



Representación del conocimiento en Sistemas de Producción

Encadenamiento hacia adelante



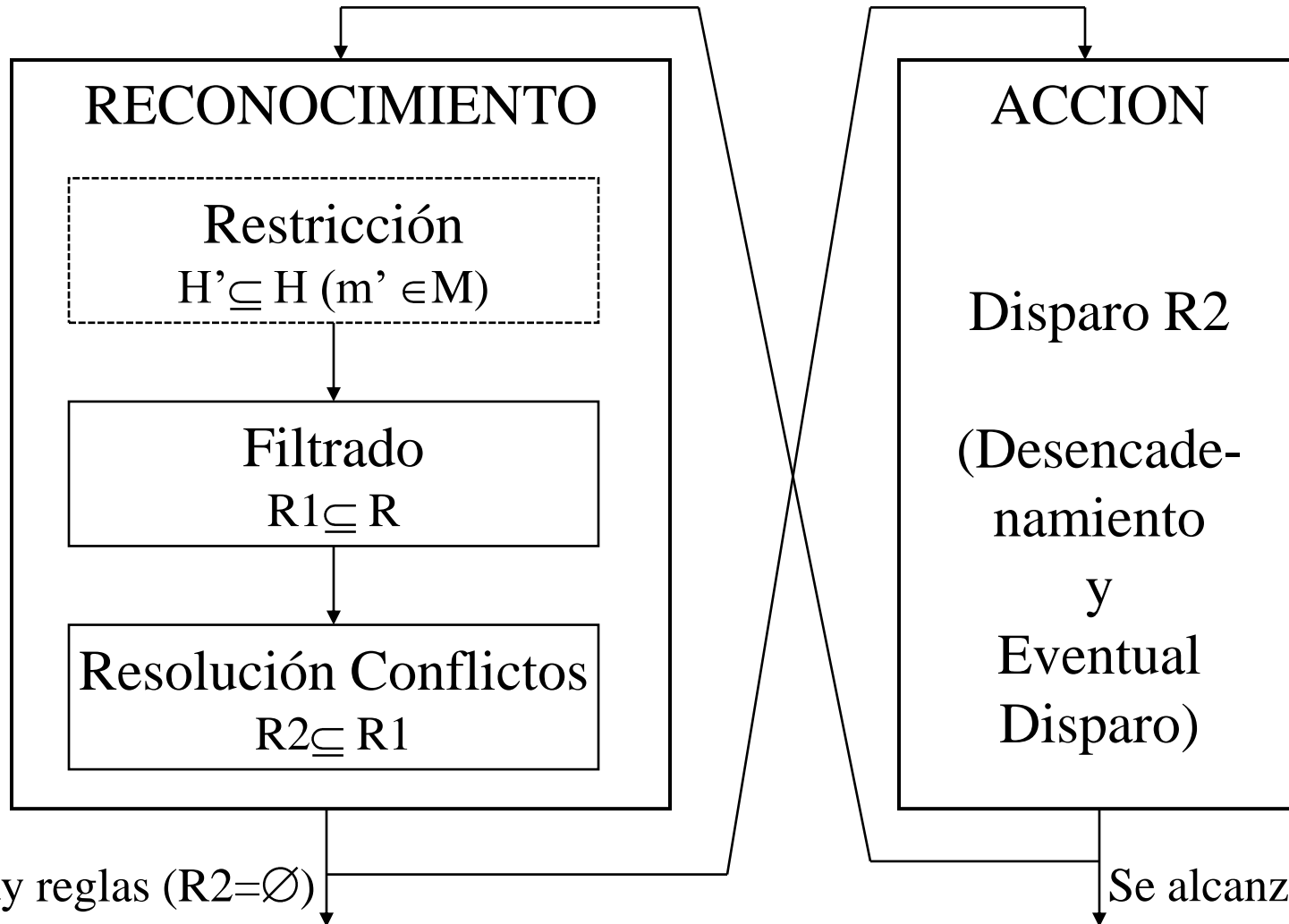


Obtención de valores

- El motor de inferencias
 - dispara reglas hasta obtener el valor de una característica especificada (meta)
 - suele proporcionar mecanismos adicionales de obtención de valores (por ejemplo, preguntar al usuario)
- Ampliación declaración características
 - metas: subíndice **g**
 - hechos que se pueden preguntar: subíndice **p**

	univaluada	multivaluada
tipada	$x_g^s : \tau$	$x_g^m : 2^\tau$
no tipada	x_g^s	x_g^m

