



Sistemas de Producción

Mari Carmen Suárez de Figueroa Baonza

mcsuarez@fi.upm.es

Asunción Gómez Pérez

Jorge Gracia del Río

Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain



Objetivo del Tema

Conocer y saber utilizar los principales
elementos de los sistemas de
producción



Resultados de Aprendizaje

Ser capaz de representar conocimiento utilizando el formalismo de sistemas de producción

Ser capaz de identificar y describir el proceso de inferencia de un sistema de producción



Índice

1. Sistemas de Producción

- **Arquitectura**

2. Base de Hechos

3. Base de Reglas

4. Estrategia de Control

4.1. Ciclo de Funcionamiento

4.2. Encadenamientos



Reglas de Producción

- ❑ **Producción:** Término utilizado en Psicología Cognitiva para describir relaciones entre situaciones y acciones
- ❑ **Regla de Producción:** Término utilizado en IA para denotar un mecanismo de representación del conocimiento que implementa una producción
 - ❑ **SI situación ENTONCES acción**
 - ❑ La **situación** (condición ó antecedente) establece las condiciones que se han de satisfacer en un momento dado para que la regla sea aplicable
 - ❑ La **acción** (consecuente) establece las acciones que se han de realizar cuando la regla “se active”
 - ❑ Añadir/Suprimir algún dato (hecho)
 - ❑ Ejecutar algún procedimiento



Sistemas de Producción

❑ Sistema basado en reglas de producción

Rule 1 : IF s is NB THEN u_f is bigger

Rule 2 : IF s is NM THEN u_f is big

Rule 3 : IF s is Z THEN u_f is medium

Rule 4 : IF s is PM THEN u_f is small

Rule 5 : IF s is PB THEN u_f is smaller



Arquitectura de los Sistemas de Producción

SP: $\langle \text{BH}, \text{BR}, \text{EC} \rangle$

Base de Hechos

Memoria de Trabajo

Almacena los **datos** del problema

Guarda todos los cambios de información que se producen en el sistema durante cada proceso de inferencia

Base de Reglas

Base de Producciones

Representa los **conocimientos** sobre la solución del problema

Las reglas constan de dos partes: (a) condición o antecedente y (b) consecuente o acción

Las reglas operan sobre los datos de la base de hechos

Estrategia de Control

Motor de Inferencias o Intérprete de Reglas

Máquina deductiva

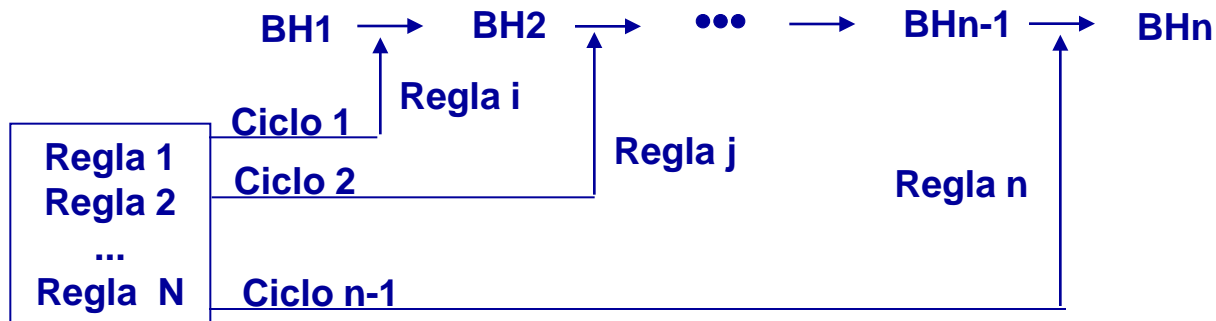
Realiza el razonamiento en ciclos que se van encadenando

Es el mecanismo que sirve para examinar la base de hechos y decidir que reglas se deben disparar



