Representación de Conocimientos

Asunción Gómez-Pérez
asun@fi.upm.es

Departamento de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain

Indice

Sistemas de Producción

Asunción Gómez-Pérez
asun@fi.upm.es

Departamento de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo sn,
28660 Boadilla del Monte, Madrid, Spain

Bibliografía de Sistemas de Producción

- **Ingeniería del Conocimiento** (ED Ceura)

  A. Gómez, N. Juristo, C. Montes, J. Pazos

- **Inteligencia Artificial** (ED Ceura)

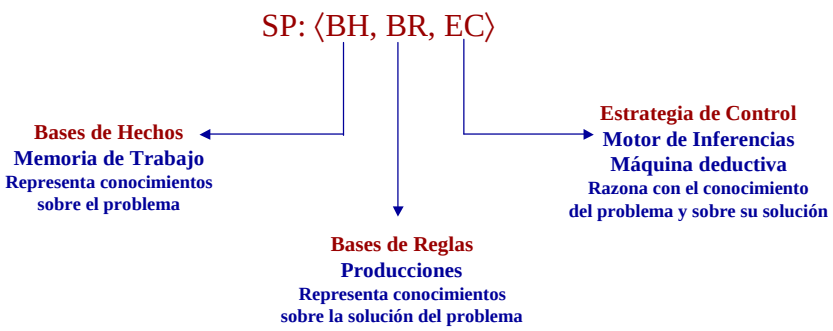
  D. Borrajo, N. Juristo, V. Martínez, J. Pazos

- **Artificial Intelligence**

  Rich and Knight

Indice

Arquitectura de los Sistemas de Producción



Indice

Base de Hechos

Representa el estado actual de la tarea o problema

- **Estado Inicial: situación inicial**
- **Estado Final: situación objeto (puede haber más de uno)**
- **Estados Intermedios: situación actual o en curso de resolución**

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

Contiene conocimiento declarativo.

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

Indice

Base de Reglas

Representa conocimientos sobre la solución del problema

Si Condiciones Entonces Acciones

Antecedente **Consecuente**

Ejemplos:

- **El coste del envío se incrementa en 1000 pesetas 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 bateria**
- **Si el dólar baja, entonces hay que comprar dólares**
- **.....**

Sintaxis

- **Cada elemento de condición debe ir encerrado entre paréntesis.**
- **Los elementos de condición del antecedentes van unidos por el símbolo "and"**
- **Los elementos de condición están formados por átomos**
- **Un átomo puede ser una constante o una variable (precedido del símbolo "$")**

  ($1 E $2)
- **Las variables utilizados en el consecuente, son reemplazados por los valores que tengan en el antecedente**
- **Predicados del consecuente: Añadir y Borrar**
- **Las acciones del consecuente van unidos por el símbolo "and"**

Ejemplos de Reglas

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

Ejemplos de Reglas

"El coste del envío se incrementa en 1000 pesetas si se recibe en el mismo día"

Si (envío $e $origen $destino) ^ (dia-entrega $e hoy) ^ (coste $e $coste)

Entonces Borrar (coste $e $coste)

Añadir (coste $e $coste + 1000)

Constantes: envío, dia-entrega, coste, hoy

Variable: $e, $coste, $origen, $destino

No son REGLAS

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

Ri: Si Condición-1 L **Condición-2** L **Condición-3** L **...**

Entonces acciones-1

En caso contrario acciones -2

Solución:

- **Identificar las condiciones que no se cumplen para la ejecución de las "acciones-2"**
- **Identificar condiciones adicionales en alguno de los bloques de acciones**
- **Crear dos reglas, una para cada bloque de acciones.**

No son REGLAS

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

Ri: Si Condición-1 L Condición-2 L Condición-3 L ...
Entonces acciones-1 V acciones-2

Solución:

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

Ri: Si Condición-1 L Condición-2 L Condición-3 L ...
Entonces acciones-1
Rj: Si Condición-1 L Condición-2 L Condición-3 L ...
Entonces acciones-2
Ri más prioritaria que Rj

No son REGLAS

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

Ri: Si Condición-1 V Condición-2
Entonces Acciones

Soluciones:

- **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 Condiciones-1 L **Condición-2** L **Condición-3** L **...**

Entonces acciones-1

Si condiciones-4 L**...**

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 Condiciones-1 L **Condición-2** L **Condición-3** L **...**

Entonces acciones-1

Añadir (S)