Ciclo básico Reconocimiento-Acción





Elementos ciclo básico

- **Restricción**
seleccionar un subconjunto H' de los hechos H (o meta m')
- **Filtrado**
seleccionar todas las reglas activadas: reglas cuyo antecedente se satisfacen según H' (o cuyo consecuente añade un valor a m')
R1: conjunto conflicto
- **Resolución de Conflictos**
Seleccionar mediante algún criterio de resolución de conflictos un subconjunto R2 de R1
- **Acción**
disparar las reglas de R2, realizando su acción (o desencadenar para intentar disparar)



Control búsqueda

- Dirección de Búsqueda
 - Encadenamiento hacia adelante: de los hechos a las metas
 - Encadenamiento hacia atrás: de las metas a los hechos
- Régimen Tentativo/Irrevocable
 - Típicamente, adelante irrevocable, atrás tentativo
- Primero Anchura/Profundidad
 - Atrás, generalmente profundidad (backtracking)
- Estrategias de Resolución de Conflictos
- Orden Evaluación Premisas (operacional)



Estrategias de Resolución de Conflictos

- Refracción
Cada regla sólo se puede disparar una vez con los mismos elementos de la memoria de trabajo
- Reciencia
Seleccionar la regla que se satisfaga con los hechos más recientemente añadidos a la memoria de trabajo
- Especificidad
Seleccionar la regla que contenga más premisas
- Prioridad
Seleccionar la regla con máxima prioridad, fijada por el programador
- Orden
Primera regla, según orden en la base
- Todas



Encadenamiento hacia adelante

- Planteamiento básico
 - Parte del conjunto de Hechos, H
 - genera nuevos hechos disparando reglas
 - parada: se alcanza la meta o no hay reglas activadas
- Interpretación Directa de Reglas

$A, B, C \rightarrow K$

“si la memoria de trabajo satisface A, B y C, entonces añadir K”
- Uso
 - inicialmente se dispone de suficientes datos
 - no hay una meta clara

Ejemplo encadenamiento hacia adelante (I)

$D = \{x^m, y^m, z^s, u^m, v^m, w^m\}$

R_1 **if** iguales(w,a) **and** iguales(x,b) **then** añadir(v, c) **fi**

R_2 **if** iguales(w,d) **and** iguales(v,c) **then** añadir(y, e) **fi**

R_3 **if** iguales(v, c) **then** añadir(z, k) **fi**

R_4 **if** iguales(x,j) **and** iguales(y,e) **then** añadir(z, k) **fi**

R_5 **if** iguales(u,f) **and** iguales(x,g) **then** añadir(z, l) **fi**

$H = \{w=\{a\}, x=\{b,j,g\}, u=\{f\}\}$

Resolución Conflictos: refracción, orden

Iter.	Conjunto Conflictivo	Res. Conflictos	Mod. Mem. Trabajo
1	R_1, R_5	R_1	+ v={c}
2	R_1, R_3, R_5	R_3	+ z=k

Comprobar que si no se declara la meta, se obtiene z=l.

Ejemplo encadenamiento hacia delante (II)

$D = \{x^m, y^m, z^s, u^m, v^m, w^m\}$

R_1 **if** iguales(w,a) **and** iguales(x,b) **then** añadir(v, c) **fi**

R_2 **if** iguales(w,d) **and** iguales(v,c) **then** añadir(y, e) **fi**

R_3 **if** iguales(v, c) **then** añadir(z, k) **fi**

R_4 **if** iguales(x,j) **and** iguales(y,e) **then** añadir(z, k) **fi**

R_5 **if** iguales(u,f) **and** iguales(x,g) **then** añadir(z, l) **fi**

$H = \{w=\{a\}, x=\{b,j,g\}, u=\{f\}\}$

Resolución Conflictos: refracción, especificidad y orden

Iter.	Conjunto Conflictivo	Res. Conflictos	Mod. Mem. Trabajo
1	R_1, R_5	R_1	+ $v=\{c\}$
2	R_1, R_3, R_5	R_5	+ $z=l$

Comprobar que si no se declara la meta, se obtiene $z=k$.