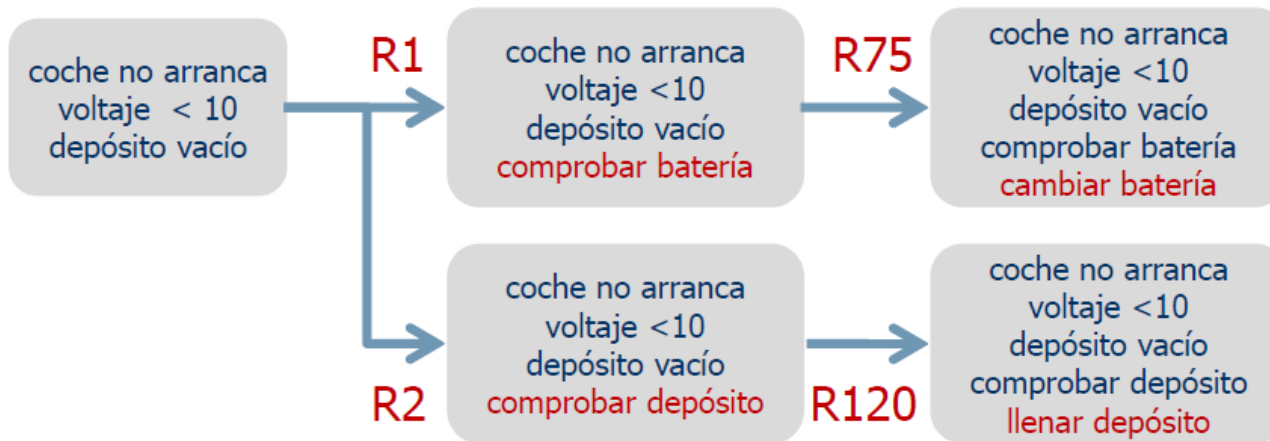
Motor de Inferencia

- ❑ ¿Cómo se van aplicando las reglas de la base de conocimiento sobre los hechos de la memoria de trabajo?
- ❑ El **motor de inferencia** es el encargado y realiza el razonamiento en ciclos que se van encadenando



- ❑ Pueden utilizarse **dos mecanismos básicos de inferencia**:
 - ❑ Hacia adelante (p.ej. CLIPS, Drools)
 - ❑ Hacia atrás (p.ej. PROLOG)

Motor de Inferencia: Ejemplo



El motor de inferencia determina el orden en el que se aplican las reglas 'activas' (aquellas para las que se cumple su antecedente)



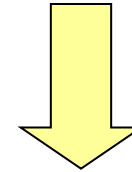
Sistema de Producción: Ejemplo (I)

Base de Reglas

- R1:** Si (Animal \$A) ^ (Esqueleto \$A sí)
Entonces (Vertebrado \$A)
- R2:** Si (Animal \$A) ^ (Esqueleto \$A no)
Entonces (Invertebrado \$A)
- R3:** Si (Vertebrado \$A) ^ (Ladra \$A)
Entonces (Perro \$A)

Base de Hechos

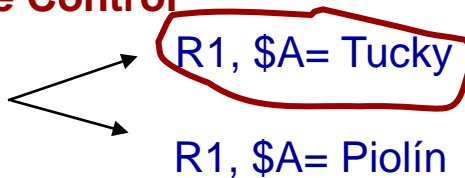
(Animal Tucky)
(Animal Piolín)
(Esqueleto Piolín sí)
(Esqueleto Tucky sí)
(Ladra Tucky)



(Animal Tucky)
(Animal Piolín)
(Esqueleto Piolín sí)
(Esqueleto Tucky sí)
(Ladra Tucky)
(Vertebrado Tucky)

Estrategia de Control

Ciclo 1:





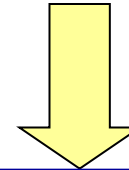
Ejemplo de Sistema de Producción (II)

Base de Reglas

- R1:** Si (Animal \$A) ^ (Esqueleto \$A sí)
Entonces (Vertebrado \$A)
- R2:** Si (Animal \$A) ^ (Esqueleto \$A no)
Entonces (Invertebrado \$A)
- R3:** Si (Vertebrado \$A) ^ (Ladra \$A)
Entonces (Perro \$A)

Base de Hechos

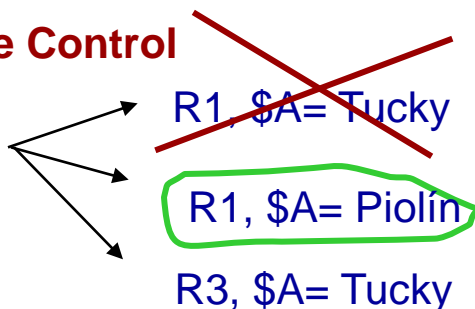
(Animal Tucky)
(Animal Piolín)
(Esqueleto Piolín sí)
(Esqueleto Tucky sí)
(Ladra Tucky)
(Vertebrado Tucky)



(Animal Tucky)
(Animal Piolín)
(Esqueleto Piolín sí)
(Esqueleto Tucky sí)
(Ladra Tucky)
(Vertebrado Tucky)
(Vertebrado Piolín)

Estrategia de Control

Ciclo 2:





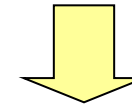
Ejemplo de Sistema de Producción (III)

Base de Reglas

- R1:** Si (Animal \$A) ^ (Esqueleto \$A sí)
Entonces (Vertebrado \$A)
- R2:** Si (Animal \$A) ^ (Esqueleto \$A no)
Entonces (Invertebrado \$A)
- R3:** Si (Vertebrado \$A) ^ (Ladra \$A)
Entonces (Perro \$A)

Base de Hechos

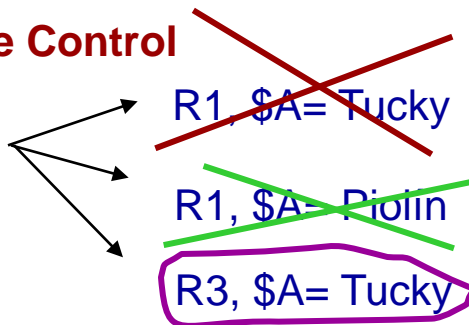
(Animal Tucky)
(Animal Piolín)
(Esqueleto Piolín sí)
(Esqueleto Tucky sí)
(Ladra Tucky)
(Vertebrado Tucky)
(Vertebrado Piolín)



