



Inteligencia
Artificial

Tema 7

Sistemas basados en Reglas

Elisa Guerrero Vázquez

Esther L. Silva Ramírez

Departamento Ingeniería Informática

Departamento de
Ingeniería Informática



Objetivos

Al finalizar el tema el alumno debe ser capaz de:

- Diseñar Sistemas Basados en el Conocimiento (SBR) completos, construyendo tanto la base de hechos como la base de conocimientos.
- Saber realizar inferencias sobre SBR ya contruidos para obtener nuevo conocimiento.
- Comprender, conocer y saber simular el funcionamiento interno del motor de inferencias en un SBR:
 - el ciclo de ejecución,
 - modos de inferencia y razonamiento,
 - técnicas de equiparación,
 - así como usar distintas estrategias para la resolución de conflictos.
- Saber implementar un SBR usando las herramientas de desarrollo apropiadas para ello.

Contenidos

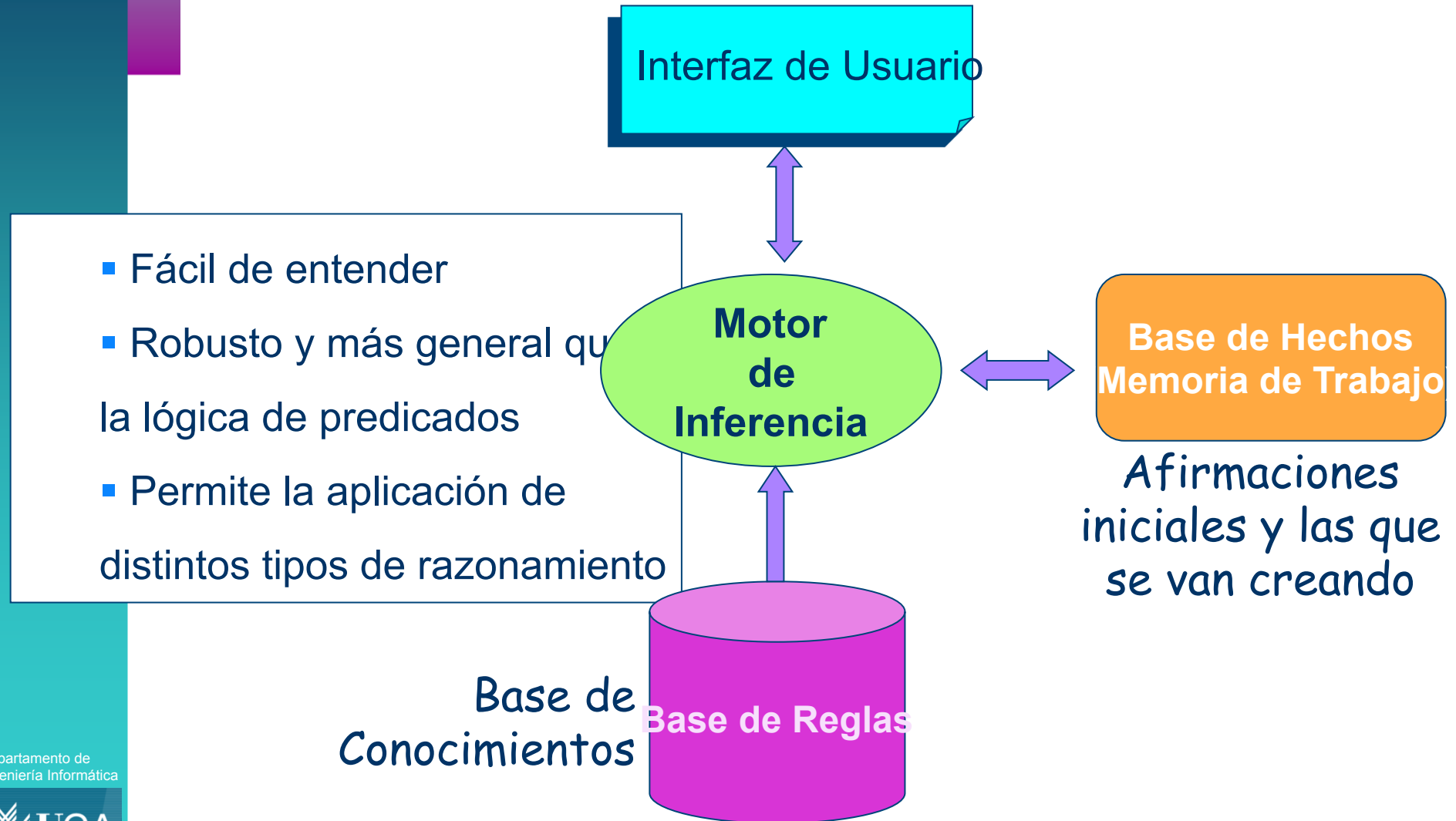
1. Definición de Sistemas basados en Reglas
2. Motor de Inferencia
 - Encadenamiento hacia delante
 - Encadenamiento hacia atrás
3. Técnicas de Equiparación
 - RETE
4. Resolución de Conflictos

1. Características Básicas de los SBR

- **Base de hechos** o memoria de trabajo (BH o WM): conocimiento sobre el dominio del problema
- **Base de reglas** (BR): conceptos y relaciones estáticas entre conceptos. Conjunto de reglas (producciones)
 - SI A ENTONCES B
 - A: condiciones de aplicación
 - B: acciones sobre la BH o mundo externo
- **Estrategia de control**, intérprete de reglas, o motor de inferencias (EC o MI): responsable de encadenar los ciclos de funcionamiento.
 - Fase de decisión: selección de reglas
 - Fase de acción: ejecución de reglas

Una regla se activa cuando sus precondiciones son ciertas en el estado actual de la BH o cuando la regla concluye algo que se busca establecer

Sistemas Basados en Reglas



Reglas de Producción

SBR: SISTEMAS BASADOS EN REGLAS

si <antecedente> **entonces** <consecuente>

- Independencia de unas reglas respecto de otras: una regla no puede hacer referencia a otra regla
- Carácter aproximado del conocimiento que se maneja. Factor de certidumbre

Ejemplo 1

Observaciones

- Especie 1 tiene color rosa
- Especie1 vuela
- Especie 1 pone huevos
- Especie 1 tiene patas largas

Base de Hechos
(Memoria de Trabajo)

Conocimientos

- **Si** vuela y pone huevos **entonces** es un ave
- **Si** un ave tiene patas largas, y color rosa **entonces** es un flamenco

Base de Reglas

Ejemplo 1

**Motor
de
Inferencia**

Base de Reglas

- **R1:** Si vuela y pone huevos entonces es un ave
- **R2:** Si un ave tiene patas largas, y color rosa entonces es un flamenco

Base de Hechos

- Especie 1 tiene color rosa
- Especie 1 vuela
- Especie 1 pone huevos
- Especie 1 tiene patas largas

2. Motor de Inferencia

- Examina la base de hechos y determina qué regla disparar a continuación.