Rj: Si S ^ condiciones-4 L

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 L **Condición-2**

Entonces Rj

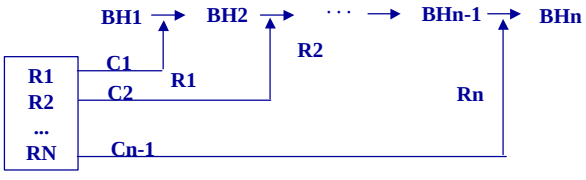
Solución:

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

Indice

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

Ejemplo de Sistemas de Producción

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: R1, $A= Tucky

R1, $A= Piolín

Ejemplo de Sistemas de Producción

Base de Reglas

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

R2: Si (Animal $A) ^(Esqueleto $A no)
Entonces (Invertebrado $A)

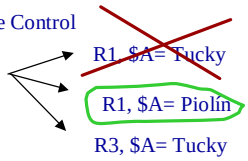
R3: Si (Vertebrado $A) (Ladra $A)
Entonces (Perro $A)

Base de Hechos

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

Estrategia de Control

Ciclo 2:



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

Ejemplo de Sistemas de Producción

Base de Reglas

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

R2: Si (Animal $A) ^(Esqueleto $A no)
Entonces (Invertebrado $A)

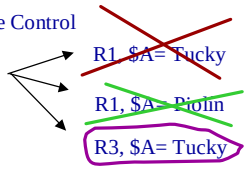
R3: Si (Vertebrado $A) (Ladra $A)
Entonces (Perro $A)

Base de Hechos

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

Estrategia de Control

Ciclo 3:



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

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 M.T.

R1: (A B) ® (C D)

.....

MT: (A B)

(B C)

1 equiparación

Conjunto Conflicto ={R1 (A B)}

La ejecución de la regla introduce en la MT a (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 M.T.

R1: Si (A $x (B) Entonces .....

MT: (A C B) (A 3 B) (A D C)

2 Equiparaciones de la misma regla

$x = C « eq. 1

$x = 3 « eq.2

$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) L **(piel pelo $x) ® (especie mamífero $x)**

MT: (animal Tucky)

(Piel Pelo Dolly)

(Piel Pelo Tucky)

(animal Dolly)

(animal Dumbo)

Eq1: $x = Tuky : (animal Tuky) (Piel Pelo Tuky)

Eq2: $x = Dolly : (animal Dolly) (Piel Pelo Dolly)

Eq3: $x = Dumbo: no se equipara por no tener (Piel Pelo Dumbo) en 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 ¹ Rojo) ($i ¹ $j) ...

Equiparación de variables (V)

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

La regla se equipara si:

a) Existen elementos de la MT que satisfacen todos los elementos de condición no negados

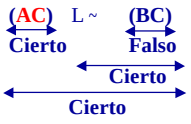
b) No existen elementos de la MT que hacen cierto el elemento de condición negado

Hipótesis del Mundo Cerrado: Todo lo que no está en la MT es falso

R_i: (A $x) ~ (B $x) ® ...
MT: (AC) (BD) (AB) (AA) (BA)

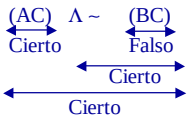
R_i: (A $x) ~ (B $x) ® ...
MT: (AC) (BD) (AB) (AA) (BA)

Eq_1: $x = C :
Válida

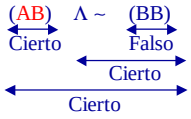


R_i: (A $x) ~ (B $x) → ...
MT: (AC) (BD) (AB) (AA) (BA)

Eq_1: $x = C :
Válida

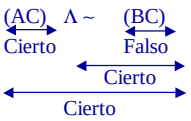


Eq_2: $x = B
Válida

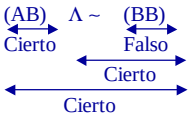


R_i: (A $x) ~ (B $x) → ...
MT: (AC) (BD) (AB) (AA) (BA)

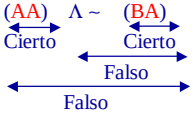
Eq_1: $x = C :
Válida



Eq_2: $x = B
Válida



Eq_3: $x = A :
Inválida



Instanciación de una Regla

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

Ejemplo anterior: 2 instanciaciones

R_i: (A $x) ~ (B $x) ® ...
MT: (AC) (BD) (AB) (AA) (BA)

Eq_1: $x = C :
Eq_2: $x = B

Conjunto Conflicto ={ Ri (AC) Ri, (AB)}

Hipótesis del Mundo Cerrado

Supone que es falso todo lo que no se encuentra en la MT al equiparar la regla.