(Animal Tucky)
(Animal Piolín)
(Esqueleto Piolín sí)
(Esqueleto Tucky sí)
(Ladra Tucky)
(Vertebrado Tucky)
(Vertebrado Piolín)
(Perro Tucky)

Estrategia de Control

Ciclo 3:





Mismo Ejemplo en CLIPS

<http://www.clipsrules.net/>

Base de Reglas

```
(defrule R1 (animal ?a) (esqueleto ?a si) => (assert (vertebrado ?a)))  
(defrule R2 (animal ?a) (esqueleto ?a no) => (assert (invertebrado ?a)))  
(defrule R3 (animal ?a) (vertebrado ?a) (ladra ?a) => (assert (perro ?a)))
```

Base de Hechos Inicial

```
(defacts estado-inicial  
  (animal Tucky)  
  (animal Piolín)  
  (esqueleto Piolín si)  
  (esqueleto Tucky si)  
  (ladra Tucky))
```



Índice

1. **Sistemas de Producción**
2. **Base de Hechos**
3. **Base de Reglas**
4. **Estrategia de Control**
 - 4.1. **Ciclo de Funcionamiento**
 - 4.2. **Encadenamientos**



Base de Hechos (BH)

Representa el **estado** actual de la tarea o problema

Contiene **conocimiento declarativo**

(Animal Tucky)
(Animal Piolín)
(Esqueleto Piolín sí)
(Esqueleto Tucky sí)
(Ladra Tucky)

- Base de Hechos Inicial: situación inicial
- Base de Hechos Final: situación objeto (puede haber más de una)
- Base de Hechos Intermedias: situación actual o en curso de resolución

Parte de la BH es permanente, otra es temporal (pertenece a la solución del problema en curso)

La ejecución de reglas modifica la BH al añadir o borrar hechos en ella



Base de Hechos (BH)

(Animal Tucky)
(Animal Piolín)
(Esqueleto Piolín sí)
(Esqueleto Tucky sí)
(Ladra Tucky)

Sintaxis:

- Cada elemento de la BH debe ir encerrado entre paréntesis
- Sólo hay constantes en la BH

Ejemplos:

- ❑ (Casilla 3 3 Roja): En la casilla (3 3) hay una ficha de color rojo
- ❑ (Fiebre Pepe 37,5): La fiebre de Pepe es de 37,5



Índice

1. Sistemas de Producción
2. Base de Hechos
3. **Base de Reglas**
4. Estrategia de Control
 - 4.1. Ciclo de Funcionamiento
 - 4.2. Encadenamientos



Base de Reglas

Representa conocimientos sobre la **solución del problema**

Si **Condiciones** Entonces **Acciones**

Antecedente

Consecuente

Ejemplos:

- El coste del envío se incrementa en 1€ si se recibe en el mismo día
- Si la edad del paciente es inferior a 10 años, tiene manchas rojas y fiebre, entonces tiene varicela
- Si el coche no arranca, lo primero a revisar es la batería
- Si el dólar baja, entonces hay que comprar dólares
-



Sintaxis de la Base de Reglas

R1: Si (Animal \$A) ^ (Esqueleto \$A sí)
Entonces Añadir (Vertebrado \$A)

- **Antecedente:**

- Cada **elemento de condición** debe ir encerrado entre paréntesis
- Los elementos de condición están formados por **átomos**
- Un átomo puede ser una **constante** o una **variable** (precedido del símbolo "\$")
(Esqueleto \$A sí)
- Los elementos de condición del antecedente van unidos por el símbolo AND ("^")
- **Negación:** Un elemento de condición puede estar negado ("¬")

- **Consecuente:**

- Predicados del consecuente: **Añadir** y **Borrar**
- Las acciones del consecuente van unidas por el símbolo **"and"** ("^")
- Las variables utilizadas en el consecuente son reemplazadas por los valores que tengan en el antecedente



Reglas: Ejemplo

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

Elemento de Condición

Si (Paciente \$p \$edad) and
(\$edad < 10) and
(Síntomas \$p Fiebre) and
(Síntomas \$p Manchas-rojas)
Entonces (Enfermedad \$p Varicela)

Constantes: Paciente, 10, Síntomas, Fiebre, Manchas-rojas, Enfermedad, Varicela

Variables: \$p, \$edad



Instanciación de una Regla

- Par formado por una regla y los elementos de MT que hacen cierta la regla
- En un ciclo de la BC, una regla puede tener cero, una o N instanciaciones

Ejemplo: 2 instanciaciones

$R_i: (A \text{ } \$x) \wedge \sim (B \text{ } \$x) \rightarrow \dots$

MT: (AC) (BD) (AB) (AA) (BA)

$Eq_1: \$x = C$

$Eq_2: \$x = B$

Conjunto Conflicto = $\{R_i (AC), R_i (AB)\}$



No son REGLAS

Problema 1: Una regla no es una estructura “If ... Then ... Else”

Ri: Si Condición-1 \wedge Condición-2 \wedge Condición-3 \wedge ...