1. Procedimiento de Control

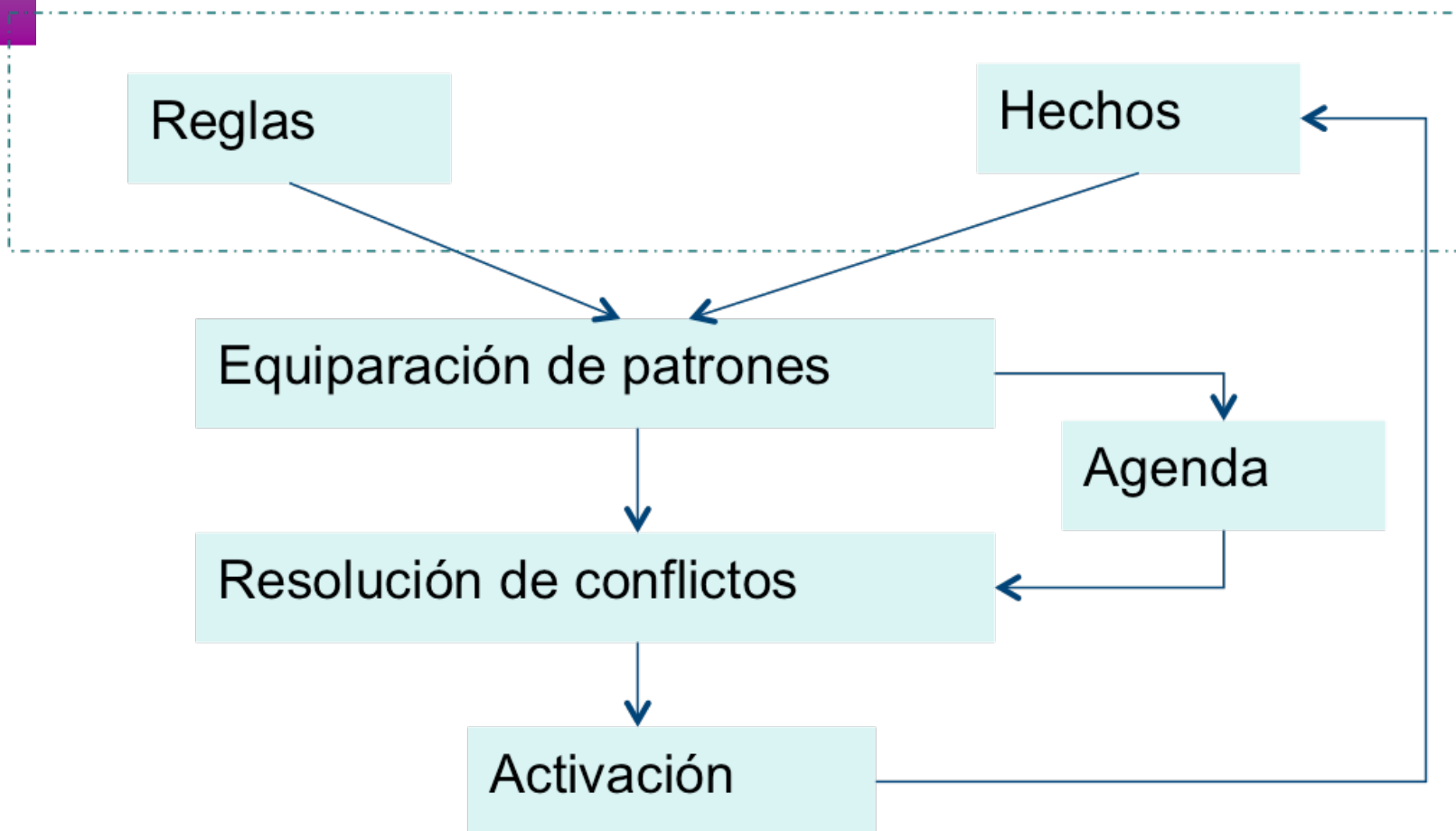
2. Modos de Inferencia

3. Tipo de Razonamiento

4. Técnicas de Equiparación

5. Resolución de Conflictos

2.1 Ciclo de ejecución



2.1 Procedimiento de Control

HECHOS \leftarrow Base de Hechos Inicial

Hasta que HECHOS satisfaga la condición de terminación o se ejecute una acción de parada

- **Selección de R** que pueda aplicarse a HECHOS
- HECHOS \leftarrow resultado de aplicar R a HECHOS

El ciclo termina cuando:

- No hay más reglas que se puedan disparar
- Una regla expresamente indica en su consecuente que acabe la ejecución

2.1 Procedimiento de Control

HECHOS \leftarrow Base de Hechos Inicial

Hasta que HECHOS satisfaga la condición de terminación o se ejecute una acción de parada

- **Selección de R** que pueda aplicarse a HECHOS
 - HECHOS \leftarrow resultado de aplicar R a HECHOS
-
- La eficiencia de la inferencia está estrechamente relacionada con el orden de selección de las reglas
 - Requisitos para ser una buena estrategia de control:
 - Causar movimiento
 - Ser sistemática
 - Ser eficiente

2.2 Modos de Inferencia

Inferir: obtener una consecuencia o deducir algo

- **Modus Ponens**

Si A entonces B

A

B

- **Modus Tollens** (inferencia contrapositiva)

Si A entonces B

$\neg B$

$\neg A$

2.3 Modos de Razonamiento

Razonar: Discurrir ordenando ideas para llegar a una conclusión:

- Encadenamiento hacia adelante
- Encadenamiento hacia atrás

2.3.1 Encadenamiento hacia adelante

- Hasta que el problema esté resuelto o no haya más reglas que aplicar:
 - **Seleccionar** aquellas reglas cuyo antecedente se cumple a partir de la base de hechos actual
 - Si hay más de una regla seleccionada, emplear una estrategia de **resolución de conflictos** que elimine todas las reglas anteriores menos una
 - **Ejecutar** la regla resultante del paso anterior

2.3.1 Encadenamiento hacia adelante

- Cuando se ejecuta una regla los nuevos hechos se añaden a la Base de Hechos
- Una vez disparada una regla puede eliminarse del sistema para evitar ciclos no deseados
- Existen diversas estrategias para ejecutar la próxima regla
 - Por orden de numeración
 - Por cantidad de condiciones
 - Por orden de aplicabilidad: la más recientemente usada p.e.
 - Etc.

Ejemplo 1

Base de Reglas

- **R1:** Si vuela y pone huevos entonces es un ave
- **R2:** Si un ave tiene patas largas, y color rosa entonces es un flamenco

Base de Hechos

- Especie 1 tiene color rosa
- Especie 1 vuela
- Especie 1 pone huevos
- Especie 1 tiene patas largas

Agenda

- R1

**Motor
de
Inferencia**

Ejemplo 1

Base de Reglas

- **R1:** Si vuela y pone huevos entonces es un ave
- **R2:** Si un ave tiene patas largas, y color rosa entonces es un flamenco

Agenda

- R2

Base de Hechos

- Especie 1 tiene color rosa
 - Especie 1 vuela
 - Especie 1 pone huevos
 - Especie 1 tiene patas largas
-
- ESPECIE1 ES UN AVE

Motor
de
Inferencia

Ejemplo 1

Base de Reglas

- **R1:** Si vuela y pone huevos entonces es un ave
- **R2:** Si un ave tiene patas largas, y color rosa entonces es un flamenco

Agenda

**Motor
de
Inferencia**

Base de Hechos

- Especie 1 tiene color rosa
- Especie 1 vuela
- Especie 1 pone huevos
- Especie 1 tiene patas largas

■ ESPECIE1 ES UN AVE

■ ESPECIE1 ES UN
FLAMENCO

Encadenamiento hacia adelante (Ejemplo 2)

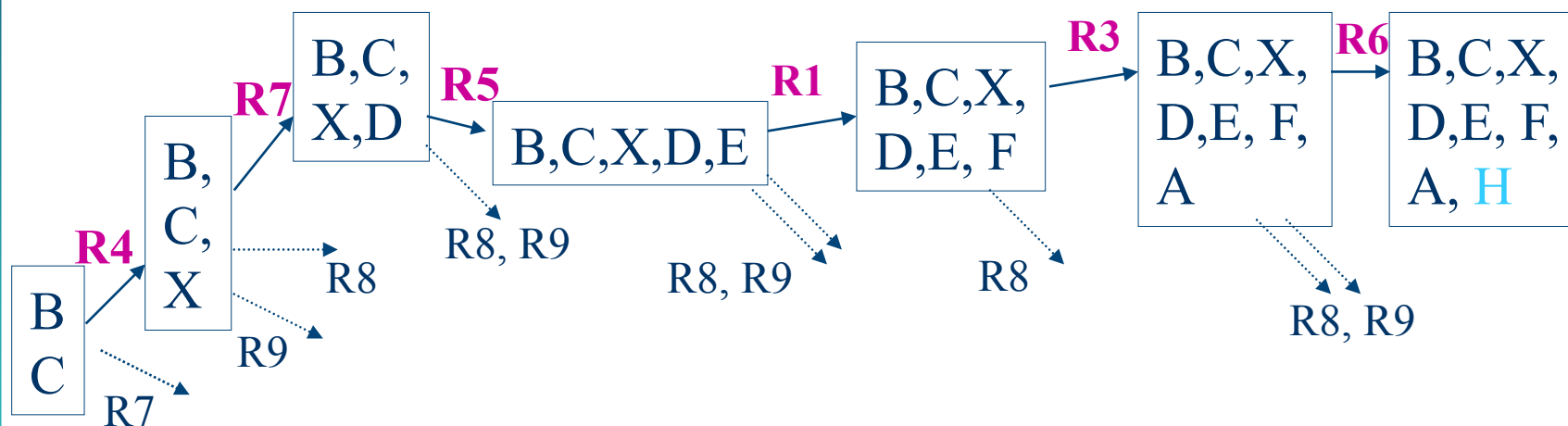
- Conjunto de Reglas

1. Si B y D y E	entonces	F
2. Si D y G	entonces	A
3. Si C y F	entonces	A
4. Si B	entonces	X
5. Si D	entonces	E
6. Si A y X	entonces	H
7. Si C	entonces	D
8. Si X y C	entonces	A
9. Si X y B	entonces	D

- Base de Hechos: **B, C**
- Meta: **H**

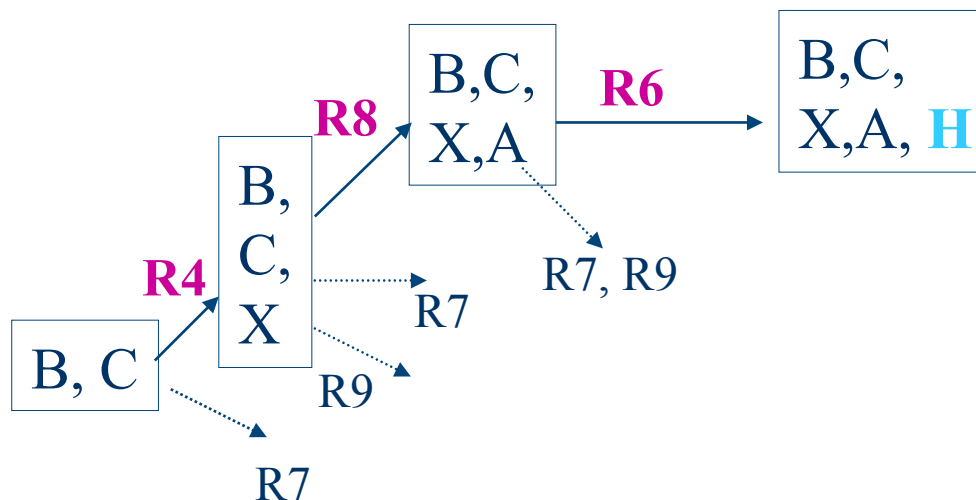
Encadenamiento hacia adelante (Ejemplo)

- Por orden de numeración



Encadenamiento hacia adelante (Ejemplo)

- Por cantidad de condiciones



2.3.2 Encadenamiento hacia atrás

- La regla

Si A entonces B

se utiliza para reemplazar la meta B, por un conjunto de submetas equivalentes expresadas por la premisa A.

- Hay que ir verificando todas las submetas generadas

A partir de una hipótesis intenta encontrar los hechos específicos que la avalan

Encadenamiento hacia atrás (Ejemplo)

- **Reglas:**

1. Si bomba_gasolina_OK y motor_arranca
 entonces problema_bujías
2. Si No motor_arranca y No luces_encienden
 entonces problema-batería
3. Si No motor_arranca y luces-encienden
 entonces problema-starter
4. Si gasolina_en_depósito
 entonces bomba_gasolina_OK

- **Hechos:**

gasolina_en_depósito motor_arranca

Encadenamiento hacia atrás (Ejemplo)

Objetivo: *Diagnosticar Avería de un Vehículo*

■ Inicio:

Paso 1: Regla 1 (problema_bujías)

Si bomba_gasolina_OK **y** motor_arranca

entonces problema_bujías

Verificación

- bomba_gasolina_OK
- motor_arranca

Paso 2: Regla 4 (bomba_gasolina_OK)

Si gasolina_en_depósito

entonces bomba_gasolina_OK

Verificación

- Gasolina_en_depósito

Encadenamiento hacia atrás (Ejemplo)

Paso 3: Base de Hechos

Verificado

✓ Gasolina_en_depósito OK

Verificado

✓ Bomba_gasolina OK

Paso 4: Base de Hechos

Motor_arranca OK

CONCLUSIÓN: PROBLEMA EN BUJÍAS

Hacia adelante v.s. Hacia atrás

- Encadenamiento hacia adelante
 - De los hechos hacia las conclusiones
 - Monitorización y control
 - Problemas dirigidos por los datos
 - Sin necesidad de explicación
- Encadenamiento hacia atrás
 - De los objetivos a los hechos
 - Diagnóstico
 - Problemas dirigidos por los objetivos
 - Interacción/Explicación al usuario

CLIPS (C Language Integrated Production System)

- Herramienta de desarrollo de Sistemas de Producción
- Encadenamiento hacia delante
- Intérprete que permite escribir comandos de manera interactiva, puede mostrar información sobre el estado actual de la memoria de trabajo y las reglas activadas en cada momento.



Ejemplo 1: Base de Hechos en CLIPS

Base de Hechos

- Especie 1 tiene color rosa
- Especie 1 vuela
- Especie 1 pone huevos
- Especie 1 tiene patas largas

f-0 (**Color** Especie1 Rosa)
f-1 (**Vuela** Especie1)
f-2 (**PoneHuevos** Especie1)
f-3 (**PatasLargas** Especie 1)

Formato de los hechos

(**<símbolo>** <datos>*)

El primer campo representa una relación entre los restantes
(**Color** Especie1 Rosa)

Ejemplo 1. Creación de la Base de Hechos: Deffacts

deffacts: para crear los hechos iniciales
(**deffacts** <nombre>
 <hecho>)

(**deffacts** iniciales

 (**Color** Especie1 Rosa)

 (**Vuela** Especie1)

 (**PoneHuevos** Especie1)

 (**PatasLargas** Especie1)

)

(**reset**)

(**facts**) ;; muestra los hechos existentes



f-1 (Color Especie1 Rosa)

f-2 (Vuela Especie1)

f-3 (PoneHuevos Especie1)

f-4 (PatasLargas Especie 1)

Ejemplo 1. Inserción de Hechos: Assert

Inserción de Nuevos Hechos en la Base de Hechos con la orden **assert**:

(**assert** <hecho>*)

(**assert** (**Color** Especie2 Blanco))

(**assert** (**PatatasCortas** Especie2))

(**facts**) ;;; muestra los hechos existentes

f-1 (Color Especie1 Rosa)

f-2 (Vuela Especie1)

f-3 (PoneHuevos Especie1)

f-4 (PatatasLargas Especie 1)

f-5 (Color Especie2 Blanco)

f-6 (PatatasCortas Especie2)

Ejemplo 1. Base de Conocimiento en Clips

(**defrule** <nom_regla>
 <condición>*
=> <acción>*)

Base de Reglas

R1: Si vuela y pone huevos
entonces es un ave

(**defrule** R1
 (Vuela ?x) (PoneHuevos ?x)
=>(assert (Ave ?x)))

R2: Si un ave tiene patas
largas, y color rosa entonces
es un flamenco

(**defrule** R2
 (PatasLargas ?x) (Color ?x Rosa)
=>(assert (Flamenco ?x)))

Resumiendo

- **defacts:** Creación de los hechos iniciales o “*permanentes*”
- **assert:** Creación de nuevos hechos durante el ciclo de ejecución

(**defacts** <nombre>
<hecho>*)

(**assert** <hecho>*)

- **defrule:** Definición de reglas

(**defrule** <nom_regla>
<condición>*
=> <acción>*)

Ejercicio



Diseña en CLIPS el siguiente sistema basado en reglas:

- R1: Si un animal tiene pelo, entonces es mamífero
- R2: Si un animal tiene plumas, entonces es ave
- R3: Si un animal come carne, es carnívoro
- R4: Si un mamífero carnívoro tiene color leonado con manchas oscuras se trata de un leopardo
- R5: Si un mamífero carnívoro tiene color leonado con rayas negras es un tigre

HECHOS

- Leo es un animal con pelo que come carne, tiene manchas oscuras y color leonado

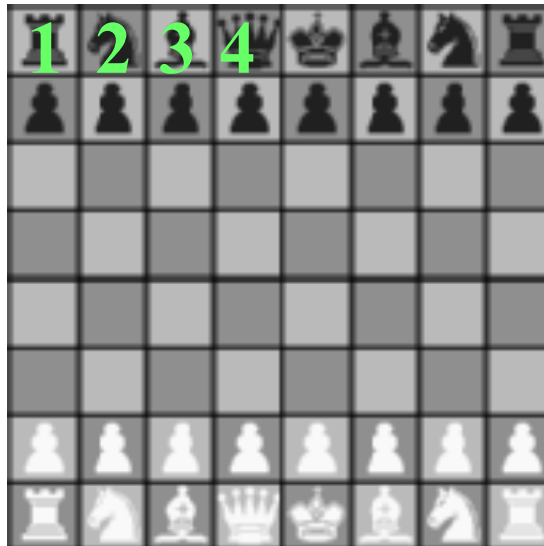
3. Técnicas de Equiparación

- La forma en que se extraen de un conjunto completo de reglas aquéllas que se aplican en un punto dado
- Se necesita algún tipo de equiparación entre el estado actual de la BH y las condiciones de las reglas
- La búsqueda a través de todo el conjunto de la BR puede ser muy ineficiente

3. Técnicas de Equiparación

- **Indexación:**

- Se utiliza el estado actual como índice dentro de las reglas para seleccionar inmediatamente la equiparada.
- Por ejemplo: en ajedrez



Almacena las reglas asociadas con una determinada posición con la misma clave

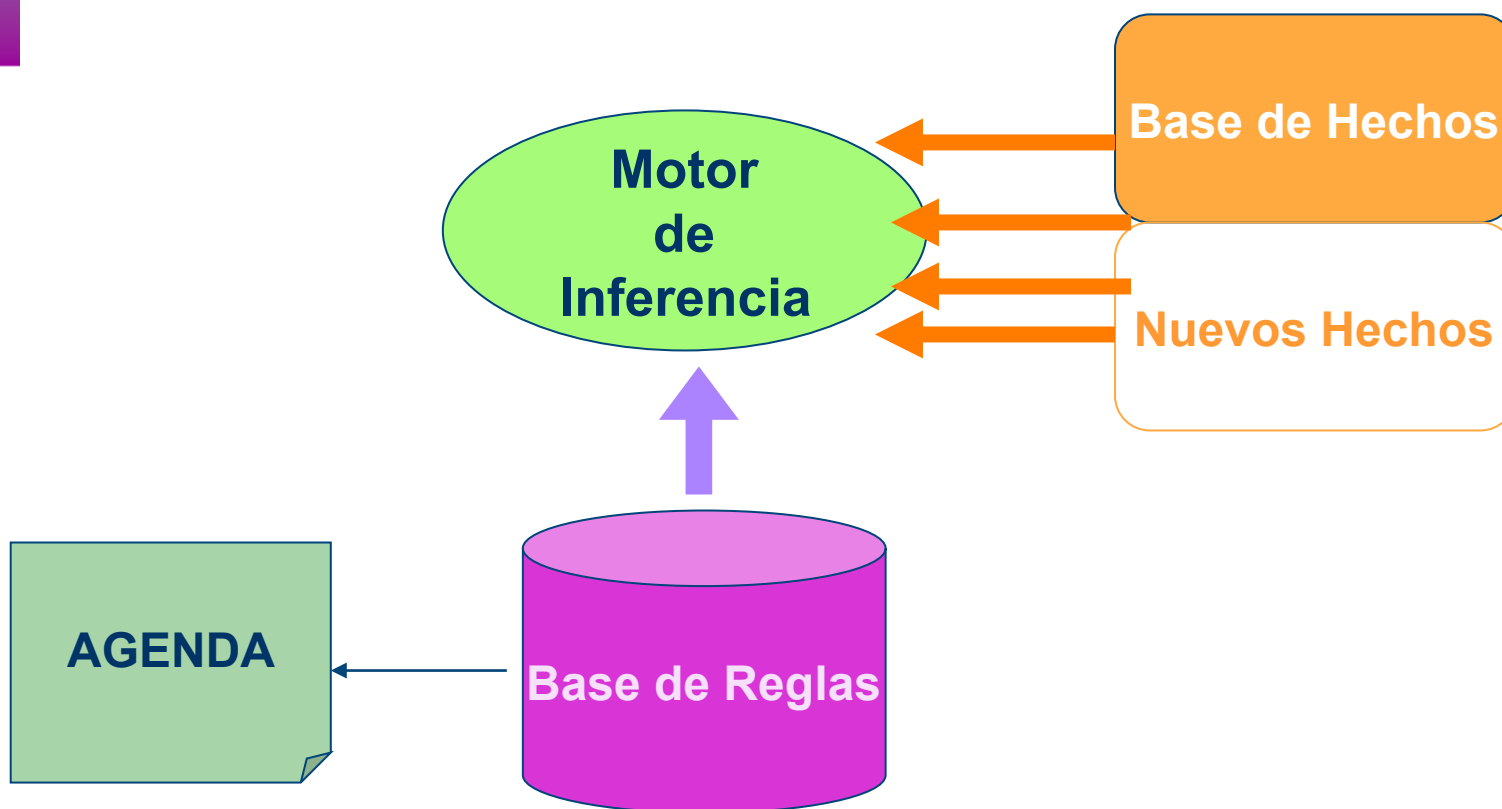
Pérdida de generalidad de las reglas

RETE (Charles Forgy)

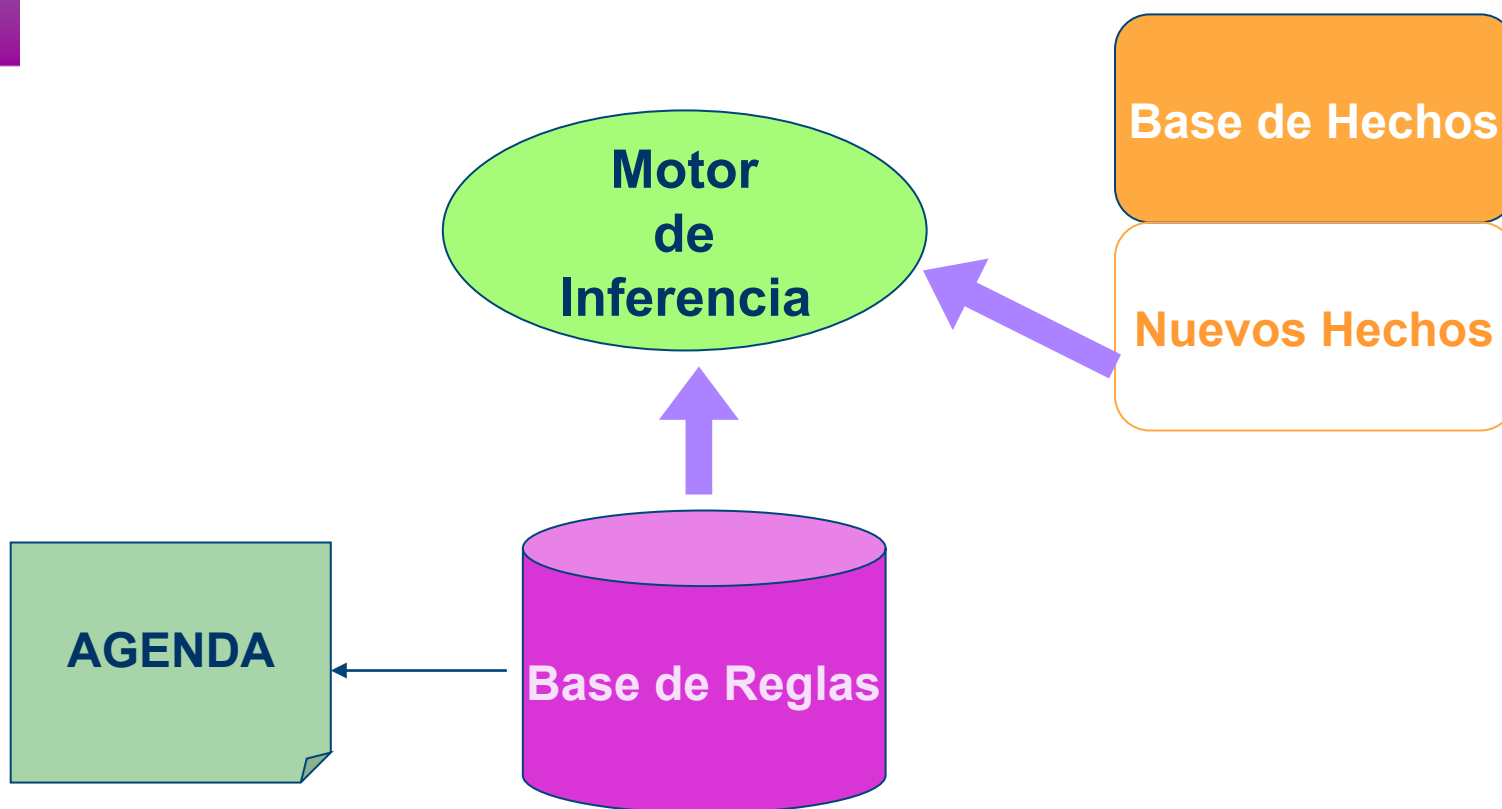
Red de Redundancia Temporal

- Búsqueda en Profundidad con Retroceso :
Encadenamiento Hacia Atrás
- **RETE** : Encadenamiento hacia delante
Construye una red en función de la BR
 - Mantiene información sobre conjunciones de antecedentes que se cumplen
 - Actualiza la red al inferir un nuevo hecho

Sistemas que no usan RETE



RETE



RETE

- Optimiza la construcción de la agenda en cada vuelta del ciclo de ejecución.
 - El disparo de una regla produce, generalmente, pocos cambios en la memoria de trabajo.
 - Se suele utilizar un mismo patrón de hecho en varias reglas.
- Construye y mantiene el grafo enraizado, dirigido y acíclico:
 - Nodos: Representan patrones de hechos (menos la raíz)
 - Caminos: Representan las condiciones de una regla (desde la raíz)

RETE

- Cada nodo contiene información acerca de los hechos que equiparan con los patrones de los nodos desde la raíz (incluido él mismo), junto con las asociaciones necesarias de las distintas variables.
- Los nodos de entrada a la red están asociados a los predicados que aparecen en los antecedentes de las reglas
- Cuando se **aserta** un nuevo hecho entra por el nodo que identifica al predicado

Ejemplo RETE

Para cada regla se construye la **Red de Patrones** y la **Red de Enlace**

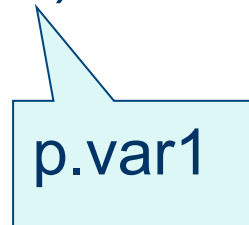
■ Base de Reglas

- R1: **Si** $p(?X)$ **y** $q(?X, ?Y)$
entonces $r(?X)$
- R2: **Si** $q(a, ?T)$ **y** $r(?T)$
entonces $s(?T)$
- R3: **Si** $p(?X)$ **y** $q(?X, ?L)$ **y** $s(?L)$
entonces $r(?L)$

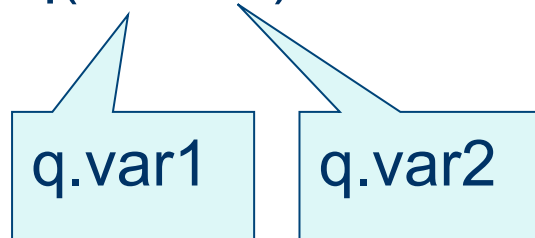
Rete

Cada argumento de un hecho se asocia a una variable, que se va numerando consecutivamente según su posición dentro del hecho (o la plantilla):

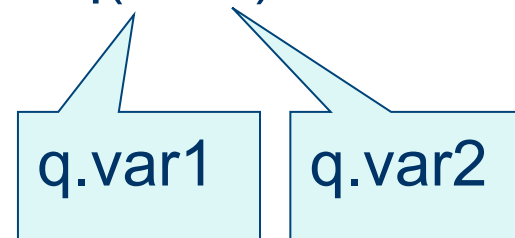
- $p(?X)$



- $q(?X, ?Y)$



- $q(a, ?T)$

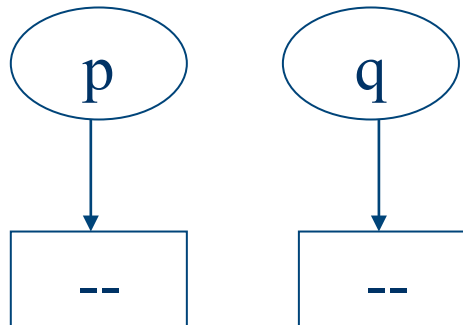


Ejemplo RETE: Construcción de la Red

- **Red de Patrones:** contiene nodos de una entrada con las restricciones que cada hecho establece.

R1: Si $p(?X)$ y $q(?X,?Y)$ entonces $r(?X)$

Red de
Patrones



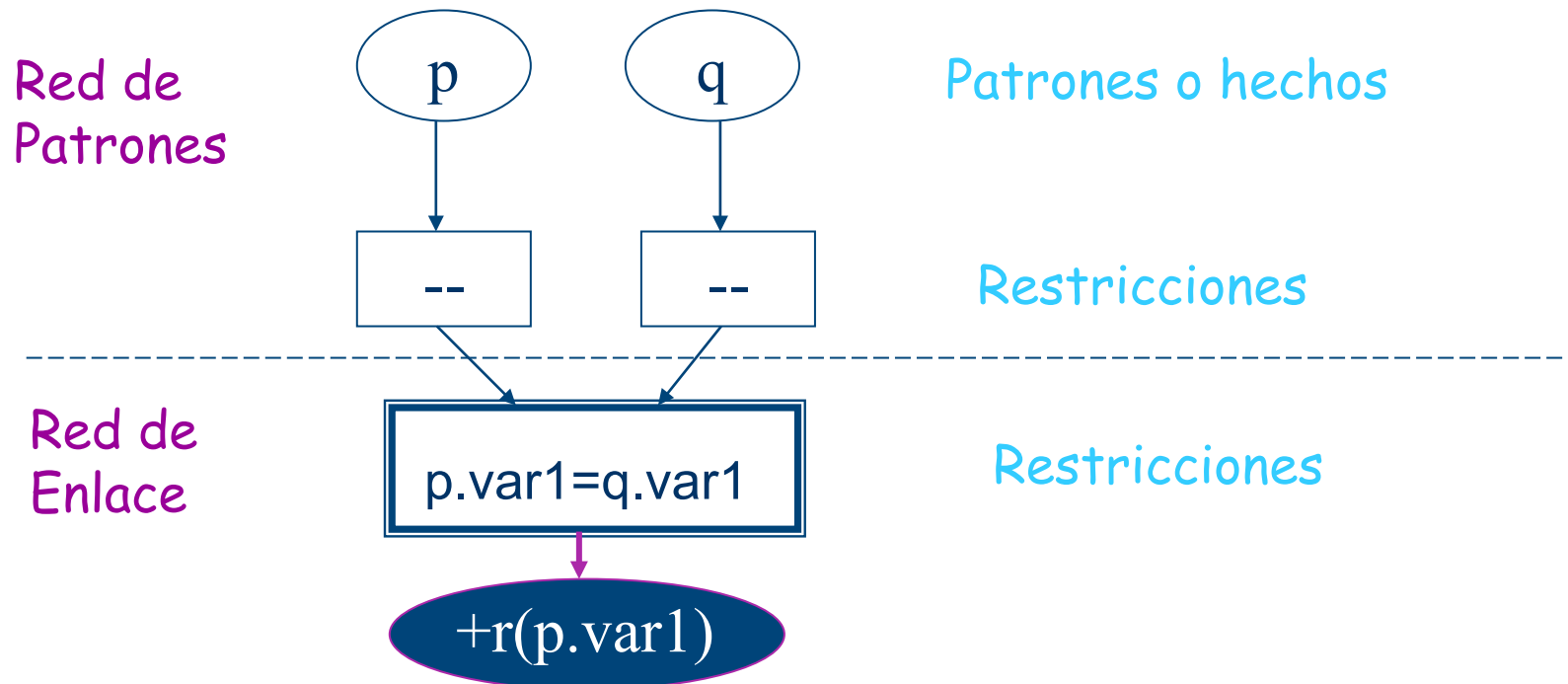
Patrones o hechos

Restricciones

Ejemplo RETE: Construcción de la Red

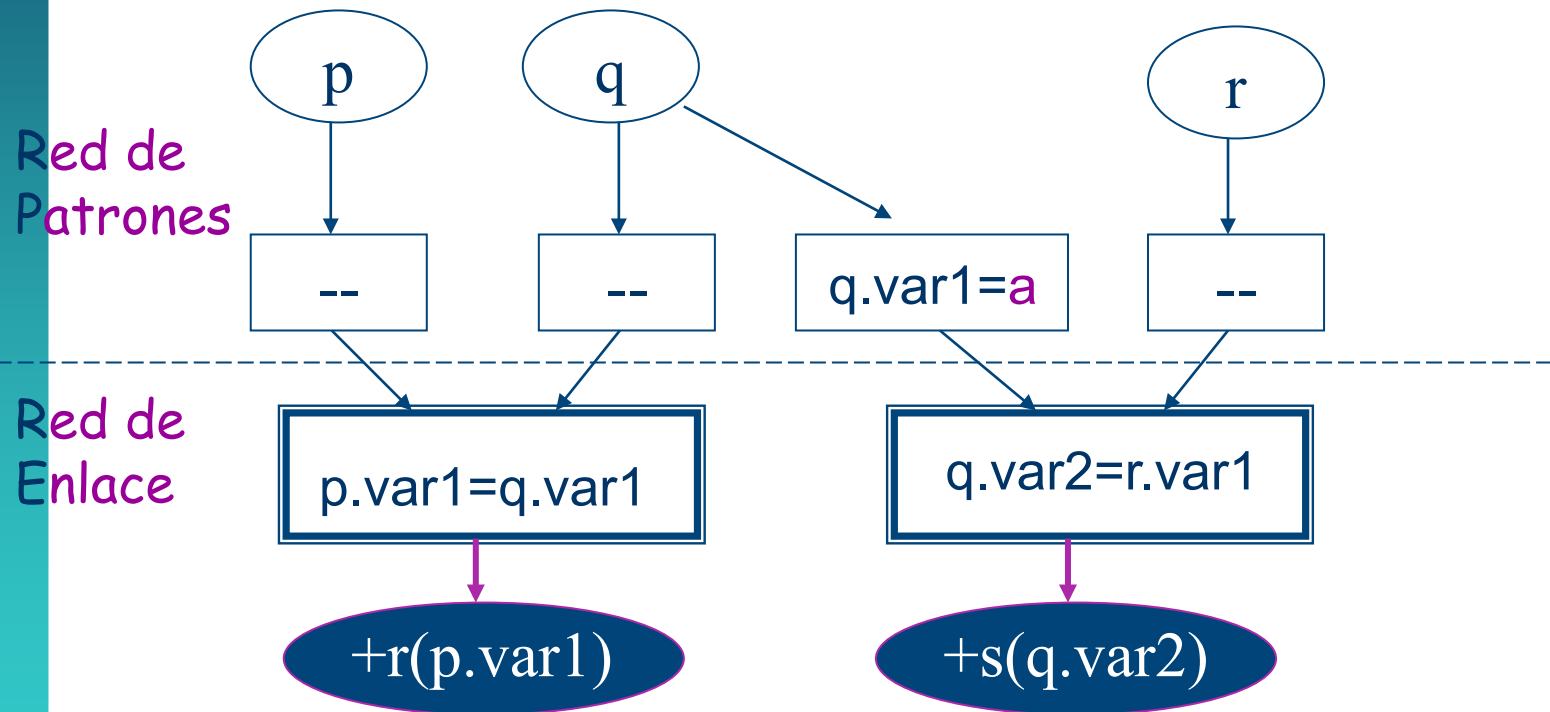
- **Red de Enlace:** contiene nodos de doble entrada que unen los patrones.

R1: Si $p(?X)$ y $q(?X,?Y)$ entonces $r(?X)$



Ejemplo RETE: Construcción de la Red

R2: Si $q(a, ?T)$ y $r(?T)$ entonces $s(?T)$

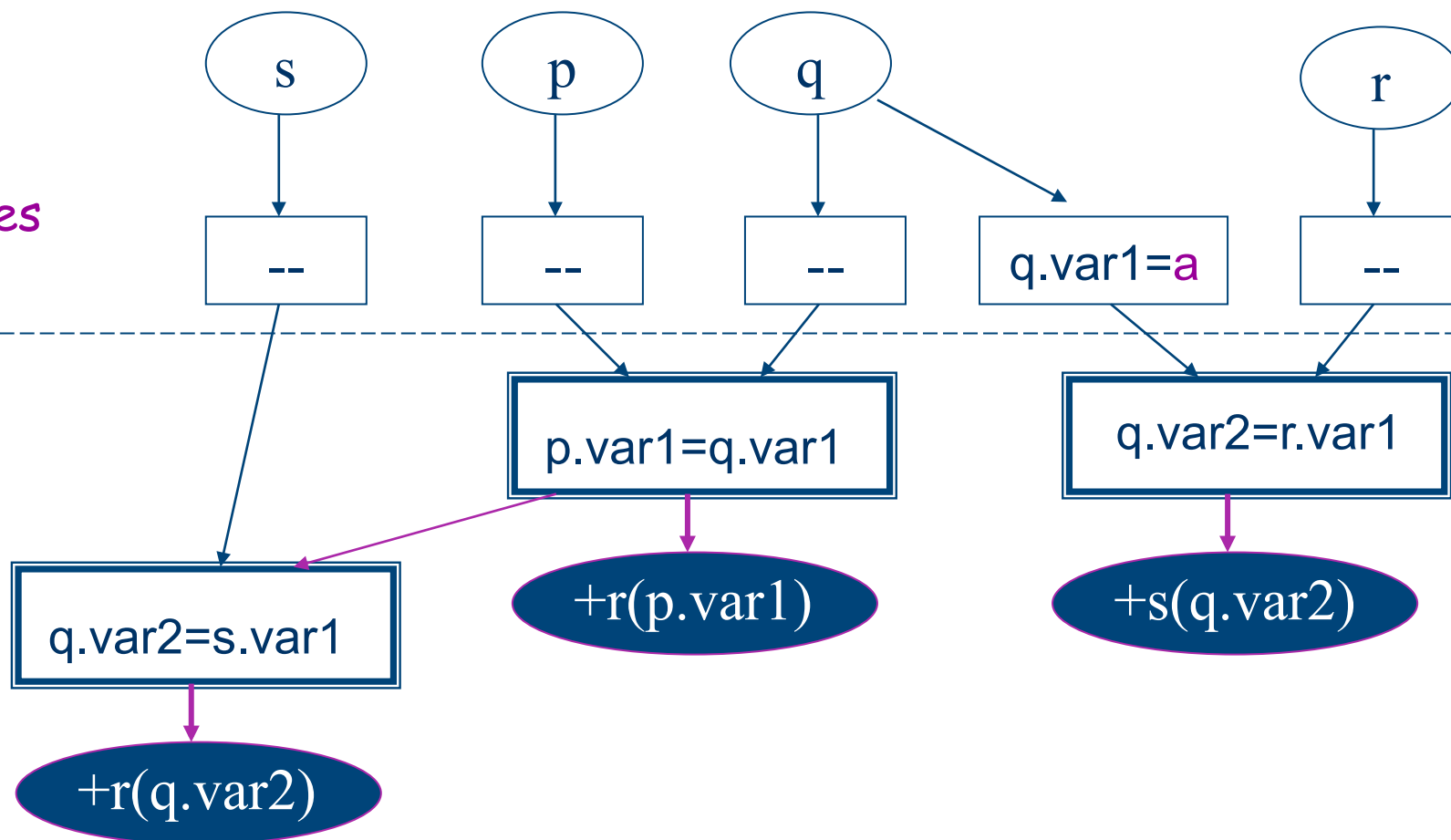


Ejemplo RETE: Construcción de la Red

R3: Si $p(?X)$ y $q(?X,?L)$ y $s(?L)$ entonces $r(?L)$

Red de Patrones

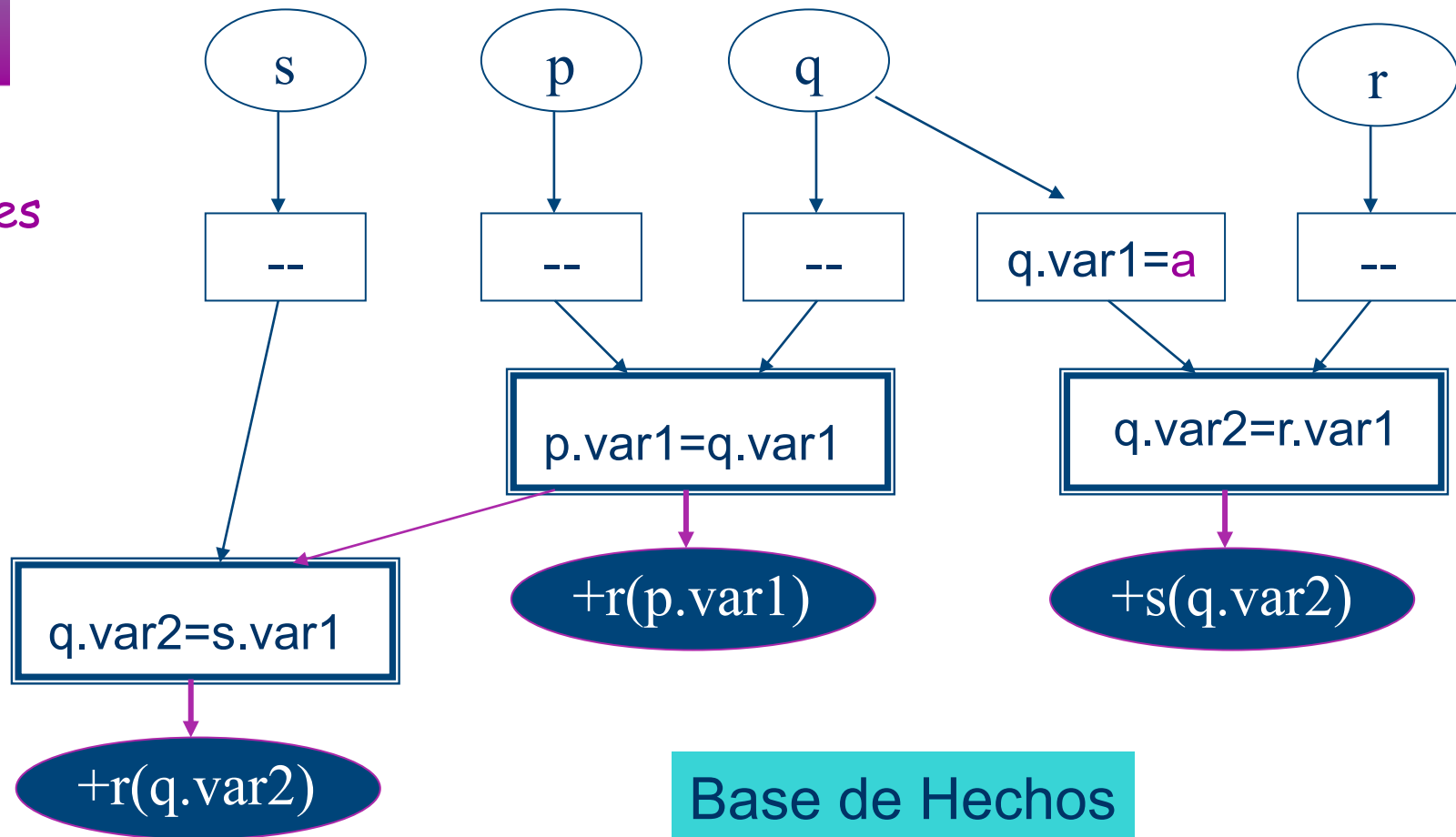
Red de Enlace



Ejemplo RETE: Uso de la Red

Red de Patrones

Red de Enlace



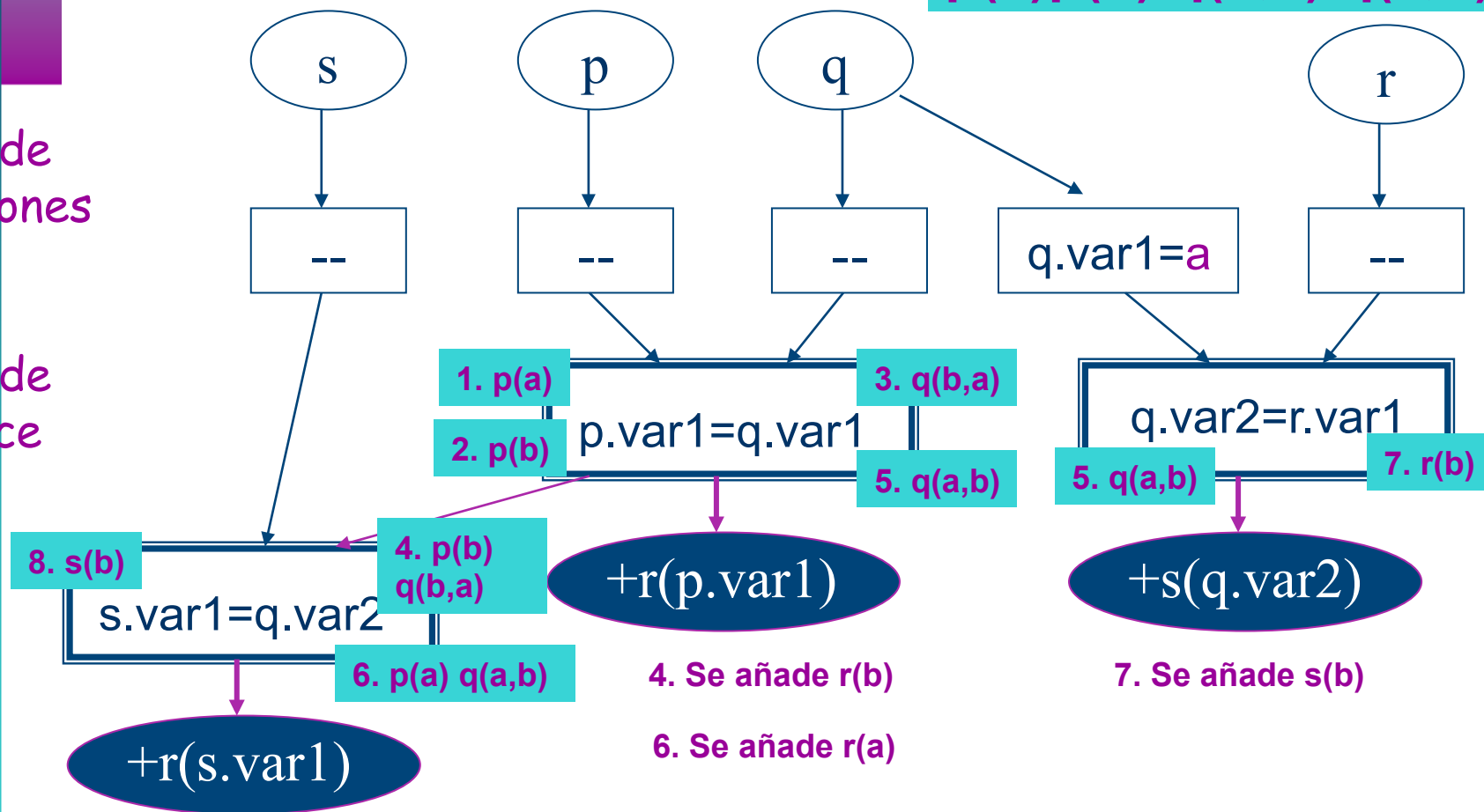
Base de Hechos
p(a) p(b)
q(b,a) q(a,b)

Ejemplo RETE: Uso de la Red

p(a)p(b) q(b,a) q(a,b)

Red de Patrones

Red de Enlace



4. Se añade r(b)

6. Se añade r(a)

7. Se añade s(b)

9. Se añade r(b) (ya está añadido)

Nuevos hechos

r(b) r(a) s(b)

Algoritmo RETE...

Desventajas

- Consume gran cantidad de memoria para incrementar velocidad de procesamiento

Ventajas

- Elimina la duplicidad en la reglas
- La fase de resolución de conflictos sirve para decidir cuál de las sugerencias se va a aceptar

4. Resolución de Conflictos

1. SI Válvula1 abierta ENTONCES activa Sensor 1
 2. SI Válvula1 cerrada ENTONCES activa Sensor 2
 3. SI Válvula1 cerrada ENTONCES cierra Válvula2
- **Conjunto conflicto:** Las reglas 2 y 3 se activarán al mismo tiempo ¿Qué regla ejecutar antes?

4. Resolución de Conflictos

- **Ordenación de Reglas:** se colocan primero las reglas que se desean examinar antes.
- **Control de las agendas:** las reglas listas para ser ejecutadas se colocan en una agenda en la que se ordenan según un valor de **prioridad** asignado a cada una de ellas.
- **Criterio de actualidad:** ejecutar aquellas reglas cuyo antecedente se apoya en información más reciente.
- **Criterio de especificidad:** reglas con más antecedentes primero.
- Mecanismo de **refracción:** no aplicar siempre la misma regla
- **Metarreglas**
- **Conjuntos de reglas:** para activar o desactivar varias reglas en bloque.

4. Resolución de Conflictos en CLIPS

(set-strategy <estrategia>)

- La prioridad se establece con:

(declare (salience xxx))

Además se puede establecer una de las siguientes estrategias de resolución de conflictos entre reglas de igual prioridad:

- **Depth Strategy** (estrategia por defecto): las nuevas activaciones se sitúan **por encima** de las activaciones con igual prioridad.
- **Breadth Strategy**: las antiguas primero: las nuevas activaciones **se sitúan por debajo** de las activaciones con igual prioridad.
- **Simplicity Strategy**: reglas con menos antecedentes primero.
- **Complexity Strategy**: reglas con más antecedentes primero.
- **LEX : novedad y especificidad**
- **MEA: novedad y especificidad** mirando sólo el primer hecho de cada regla

Ejercicio Propuesto

- Dado la siguiente Base de Reglas
 - Implementar en CLIPS el SBR
 - Construir la Red de Redundancia Temporal (RETE)
- Realizar la traza de ejecución correspondiente con la red construida, la base de hechos y aplicando las siguientes estrategias de Resolución de Conflictos:
 - Especificidad
 - Refracción

Ejemplo 1

Base de Reglas

- **R1:** Si vuela y pone huevos entonces es un ave
- **R2:** Si un ave tiene patas largas, y color rosa entonces es un flamenco
- **R3:** Si es carnívoro y tiene color amarillo y rayas negras entonces es un tigre

Base de Hechos

- Especie 1 tiene color rosa
- Especie1 vuela
- Especie 1 pone huevos
- Especie 1 tiene patas largas
- Fido tiene rayas negras
- Fido tiene color amarillo
- Fido es carnívoro

**Motor
de
Inferencia**

Bibliografía

1. **Borrajo D.**, Juristo N., Martínez V. y Pazos J.: *Inteligencia Artificial. Métodos y Técnicas*, Centro de Estudios Universitarios Ramón Areces, Madrid, 1993.
2. **Rich E.** y Knight K.: *Inteligencia Artificial*. McGraw-Hill, 1994
3. **Fernández S.**, González J. y Mira J. :*Problemas Resueltos de Inteligencia Artificial Aplicada. Búsqueda y Representación*. Pearson, Madrid, 2003.