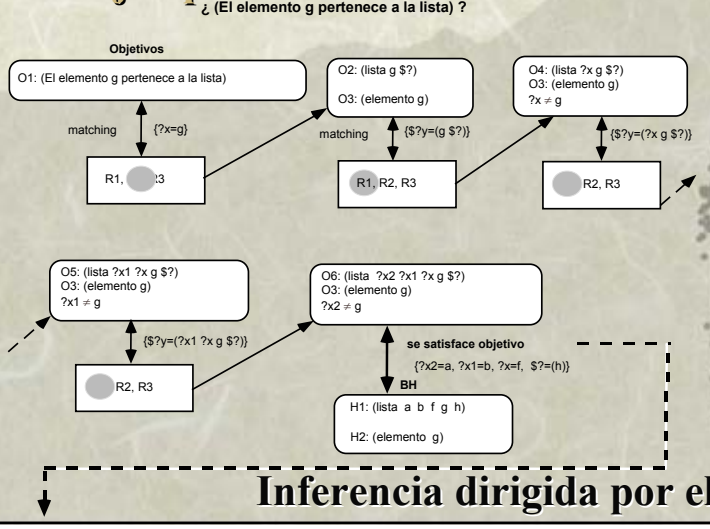
2.3.5.2. Ejemplo – 2:



Inferencia dirigida por los datos

2.3.5. Patrones en Sistemas Basados en Reglas

2.3.5.2. Ejemplo - 2

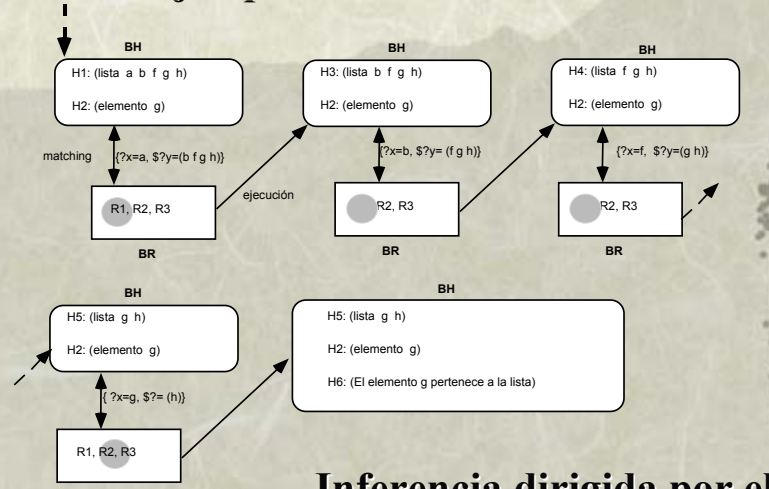


Inferencia dirigida por el objetivo

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.5. Patrones en Sistemas Basados en Reglas

2.3.5.2. Ejemplo - 2



Inferencia dirigida por el objetivo

2.3.5. Patrones en Sistemas Basados en Reglas

2.3.5.2. Ejemplo - 3

BH = {(libro nombre Quijote cuesta 5000 calidad media)
(persona nombre Juan gana 2000)}

```
( defrule R1 "Un libro que cuesta más de 3000, es bueno"
   (libro nombre ?x cuesta ?y calidad ?z)
   (test  (> ?y 3000))
    =>
   (assert (libro nombre ?x cuesta ?y calidad ?z alta)
))
```

```
( defrule R2 "Una persona que gana más de 1000 es rica"
   (persona nombre ?x gana ?y)
   (test  (> ?y 1000))
    =>
   (assert (persona nombre ?x status alto)
))
```

Tema 2 (II): Representación del Conocimiento en IA: Representación Basada en Reglas

2.3.5. Patrones en Sistemas Basados en Reglas

2.3.5.2. Ejemplo - 3

BH = {(libro nombre Quijote cuesta 5000 calidad media)
(persona nombre Juan gana 2000)}

```
( defrule R3 "Una persona de status alto, compra libros buenos"
   (persona nombre ?x status alto)
   (libro nombre ?y cuesta $? calidad $?z)
   (test  (member $?z alta))
    =>
   (assert (compra nombre ?x libro ?y)
))
```