Entonces acciones-1

En caso contrario acciones -2



Solución: Crear dos reglas, una para cada bloque de acciones

- Identificar las condiciones que no se cumplen para la ejecución de las “acciones-2”
- Identificar condiciones adicionales en alguno de los bloques de acciones



No son REGLAS

Problema 2: No puede aparecer OR en el consecuente de la regla

Ri: Si Condición-1 \wedge Condición-2 \wedge Condición-3 \wedge ...

Entonces acciones-1 \vee acciones-2



Solución:

- Crear dos reglas, una para cada bloque de acciones
- Establecer prioridades entre las reglas
- Identificar condiciones adicionales

Ri: Si Condición-1 \wedge Condición-2 \wedge Condición-3 \wedge ...

Entonces acciones-1

Rj: Si Condición-1 \wedge Condición-2 \wedge Condición-3 \wedge condiciones adicionales

Entonces acciones-2

Ri más prioritaria que Rj



No son REGLAS

Problema 3: No debe aparecer OR en el antecedente de la regla

Ri: Si Condición-1 \vee Condición-2
Entonces Acciones



Solución:

- Crear dos reglas, una para cada bloque de condiciones
- Establecer prioridades entre las reglas
- Intentar averiguar si faltan condiciones en alguna de las reglas
- Comprobar que las acciones son realmente las mismas

Ri: Si Condición-1 Entonces acciones

Rj: Si Condición-2 Entonces acciones

- Ri más prioritaria que Rj
- A la regla Rj se le añade la condición-3
-



No son REGLAS

Problema 4: En el consecuente de la regla no hay elementos de decisión

Ri: Si Condición-1 \wedge Condición-2 \wedge Condición-3 \wedge ...

Entonces acciones-1

Si Condición-4 \wedge ...

Entonces acciones-2



Solución:

- Nunca se deben introducir elementos de decisión en el consecuente de la regla
- Introducir señalizadores que provoquen la ejecución prioritaria de reglas con tales elementos
- No olvidar borrar los señalizadores

Ri: Si Condición-1 \wedge Condición-2 \wedge Condición-3 \wedge ...

Entonces acciones-1

Añadir (S)

Rj: Si S \wedge Condición-4 \wedge

Entonces acciones-2

Borrar (S)

Rj no se puede ejecutar
si no se ha ejecutado antes Ri

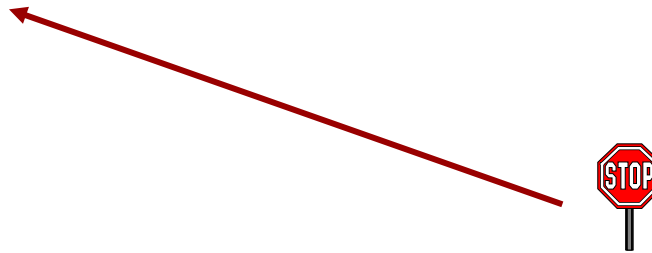
Las reglas con señalizadores, las mas prioritarias



No son REGLAS

Problema 5: Desde una regla nunca se puede lanzar otra regla

Ri: Si Condición-1 \wedge Condición-2
Entonces Rj



Solución:

- Introducir señalizadores que provoquen la ejecución de la regla Rj
- No olvidar borrar los señalizadores al ejecutar la regla Rj



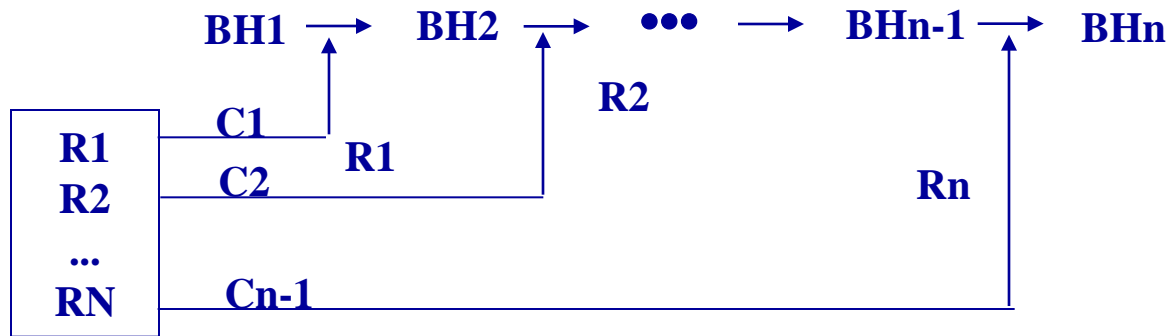
Índice

1. Sistemas de Producción
2. Base de Hechos
3. Base de Reglas
4. Estrategia de Control
 - 4.1. **Ciclo de Funcionamiento**
 - 4.2. Encadenamientos



Estrategia de Control

Examina en cada ciclo de funcionamiento la BH y decide qué regla ejecutar



Características:

- Causar movimiento
- Ser sistemática
- Ser eficiente



Estrategia de Control: Fases

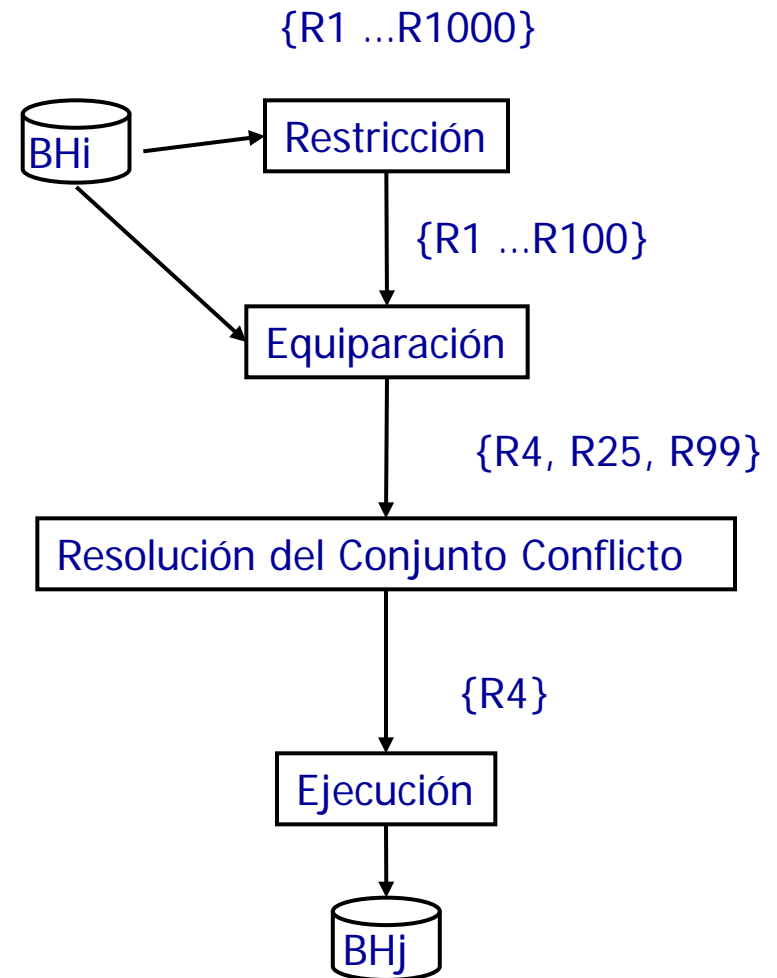
1. Selección

1.1. Restricción

1.2. Equiparación

1.3. Resolución del Conjunto Conflicto

2. Acción o Ejecución





Equiparación

Permite elegir aquellas reglas que conducen a la solución del problema,
al comparar cada una de las **condiciones** de las reglas
con el estado actual de la **Memoria de Trabajo**

El resultado de la equiparación es el **Conjunto Conflicto**, formado por
el conjunto de posibles reglas instanciadas que se pueden ejecutar

- Equiparación de Constantes
- Equiparación de Variables



Equiparación de Constantes

Las constantes son cualquier secuencia de caracteres no precedidos del símbolo \$

Una constante se equipara con otra constante igual a ella que ocupe la misma posición en un elemento de la MT

R1: (A B) \rightarrow (C D)

.....

MT: (A B)

(B C)

1 equiparación

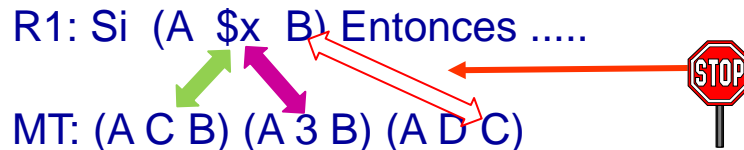
Conjunto Conflicto = {R1 (A B)}

La ejecución de la regla (R1(A B)) introduce en la MT el hecho (C D)



Equiparación de Variables (I)

Una variable que aparece **una sola vez** en una regla se equipara con cualquier valor que ocupe la misma posición en un elemento de la MT



Dos equiparaciones de la misma regla

$$\text{\$}x = C \leftrightarrow \text{eq.1}$$

$$\text{\$}x = 3 \leftrightarrow \text{eq.2}$$

$\text{\$}x = D$ no es posible porque B es distinto de C

Conjunto Conflicto = {R1 (A C B), R1 (A 3 B)}



Equiparación de Variables (II)

Una variable que aparece **dos o más veces** en una regla debe equipararse en todas las ocurrencias con el mismo valor

R1: (animal \$x) \wedge (piel \$x pelo) \rightarrow (especie \$x mamífero)

MT: (animal Tucky)

(piel Dolly pelo)

(piel Tucky pelo)

(animal Dolly)

(animal Dumbo)

(animal Piolin)

Eq1: \$x = Tucky : (animal Tucky) (piel Tucky pelo)

Eq2: \$x = Dolly : (animal Dolly) (piel Dolly pelo)

Eq3: \$x = Piolín: (animal Piolín) ---- (piel Piolín pelo) no existe en la MT

Eq4: \$x = Dumbo: (animal Dumbo) ---- (piel Dumbo pelo) no existe en la MT

Conjunto Conflicto = {R1 (animal Tucky), R1 (animal Dolly)}



Equiparación de Variables (III)

Se pueden equiparar variables distintas con el mismo valor

R1: Si (A \$x) (B \$y) Entonces

MT: (A C) (B C) (M V)

Eq1: \$x=C, \$y=C



Conjunto Conflicto = { R1 ((A C) (B C)) }



Equiparación de Variables (IV)

Se pueden realizar comprobaciones adicionales sobre las variables

Ri: Si (Casilla \$i \$j) (\$i < 4) (\$j > 0) ...