Estrategia de Control: Fases

1. **Selección**
   - 1.1. **Restricción**
   - 1.2. **Equiparación:**
   - 1.3. **Resolución del Conjunto Conflicto**
2. **Acción o ejecución**

Conjunto Conflicto (CC): conjunto de instanciaciones de posibles reglas ejecutable

Resolución del Conjunto Conflicto

Selección de la regla que va a ser ejecutada en la fase de ejecución

Estrategias de selección:

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

Combinación de estrategias

Estrategias

Explícito 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**

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

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

Rj: Si (Envio $e) (frágil $e) Entonces (Desplazar $e cuidadosamente)

En este caso, ninguna regla es más específica que la otra

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

Edad del elemento en la MT

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

Edad

Nº de ciclos transcurridos desde su creación o modificación.
En un ciclo se pueden ejecutar varias acciones
Distintos elementos en la MT pueden tener la misma edad

Nº de acciones ejecutadas desde su creación o modificación.
Cada elemento en la MT tiene una edad.

El contador (por ciclos/acciones) se va incrementando en una unidad al ejecutar los ciclos/acciones

Antigüedad = Contador - TimeTag

Time-Tag

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

Contador: 96

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

Edad del elemento en la MT

- **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 (BC) ($X B) (CD) Entonces ....

MT: (98 (BC)) (92 (AB)) (94 (CD))

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.

Principio de Refracción

Seleccionar una regla cuya instanciación no haya ocurrido previamente.

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

- **Proceden de distintas reglas, aunque coincidan sus elementos**
- **Las listas de elementos de la memoria de trabajo que contienen son distintas**
- **Si una instanciación A estaba en el CC en un isntante Ta y una instanciación B estaba en el CC en un isntante Tb, existe un instante Tc entre Ta y Tb tal que A y B no estaban en el CC o no estaban a la vez.**

Principio de Refracción

Seleccionar una regla cuya instanciación no haya ocurrido previamente.

Ejemplo:

Base de Reglas:
R1: (A) ® (B)
R2: (B) - (C) - (D) ® (C)
R3: (E) ® (D)
R4: (D) ® FIN
R5: (A) (C) ® (E) quitar (C)

Base de Hechos
(1 (A))

Estrategia de Control
Prioridad (1 .. 5) + PR

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

Estrategia de Control: Fase de Acción

La ejecución de la regla modifica la BH actual en una BH nueva al AÑADIR Y BORRAR elementos de la primera.

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

Problemas al añadir y borrar

Base de Reglas

R1: **Si (envío $e $origen $destino) ^ (dia-entrega $e hoy) ^ (coste $e $coste)**
Entonces Añadir (coste $e $coste + 1000)

Base de Hechos

(envío E-1 Madrid Barcelona)
(dia-entrega E-1 hoy)
(coste E-1 700)

Equiparación

(envío E-1 Madrid Barcelona)
(dia-entrega E-1 hoy)
(coste E-1 700)
(coste E-1 1700)

Estrategia de Control

Ciclo 1: →
$e= E1
$origen = Madrid
$destino = Barcelona
$coste = 700

Error: El coste del envío es único
Solución: modificar la regla

Ejemplo

Base de Reglas

R1: **Si (envío $e $origen $destino) ^ (dia-entrega $e hoy) ^ (coste $e $coste)**
Entonces Añadir (coste $e $coste + 1000)
Borrar (coste $e $coste)

Base de Hechos

(envío E-1 Madrid Barcelona)
(dia-entrega E-1 hoy)
(coste E-1 700)

(envío E-1 Madrid Barcelona)
(dia-entrega E-1 hoy)
(coste E-1 700)
(coste E-1 1700)

Estrategia de Control

Ciclo 1: →
$e= E1
$origen = Madrid
$destino = Barcelona
$coste = 700

Error: El principio de Refracción,
no impide que se vuelve a ejecutar la regla

Ejemplo

Base de Reglas

R1: **Si (envío $e $origen $destino) ^ (dia-entrega $e hoy) ^ (coste $e $coste)**
Entonces Añadir (coste $e $coste + 1000)

Borrar (dia-entrega $e hoy)

Borrar (coste $e $coste)

Base de Hechos

(envío E-1 Madrid Barcelona)
(dia-entrega E-1 hoy)
(coste E-1 700)

(envío E-1 Madrid Barcelona)
(dia-entrega E-1 hoy)
(coste E-1 700)
(coste E-1 1700)

Estrategia de Control

Ciclo 1: →
$e= E1
$origen = Madrid
$destino = Barcelona
$coste = 700

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.