Rj: Si (Casilla \$i \$j \$color) (\$color ≠ Rojo) (\$i ≠ \$j) ...



Equiparación de Variables (V)

Las reglas pueden tener elementos de condición negados (precedidos del símbolo \sim)

La regla se equipara si se cumplen:

- a) Existen elementos de la MT que satisfacen todos los elementos de condición no negados
- b) No existen elementos de la MT que hacen cierto el elemento de condición negado

Hipótesis del Mundo Cerrado: Todo lo que no está en la MT (durante la equiparación de una regla) se considera falso



EJEMPLO



$R_i: (A \text{ \$ } x) \sim (B \text{ \$ } x) \rightarrow \dots$

MT: (AC) (BD) (AB) (AA) (BA)



EJEMPLO

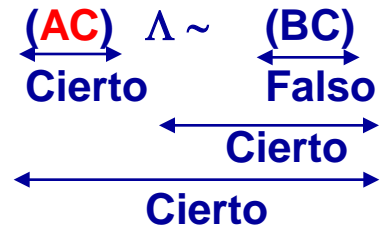


$R_i: (A \text{ \$ } x) \wedge \sim (B \text{ \$ } x) \rightarrow \dots$

MT: (AC) (BD) (AB) (AA) (BA)

Eq₁: $\$x = C :$

Válida



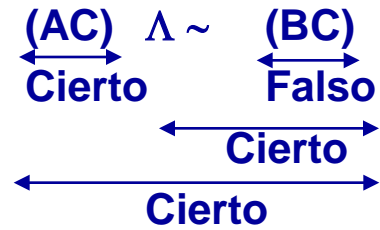


$R_i: (A \text{ \$x}) \sim (B \text{ \$x}) \rightarrow \dots$

MT: (AC) (BD) (**AB**) (AA) (BA)

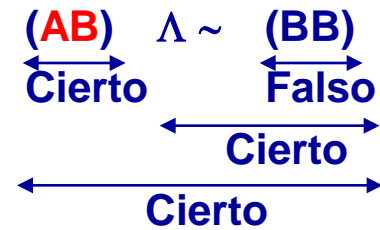
$Eq_1: \$x = C :$

Válida



$Eq_2: \$x = B$

Válida



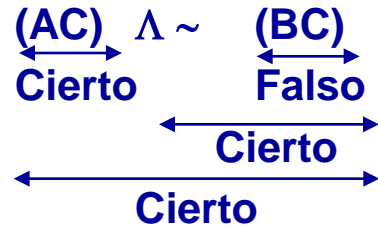


$R_i: (A \ \$x) \sim (B \ \$x) \rightarrow \dots$

MT: (AC) (BD) (AB) (AA) (BA)

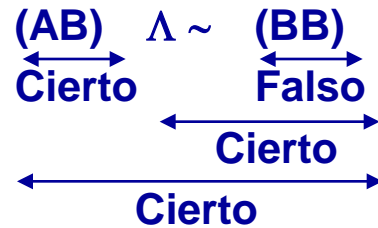
$Eq_1: \$x = C :$

Válida



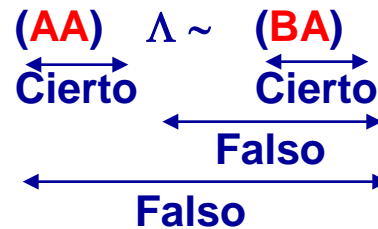
$Eq_2: \$x = B :$

Válida



$Eq_3: \$x = A :$

Inválida





Estrategia de Control: Fases

1. Selección

1.1. Restricción

1.2. Equiparación

1.3. Resolución del Conjunto Conflicto

2. Acción o ejecución



Resolución del Conjunto Conflicto

Conjunto Conflicto (CC): Conjunto de instanciaciones de posibles reglas ejecutables

Resolución del CC: Selección de la regla que va a ser ejecutada en la fase de ejecución

Estrategias de selección:

1. Explicitar un **orden lineal** en la BR
2. La regla de mayor **prioridad**
3. La regla más **específica**: con elementos de condición restrictivos
4. La regla más **general**
5. **Edad del elemento en la MT**: la regla con elementos de la BH más recientemente añadidos
6. Utilizar el **principio de refracción**: no pueden ejecutar instanciaciones de reglas ya ejecutadas
7. Arbitrariedad

Combinación de estrategias



Estrategias (1 y 2)

Explicito un orden lineal: Se selecciona la primera regla que equipara

Seleccionar la regla de prioridad más alta:

- La prioridad se establece en función del problema que se modeliza
- La prioridad la da el experto del dominio



Estrategia (3): Seleccionar la regla más específica

Si las instanciaciones de las reglas tienen elementos de condición iguales se selecciona la regla que tenga **más elementos** de condición

Ri: Si (Paciente \$x) ^ (Fiebre \$x) Entonces (Enfermo \$x)

Rj: Si (Paciente \$x) ^ (Fiebre \$x) ^ (Manchas Rojas \$x) Entonces (Varicela \$x)

En este caso, **Rj**



Estrategia (4): Seleccionar la regla más general

Si las instanciaciones de las reglas tienen elementos de condición iguales se selecciona la regla que tenga **menos elementos** de condición

Ri: Si (Paciente \$x) ^ (Fiebre \$x) Entonces (Enfermo \$x)

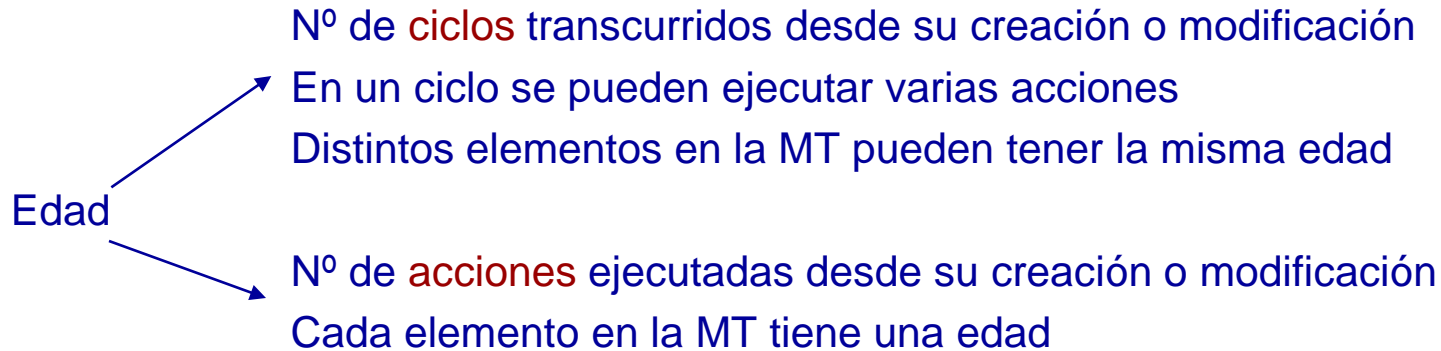
Rj: Si (Paciente \$x) ^ (Fiebre \$x) ^ (Manchas Rojas \$x) Entonces (Varicela \$x)

En este caso, **Ri**



Estrategia (5): Edad del elemento en la MT (I)

Cada elemento de la MT tiene asociado un **Time-Tag** (señal temporal) de cuándo se creó



El **contador** (por **ciclos/acciones**) se va incrementando en las unidades necesarias al ejecutar los ciclos/acciones

$$\text{Edad} = \text{Contador} - \text{TimeTag}$$

Time-Tag

BH: { (90 A C)
(80 A B)}

Contador: 96

Edad de (A C) es 6
Edad de (A B) es 16



Estrategia (5): Edad del elemento en la MT (II)

- Si el consecuente de una regla **modifica** un hecho, el Time-tag se actualiza al valor del contador
- Si el consecuente de una regla **crea** un hecho, el Time-tag es el del contador

Ejemplo: Seleccionar la instanciación con elementos añadidos más recientemente

R1: Si (A \$x) (\$x C) Entonces

R2: Si (B C) (\$x B) (C D) Entonces

MT: (98 (B C)) (92 (A B)) (94 (C D))

Contador: 100

CC = {(R1 (92 (AB)) (98 (BC)))
(R2(98 (BC)) (92 (AB)) (94 (CD)))}

Regla a ejecutar R2 por ser la regla que tiene el siguiente elemento más joven



Estrategia (6): Principio de Refracción (I)

Seleccionar una regla cuya **instanciación** no haya ocurrido previamente

Dos instanciaciones son distintas si se dan alguna de las tres situaciones:

1. Proceden de distintas reglas, aunque coincidan sus elementos
2. Las listas de elementos de la memoria de trabajo que contienen son distintas
3. Si una instanciación de una regla está en el CC en un instante T_a y T_c , existe un instante intermedio entre T_a y T_c tal que la instanciación no estuvo en el CC



Estrategia (6): Principio de Refracción (PR) (II)

Seleccionar una regla cuya **instanciación** no haya ocurrido previamente

Ejemplo:

Base de Reglas:

R1: (A) \rightarrow (B)

R2: (B) - (C) - (D) \rightarrow (C)

R3: (E) \rightarrow (D)

R4: (D) \rightarrow FIN

R5: (A) (C) \rightarrow (E) borrar (C)

Base de Hechos:

(1 (A))

Estrategia de Control:

Prioridad (1 .. 5) + PR

CICLO	BH	CC	Ejecutar
1	(1 (A))	<u>(R1 (1 (A)))</u>	R1
2	(1 (A)) (2 (B))	(R1 (1 (A))) <u>(R2 (2 (B)))</u>	R2
3	(1 (A)) (2 (B)) (3 (C))	(R1 (1 (A))) <u>(R5 (1 (A)) (3 (C)))</u>	R5
4	(1 (A)) (2 (B)) (4 (E))	(R1 (1 (A))) <u>(R2 (2 (B)))</u> (R3 (4 (E)))	R2
5	(1 (A)) (2 (B)) (4 (E)) (5 (C))	(R1 (1 (A))) <u>(R3 (4 (E)))</u> (R5 (1 (A)) (5 (C)))	R3
6	(1 (A)) (2 (B)) (4 (E)) (5 (C)) (6(D))	(R1 (1 (A))) (R3 (4 (E))) <u>(R4 (6(D)))</u> (R5 (1 (A)) (5 (C)))	R4



Estrategia de Control: Fases

1. Selección

1.1. Restricción

1.2. Equiparación

1.3. Resolución del Conjunto Conflicto

2. Acción o ejecución



Estrategia de Control: Fase de Acción

La ejecución de la regla modifica la BH actual en una BH nueva al AÑADIR y/o

BORRAR elementos de la primera

La BH nueva se tomará como punto de partida en el siguiente ciclo de funcionamiento



Índice

1. Sistemas de Producción
2. Base de Hechos
3. Base de Reglas
4. Estrategia de Control
 - 4.1. Ciclo de Funcionamiento
 - 4.2. Encadenamientos



Clasificación de los SP

Sistemas **dirigidos por el antecedente** (hacia delante):

Si el antecedente es verdad, el consecuente se procesa y se actúa sobre la MT

EC: encadenamiento hacia delante

Sistemas **dirigidos por el consecuente** (hacia atrás):

Demostrar un determinado consecuente probando recursivamente sus antecedentes

- a) Un antecedente es cierto si está en la BH del sistema
- b) Si el antecede no está en la BH, se busca si es consecuente de alguna regla y se prueban recursivamente los antecedentes de dicha regla
- c) Si no se dan las situaciones a) y b), entonces se supone la hipótesis del mundo cerrado



Encadenamiento hacia delante

- Dada una regla **R**: **Si A Entonces C**
 - La regla R se puede disparar, ejecutar o aplicar **hacia adelante** cuando se satisfacen las condiciones especificadas en su **antecedente A**
 - Cuando la regla se dispara, se procede a la ejecución de las acciones especificadas en su consecuente C.
- Un motor de inferencia con **encadenamiento hacia adelante**:
 - parte de unos **hechos** (datos en la memoria de trabajo),
 - compara (empareja) los **datos** de la memoria de trabajo con los **antecedentes** de las reglas, y
 - dispara **reglas** hasta que se satisface algún objetivo o hasta que ninguna regla es aplicable



Encadenamiento hacia delante: Ejemplo

MT (Inicial)

- Temperatura = 40
- Enfermo 2 semanas
- Garganta Inflamada

R0

- IF hay placas (puntos blancos) en la garganta
- THEN diagnóstico: posible infección de garganta

R1

- IF garganta inflamada AND sospechamos infección bacteriana
- THEN diagnóstico: posible infección de garganta

R2

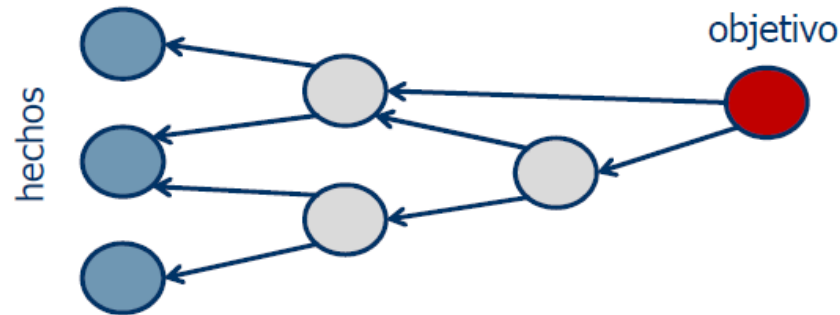
- IF temperatura paciente > 39
- THEN paciente tiene fiebre

R3

- IF paciente enfermo más de una semana AND paciente tiene fiebre
- THEN sospechamos infección bacteriana

Encadenamiento hacia atrás

- Un motor de inferencia con **encadenamiento hacia atrás**:
 - parte de unos hechos (datos en la memoria de trabajo) y de un **objetivo inicial**,
 - compara (empareja) los consecuentes de las reglas con la lista de objetivos, y
 - dispara reglas (hacia atrás) (aumentando así la lista de objetivos) hasta que todos ellos coincidan con hechos de la memoria de trabajo





Sistemas de Producción: Bibliografía

- **Ingeniería del Conocimiento** (ED Ceura)
A. Gómez, N. Juristo, C. Montes, J. Pazos
- **Inteligencia Artificial** (ED Ceura)
D. Borrajo, N. Juristo, V. Martínez, J. Pazos
- **Artificial Intelligence**
Rich and Knight
- **Inteligencia Artificial** (Mc Graw Hill)
J.Palma, R. Marín (Eds)



Sistemas de Producción

Mari Carmen Suárez de Figueroa Baonza

mcsuarez@fi.upm.es

Asunción Gómez Pérez

Jorge Gracia del Río

Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain