

Técnicas de representación y razonamiento

❑ Tema 3: Representación del conocimiento e inferencia

❑ 3.4: Sistemas de producción – Índice de contenidos

❑ Arquitectura y funcionamiento de los sistemas de producción

- ❑ Memoria de trabajo. Representación de hechos
- ❑ Base de reglas. Representación de reglas
- ❑ Motor de inferencia. Ciclo de funcionamiento

❑ Ventajas e inconvenientes de los sistemas de producción

❑ El proceso de razonamiento

- ❑ Encadenamiento progresivo y encadenamiento regresivo

❑ Implementación

- ❑ Fase de reconocimiento. Algoritmo RETE
- ❑ Fase de selección. Estrategias de resolución de conflictos
- ❑ Sistemas de producción en JESS

Técnicas de representación y razonamiento

❑ Técnicas de representación del conocimiento

❑ Representaciones básicas

- ❑ Lógica de predicados. Representación en Prolog

❑ Sistemas de producción

- ❑ Redes semánticas

❑ Representaciones estructuradas

- ❑ Marcos (frames) y guiones (scripts)

❑ Estudio comparativo de las técnicas de representación

❑ Lenguajes de representación del conocimiento

Técnicas de representación y razonamiento

- ❑ Diversos formalismos para construir bases de conocimiento
 - ❑ Representaciones basadas en relaciones
 - ❑ Lógica
 - ❑ Redes semánticas
 - ❑ Representaciones basadas en objetos
 - ❑ Marcos
 - ❑ Objeto-Atributo-Valor
 - ❑ Representaciones basadas en acciones
 - ❑ Sistemas de producción
 - ❑ Guiones
 - ❑ Combinaciones y modificaciones de los anteriores

Porqué otro Formalismo : Dominios apropiados

- ❑ Dominio que se puede ver como conjunto de transiciones entre estados
 - ❑ Pocas interacciones entre los estados (bastante independientes)
 - ❑ Se puede ver como pasos (acciones) dados
 - ❑ Según sus prerequisites se toma una acción
 - ❑ Preferibles dominios estrechos, o separables en subáreas (más eficiencia)
- ❑ Dominio muy cambiante con el paso del tiempo
 - ❑ Hechos nuevos, otros obsoletos
 - ❑ Reglas (normas, leyes, principios) nuevas y obsoletas
- ❑ Dominio en que los expertos usan heurísticas en la toma de decisiones
 - ❑ Conocimiento dinámico: qué hacen ante una situación determinada
 - ❑ Habilidad para resolver un problema: estímulo - respuesta
- ❑ Dominio que tiene conocimiento
 - ❑ Incompleto: faltan detalles y/o subáreas enteras
 - ❑ No muy bien definido o experimental y/o con muchas condiciones
 - ❑ Con incertidumbre (factor de certeza)
- ❑ Si tiene fácil solución algorítmica y no es cambiante: usar algoritmos !!

Introducción: Reglas de Producción

- ❑ Patrones (situación-acción o premisas-conclusión)
 - ❑ Cómo reaccionar ante una situación
- ❑ Representación “declarativa” de **conocimiento procedimental**
SI condiciones-LHS ENTONCES acciones-RHS
- ❑ No son implicaciones lógicas sino recomendaciones imperativas

```
(defrule fallo1 "Fallo simple"
  (frigorífico luz encendida)
  (frigorífico puerta abierta)
=>
  (assert (frigorífico comida estropeada) )
  (printout t "tirar comida, limpiar frigorífico" ) )
(defrule fallo2 "Fallo temperatura"
  (temp ?x)
  (temp-ext ?y)
  (test (>= (abs (- ?y ?x)) 10))
=>
  (printout t "El frigo ha perdido gas" crlf) )
```

} LHS

} RHS

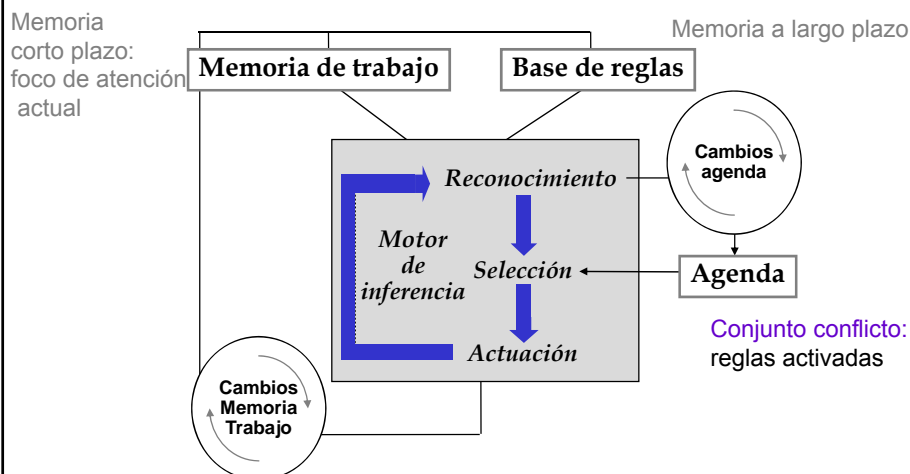
Arquitectura de un sistema de Reglas

- (conocimiento)
- ❑ Base de reglas (o producciones, –BR–)
Propiedades operativas/dinámicas del Dominio
 - ❑ Base de hechos (o Memoria de Trabajo, –MT–)
Propiedades del estado actual o del contexto del problema
- PROGRAMA (el cómo)
- ❑ Intérprete de reglas (motor de inferencia → estrategia de control)
 - ❑ Usa como entrada B. Reglas y MT
 - ❑ Ejecuta el Ciclo de Operación

Motor de Inferencia: Ciclo de Ejecución

1. **Reconocimiento o equiparación** (*matching*): LHS
 - ❑ Qué reglas son aplicables: crea el **conjunto conflicto** o **agenda**
 2. **Selección o resolución de conflictos**: Qué regla se ejecuta ahora?
 - ❑ Lo decide de acuerdo a la estrategia de control
 3. **Actuación**: se ejecuta la regla, realizando las acciones del RHS
 - ❑ Añadir/quitar hechos de MT y operaciones externas al sistema reglas
 4. **Repite el ciclo**: Inicializa la agenda y vuelve al punto 1.
 - ❑ Hasta que se alcanza el objetivo
- ❑ Permite cambiar la estrategia de control
- ❑ Resuelve conflictos cuando hay varias reglas en la agenda
 - ❑ Qué regla debo activar de todas las de la agenda?
 - ❑ La estrategia se define al diseñar el sistema (vemos más tarde)
 - ❑ Ej: (set-strategy depth) : activa ahora la más reciente activada

Diagrama Funcionamiento



Memoria de trabajo o Base de Hechos (MT)

- Representa el estado actual (contexto del problema)
 - Estado inicial: situación inicial u origen del problema (def facts...)
 - Estados finales: situaciones objetivo
 - Ej: estado después de obtener la respuesta a una consulta
 - Estados intermedios: (incluyen el estado actual)
 - situaciones por las que pasa en el proceso de resolución
- La MT es dinámica: se modifica en cada ciclo de inferencia
 - La ejecución de reglas añade o elimina hechos en ella
- Representación de hechos: *Pepe nació en Zaragoza en 1966*
 - CLIPS / JESS: (template es como una clase, assert = instancia)


```
(deftemplate Persona (slot Nombre) (slot FechaNacimiento)
                      (slot LugarNacimiento))
(assert (Persona (Nombre Pepe) (FechaNacimiento 1966)
                (LugarNacimiento Zaragoza)))
```
 - Prolog: `persona(pepe, 1966, zaragoza).`

Base de reglas

- ❑ Conocimiento procedimental sobre cómo solucionar problemas
- ❑ Formato uniforme: **SI condiciones-LHS ENTONCES acciones-RHS**
 - ❑ Condiciones: lista de hechos a verificar en la MT (patrones)
 - ❑ Acciones:
 - ❑ Cambios a realizar sobre la MT
 - ❑ Actividades externas al sistema (imprimir, órdenes a dispositivos)
 - ❑ **No** son reglas las estructuras del tipo *if-then-else*
- ❑ Las reglas son independientes unas de otras
 - ❑ Una regla **no** puede hacer referencia a otra (no están acopladas)
 - ❑ La única comunicación entre reglas es a través de la MT
 - ❑ Facilita añadir, modificar o eliminar reglas de la BR
 - ❑ Flexibilidad/modularidad versus ineficiencia
 - ❑ Cada regla modifica poco la MT
 - ❑ Existen técnicas de *chunking* para agrupar reglas
 - ❑ Varias Bases de reglas, cada una sobre un área del dominio

Ejemplo de regla en "pseudo-código"

- Si la edad del paciente es inferior a 10 años, y tiene manchas rojas y fiebre entonces tiene varicela
 - SI (Paciente ?p ?edad) and (?edad < 10) and (Síntomas ?p fiebre) and (Síntomas ?p manchas-rojas)
 - ENTONCES Añadir (Enfermedad ?p varicela)
- Constantes:
 - Paciente,
 - Síntomas,
 - Enfermedad,
 - 10,
 - fiebre,
 - manchas-rojas,
 - Varicela
- Variables:
 - ?p
 - ?edad
- Una regla resuelve un trozo del problema: un paso

Ejemplo sencillo: ordenación de cadenas

- Sistema que ordena las letras de una cadena de $\{a, b, c\}^*$
 - Conjunto de reglas
 - 1) $ba \rightarrow ab$
 - 2) $ca \rightarrow ac$
 - 3) $cb \rightarrow bc$
 - Estrategia
 - c. conflicto:
 - 1ª regla

Ejemplo de traza

iteración	MT	C.conflicto	R.disparada
0	cbaca	1, 2, 3	1
1	cabca	2	2
2	acbca	2, 3	2
3	acbac	1, 3	1
4	acabc	2	2
5	aacbc	3	3
6	aabcc	Ø	Halt

- Los sistemas de reglas son un modelo de cómputo general
 - misma potencia de cómputo que las máquinas de Turing

Ejemplo Tipos Razonamiento: progresivo y regresivo

- Las reglas pueden utilizarse de dos formas:

(P1) $\$ \rightarrow a\a

(P2) $\$ \rightarrow b\b

(P3) $\$ \rightarrow c\c

“\$” encaja con cualquier cadena

- Encadenamiento hacia adelante:

- Utilizar las reglas para generar palíndromos
- Dado un símbolo inicial como *c*, la secuencia de reglas (P1), (P2), (P3), (P2), (P3) generará la siguiente secuencia de cadenas:
aca bacab cbacabc bcbacabcb cbcbacabc

- Encadenamiento hacia atrás:

- Utilizar las reglas para reconocer palíndromos
 - Dado un palíndromo como *bacab*, podemos trazar la secuencia de reglas que llevan a su construcción
 - bacab* satisface la parte derecha de (P2), cuya parte izquierda da *aca*
 - aca* satisface la parte derecha de (P1), cuya parte izquierda da *c*
- Respuesta : SI, es un palíndromo

IAIC – Curso 2010-11

Tema 3.5 - 13

Tipos Razonamiento : Encadenamientos

- Encadenamiento progresivo (hacia adelante, forward chaining)

- Dirigido por datos (antecedentes-LHS)
 - Desde los datos (hechos) conocidos busca la conclusión
 - Si los datos verifican las condiciones (LHS) de una regla,
 - Entonces la regla se puede aplicar/activar/disparar/ejecutar

- Encadenamiento regresivo (hacia atrás, backward chaining)

- Dirigido por objetivos (consecuentes-conclusiones-RHS):
 - Selecciona conclusiones (RHS) posibles:
 - De reglas cuyos RHS equiparen a la consulta
 - Intenta probar su validez buscando evidencias (LHS) que la soporten:
 - Un antecedente-LHS es cierto si sus hechos:
 - Está en la MT del sistema
 - Sino, se busca si es consecuente (RHS) de alguna regla R:
 - Se prueban recursivamente los antecedentes de dicha regla R
 - Si no son aplicables a) y b), se asume la hipótesis del mundo cerrado

IAIC – Curso 2010-11

Tema 3.5 - 14

Ejercicio de encadenamiento de reglas

- ❑ Dado un sistema basado en reglas con la siguiente base de conocimiento
 - ❑ R1: **Si** h_2 y h_5 **entonces** h_1
 - ❑ R2: **Si** h_4 y h_3 **entonces** h_2
 - ❑ R3: **Si** h_6 **entonces** h_3

donde cada h_i representa una situación o concepto y los números al lado de las reglas marcan la prioridad de ejecución de las mismas en caso de conflicto
- ❑ La base de hechos inicial contiene los siguientes datos:

h_6, h_7, h_9, h_8, h_4 y h_5

 - a) Aplica encadenamiento hacia delante, mostrando cómo evoluciona el sistema en cada ciclo del proceso
 - b) Aplicando encadenamiento hacia atrás, determina si es posible establecer la existencia de la situación h_1 a partir de la base de hechos inicial

IAIC – Curso 2010-11

Tema 3.5 - 15

a) Solución ejercicio: encadenamiento hacia adelante

- ❑ 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

Explicación

Inicialmente se tendría BH₀, de las tres reglas existentes en la base de conocimientos sólo la 3 se selecciona. Tras su ejecución, se obtiene BH₁. Ahora existirá un conflicto entre las reglas 2 y 3. 2 es más prioritaria y 3 ya se ha aplicado. Se ejecuta 2 obteniendo BH₂. En esta situación todas las reglas forman parte de la agenda. Finalmente, es la regla 1 la que se ejecuta, quedando BH₃.

BH ₀	BH ₁	BH ₂	BH ₃
h_6	h_6	h_6	h_6
h_7	h_7	h_7	h_7
h_9	h_9	h_9	h_9
h_8	h_8	h_8	h_8
h_4	h_4	h_4	h_4
h_5	h_5	h_5	h_5
	h_3	h_3	h_3
		h_2	h_2
			h_1

IAIC – Curso 2010-11

Tema 3.5 - 16

b) Solución ejercicio: encadenamiento hacia atrás

- ☐ Se parte de un concepto objetivo a verificar a partir de la BR y la BH.
- ☐ Comprobar si dicho concepto pertenecía ya o no a la BH. En caso negativo recurrir a las reglas en la BR en cuyo consecuente figure el concepto objetivo. Así los conceptos del antecedente de dichas reglas pasan a ser subobjetivos. Si se demuestra la validez de estos subobjetivos, podrá inferirse el concepto objetivo global.
- ☐ En el ejemplo el objetivo global es *h1*, que no figura en la BH inicial;
- ☐ Ir a la BC. La regla 1 tiene *h1* en su consecuente. Ello nos permite fijar *h2* y *h5* como subobjetivos. Al encontrarse *h5* en la BH inicial, sólo quedaría por demostrar *h2*.
- ☐ La regla 2, con *h2* en su consecuente, establece *h4* y *h3* como nuevos subobjetivos. El hecho *h4* está en la BH inicial. Habrá que verificar, la existencia de *h3*.
- ☐ La regla 3 fija como subobjetivo *h6*, que está en la BH inicial, por tanto **final**.
- ☐ Entonces *h1* es válido aplicando las reglas 3, 2 y 1 a la BH inicial.

IAIC – Curso 2010-11

Tema 3.5 - 17

Criterios de elección de dirección del encadenamiento

- ☐ Dado el problema, ir de menos cantidad hacia más:
 - ☐ Complejidad del LHS y del RHS de las reglas
 - ☐ Empezar por el menor número de hechos (inicial u objetivo)
 - ☐ Aspecto del espacio de estados
 - ☐ Naturaleza y disponibilidad de los datos del problema
 - ☐ Factor de ramificación en ambas direcciones
 - ☐ Atrás: Muchos datos iniciales (hechos), pocas posibles conclusiones
 - ☐ Adelante: Pocos datos iniciales y/o muchas posibles conclusiones
- ☐ Si el proceso de resolución es interactivo
 - ☐ Utilizar el mismo que el experto usa de forma natural
- ☐ Sistemas de contestación de preguntas
 - ☐ Utilizar hacia atrás (*p.ej. sistemas de diagnóstico*)
- ☐ Sistemas que reaccionan a datos que llegan periódicamente
 - ☐ Utilizar hacia delante
 - ☐ Ej: datos sensores cada minuto en una central térmica

IAIC – Curso 2010-11

Tema 3.5 - 18

Ejemplo: reglas de identificación de frutas

- R1: SI Forma = alargada Y Color = verde o amarillo ENTONCES Fruta = banana
 R2: SI Forma = redonda u ovalada Y Diámetro > 1.6 cm ENTONCES claseFruta = planta
 R3: SI Forma = redonda Y Diámetro < 1.6 cm ENTONCES claseFruta = árbol
 R4: SI numSemillas = 1 ENTONCES claseSemilla = hueso
 R5: SI numSemillas > 1 ENTONCES claseSemilla = múltiple
 R6: SI claseFruta = planta Y Color = verde ENTONCES Fruta = sandía
 R7: SI claseFruta = planta Y Color = amarillo ENTONCES Fruta = melón
 R8: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = albaricoque
 R9: SI claseFruta = árbol Y Color = naranja Y claseSemilla = múltiple ENTONCES Fruta = naranja
 R10: SI claseFruta = árbol Y Color = rojo Y claseSemilla = hueso ENTONCES Fruta = cereza
 R11: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = melocotón
 R12: SI claseFruta = árbol Y Color = rojo o amarillo o verde Y claseSemilla = múltiple ENTONCES Fruta = manzana
 R13: SI claseFruta = árbol Y Color = morado Y claseSemilla = hueso ENTONCES Fruta = ciruela

IAIC – Curso 2010-11

Tema 3.5 - 19

Intérprete con encadenamiento progresivo

- ❑ Pasos del intérprete:
 - 1) **Reconocimiento, equiparación o encaje:** encuentra reglas aplicables y las activa
 - 2) **Resolución de conflictos:** desactiva reglas que no añadan hechos nuevos, y selecciona la **primera regla aplicable** (la de menor número en este ejemplo)
 - 3) **Actuación:** ejecuta la acción de la regla (la dispara). Si no hay, se detiene
 - 4) **Reset:** vacía la agenda (desactiva reglas aplicables) y vuelve al paso 1)

- ❑ Si la memoria de trabajo tiene los siguientes hechos iniciales:

Diámetro = 0.4 cm, Forma = redonda, Numsemillas = 1, Color = rojo

Ciclo de ejecución	Reglas aplicables	Regla seleccionada	Hecho derivado
1			

1

- ❑ Ésta es sólo una de las posibles estrategias de selección (control)

IAIC – Curso 2010-11

Tema 3.5 - 20

Ejemplo: reglas de identificación de frutas

- R1: SI Forma = alargada Y Color = verde o amarillo ENTONCES Fruta = banana
- R2: SI Forma = redonda u ovalada Y Diámetro > 1.6 cm ENTONCES claseFruta = planta
- R3: SI Forma = redonda Y Diámetro < 1.6 cm ENTONCES claseFruta = árbol
- R4: SI numSemillas = 1 ENTONCES claseSemilla = hueso
- R5: SI numSemillas > 1 ENTONCES claseSemilla = múltiple
- R6: SI claseFruta = planta Y Color = verde ENTONCES Fruta = sandía
- R7: SI claseFruta = planta Y Color = amarillo ENTONCES Fruta = melón
- R8: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = albaricoque
- R9: SI claseFruta = árbol Y Color = naranja Y claseSemilla = múltiple ENTONCES Fruta = naranja
- R10: SI claseFruta = árbol Y Color = rojo Y claseSemilla = hueso ENTONCES Fruta = cereza
- R11: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = melocotón
- R12: SI claseFruta = árbol Y Color = rojo o amarillo o verde Y claseSemilla = múltiple ENTONCES Fruta = manzana
- R13: SI claseFruta = árbol Y Color = morado Y claseSemilla = hueso ENTONCES Fruta = ciruela

Diámetro = 0.4 cm,
Forma = redonda,
Numsemillas = 1,
Color = rojo

IAIC – Curso 2010-11

Tema 3.5 - 21

Intérprete con encadenamiento progresivo (adelante)

□ Pasos del intérprete:

- 1) **Reconocimiento, equiparación o encaje:** encuentra reglas aplicables y las activa
- 2) **Resolución de conflictos:** desactiva reglas que no añadan hechos nuevos, y selecciona la **primera regla aplicable** (la de menor número en este ej.)
- 3) **Actuación:** ejecuta la acción de la regla (la dispara). Si no hay, se detiene
- 4) **Reset:** vacía la agenda (desactiva reglas aplicables) y vuelve al paso 1)

□ Si la memoria de trabajo tiene los siguientes hechos iniciales:

Diámetro = 0.4 cm, Forma = redonda, Numsemillas = 1, Color = rojo

Ciclo de ejecución	Reglas aplicables	Regla seleccionada	Hecho derivado
1	3, 4	3	claseFruta = árbol
2			

IAIC – Curso 2010-11

Tema 3.5 - 22

Ejemplo: reglas de identificación de frutas

- R1: SI Forma = alargada Y Color = verde o amarillo ENTONCES Fruta = banana
- R2: SI Forma = redonda u ovalada Y Diámetro > 1.6 cm ENTONCES claseFruta = planta
- R3: SI Forma = redonda Y Diámetro < 1.6 cm ENTONCES claseFruta = árbol
- R4: SI numSemillas = 1 ENTONCES claseSemilla = hueso
- R5: SI numSemillas > 1 ENTONCES claseSemilla = múltiple
- R6: SI claseFruta = planta Y Color = verde ENTONCES Fruta = sandía
- R7: SI claseFruta = planta Y Color = amarillo ENTONCES Fruta = melón
- R8: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = albaricoque
- R9: SI claseFruta = árbol Y Color = naranja Y claseSemilla = múltiple ENTONCES Fruta = naranja
- R10: SI claseFruta = árbol Y Color = rojo Y claseSemilla = hueso ENTONCES Fruta = cereza
- R11: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = melocotón
- R12: SI claseFruta = árbol Y Color = rojo o amarillo o verde Y claseSemilla = múltiple ENTONCES Fruta = manzana
- R13: SI claseFruta = árbol Y Color = morado Y claseSemilla = hueso ENTONCES Fruta = ciruela

Diámetro = 0.4 cm,
Forma = redonda,
Numsemillas = 1,
Color = rojo,
claseFruta = árbol

IAIC – Curso 2010-11

Tema 3.5 - 23

Intérprete con encadenamiento progresivo

□ Pasos del intérprete:

- 1) **Reconocimiento, equiparación o encaje:** encuentra reglas aplicables y las activa
- 2) **Resolución de conflictos:** desactiva reglas que no añadan hechos nuevos, y selecciona la **primera regla aplicable** (la de menor número en este ej.)
- 3) **Actuación:** ejecuta la acción de la regla (la dispara). Si no hay, se detiene
- 4) **Reset:** vacía la agenda (desactiva reglas aplicables) y vuelve al paso 1)

□ Si la memoria de trabajo tiene los siguientes hechos iniciales:

Diámetro = 0.4 cm, Forma = redonda, Numsemillas = 1, Color = rojo

Ciclo de ejecución	Reglas aplicables	Regla seleccionada	Hecho derivado
1	3, 4	3	claseFruta = árbol
2	4	4	claseSemilla = hueso
3			

Desactivada por el principio de refracción

IAIC – Curso 2010-11

Tema 3.5 - 24

Ejemplo: reglas de identificación de frutas

- R1: SI Forma = alargada Y Color = verde o amarillo ENTONCES Fruta = banana
- R2: SI Forma = redonda u ovalada Y Diámetro > 1.6 cm ENTONCES claseFruta = planta
- R3: SI Forma = redonda Y Diámetro < 1.6 cm ENTONCES claseFruta = árbol
- R4: SI numSemillas = 1 ENTONCES claseSemilla = hueso
- R5: SI numSemillas > 1 ENTONCES claseSemilla = múltiple
- R6: SI claseFruta = planta Y Color = verde ENTONCES Fruta = sandía
- R7: SI claseFruta = planta Y Color = amarillo ENTONCES Fruta = melón
- R8: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = albaricoque
- R9: SI claseFruta = árbol Y Color = naranja Y claseSemilla = múltiple ENTONCES Fruta = naranja
- R10: SI claseFruta = árbol Y Color = rojo Y claseSemilla = hueso ENTONCES Fruta = cereza
- R11: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = melocotón
- R12: SI claseFruta = árbol Y Color = rojo o amarillo o verde Y claseSemilla = múltiple ENTONCES Fruta = manzana
- R13: SI claseFruta = árbol Y Color = morado Y claseSemilla = hueso ENTONCES Fruta = ciruela

...
Color = rojo,
claseFruta = árbol,
claseSemilla = hueso

IAIC – Curso 2010-11

Tema 3.5 - 25

Intérprete con encadenamiento progresivo

□ Pasos del intérprete:

- 1) **Reconocimiento, equiparación o encaje:** encuentra reglas aplicables y las activa
- 2) **Resolución de conflictos:** desactiva reglas que no añadan hechos nuevos, y selecciona la **primera regla aplicable** (la de menor número en este ej.)
- 3) **Actuación:** ejecuta la acción de la regla (la dispara). Si no hay, se detiene
- 4) **Reset:** vacía la agenda (desactiva reglas aplicables) y vuelve al paso 1)

□ Si la memoria de trabajo tiene los siguientes hechos iniciales:

Diámetro = 0.4 cm, Forma = redonda, Numsemillas = 1, Color = rojo

Ciclo de ejecución	Reglas aplicables	Regla seleccionada	Hecho derivado
1	3, 4	3	claseFruta = árbol
2	4	4	claseSemilla = hueso
3	10	10	Fruta = cereza
4			

Ya no hay más reglas en la agenda – FIN

Conclusión: la fruta es una cereza

IAIC – Curso 2010-11

Tema 3.5 - 26

Intérprete con encadenamiento regresivo (atrás)

❑ Pasos de un intérprete con encadenamiento regresivo

- 1) Formar una pila con todos los objetivos iniciales
- 2) Reunir todas las reglas capaces de satisfacer el 1º objetivo
- 3) Para cada una de estas reglas, **examinar sus premisas**:

orden ↗ a) Si las premisas son satisfechas, ejecutar la regla para derivar sus conclusiones. Eliminar el objetivo de la pila y volver al paso 2)

b) Si una de las premisas no se cumple, buscar las reglas que pueden derivar esta premisa. Si se encuentra alguna, se considera la premisa como subobjetivo, se coloca al principio de la pila, y se va al paso 2)



c) Si el paso b) no puede encontrar una regla, entonces **preguntar al usuario** por el valor del parámetro, y añadir éste a la memoria de trabajo. Si este valor satisface la premisa en curso, continuar con la siguiente premisa de esta regla. Si no, continuar con la siguiente regla
- 4) Si todas las reglas que pueden satisfacer el objetivo actual se han probado y han fallado, entonces el objetivo en curso permanece indeterminado. Sacar éste de la pila y volver al paso 2). Si la pila de objetivos está vacía, el intérprete se detiene

IAIC – Curso 2010-11

Tema 3.5 - 27

Ejemplo de encadenamiento regresivo

❑ Traza de ejecución para derivar cereza como valor de fruta

- 1) Objetivos: (Fruta) Base de hechos inicial: vacía
- 2) Reglas que pueden satisfacer el objetivo: 1, 6 - 13
- 3) Examinamos premisas:
 - ❑ Regla 1: La 1ª premisa (Forma= alargada) no se encuentra en la memoria de trabajo. No hay reglas que deriven este valor → c)
 - ❑ ¿Cuál es el valor de Forma? redonda
 - Memoria de trabajo: ((Forma redonda)) 
 - La regla 1 falla
 - ❑ Regla 6: La 1ª premisa (claseFruta = planta) no se encuentra en la MT. Las reglas 2 y 3 pueden derivar el valor claseFruta → b)
 - Objetivos: (claseFruta, Fruta) 
- 2) Reglas que pueden satisfacer el objetivo: 2, 3
- 3) Examinamos premisas

IAIC – Curso 2010-11

Tema 3.5 - 28

Objetivo = Fruta

- ✗ R1: SI Forma = alargada Y Color = verde o amarillo ENTONCES Fruta = banana
- R2: SI Forma = redonda u ovalada Y Diámetro > 1.6 cm ENTONCES claseFruta = planta
- R3: SI Forma = redonda Y Diámetro < 1.6 cm ENTONCES claseFruta = árbol
- R4: SI numSemillas = 1 ENTONCES claseSemilla = hueso Objetivos: (Fruta)
- R5: SI numSemillas > 1 ENTONCES claseSemilla = múltiple Hechos: Forma = redonda
- R6: SI claseFruta = planta Y Color = verde ENTONCES Fruta = sandía
- R7: SI claseFruta = planta Y Color = amarillo ENTONCES Fruta = melón
- R8: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = albaricoque
- R9: SI claseFruta = árbol Y Color = naranja Y claseSemilla = múltiple ENTONCES Fruta = naranja
- R10: SI claseFruta = árbol Y Color = rojo Y claseSemilla = hueso ENTONCES Fruta = cereza
- R11: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = melocotón
- R12: SI claseFruta = árbol Y Color = rojo o amarillo o verde Y claseSemilla = múltiple ENTONCES Fruta = manzana
- R13: SI claseFruta = árbol Y Color = morado Y claseSemilla = hueso ENTONCES Fruta = ciruela

IAIC – Curso 2010-11

Tema 3.5 - 29

Objetivo = claseFruta

- ✗ R1: SI Forma = alargada Y Color = verde o amarillo ENTONCES Fruta = banana
- R2: SI Forma = redonda u ovalada Y Diámetro > 1.6 cm ENTONCES claseFruta = planta
- R3: SI Forma = redonda Y Diámetro < 1.6 cm ENTONCES claseFruta = árbol
- R4: SI numSemillas = 1 ENTONCES claseSemilla = hueso Objetivos: (claseFruta, Fruta)
- R5: SI numSemillas > 1 ENTONCES claseSemilla = múltiple Hechos: Forma = redonda
- R6: SI claseFruta = planta Y Color = verde ENTONCES Fruta = sandía
- R7: SI claseFruta = planta Y Color = amarillo ENTONCES Fruta = melón
- R8: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = albaricoque
- R9: SI claseFruta = árbol Y Color = naranja Y claseSemilla = múltiple ENTONCES Fruta = naranja
- R10: SI claseFruta = árbol Y Color = rojo Y claseSemilla = hueso ENTONCES Fruta = cereza
- R11: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = melocotón
- R12: SI claseFruta = árbol Y Color = rojo o amarillo o verde Y claseSemilla = múltiple ENTONCES Fruta = manzana
- R13: SI claseFruta = árbol Y Color = morado Y claseSemilla = hueso ENTONCES Fruta = ciruela

IAIC – Curso 2010-11

Tema 3.5 - 30

Ejemplo de encadenamiento regresivo

- ❑ Regla 2: La 1ª premisa (Forma = redonda u ovalada) es satisfecha puesto que el valor de Forma es redonda. Se continúa con la siguiente premisa (Diámetro > 1.6 cm). Puesto que no existe un valor de Diámetro ni se puede derivar de otras reglas → c)

❑ ¿Cuál es el valor de Diámetro? 0.4

Memoria de trabajo: ((Forma redonda) (Diámetro 0.4))

La regla 2 falla

- ❑ Regla 3: Las premisas se cumplen: (Forma = redonda), (Diámetro < 1.6 cm) → a) Se deriva (claseFruta = árbol)

Memoria de trabajo: ((Forma redonda) (Diámetro 0.4) (claseFruta árbol))

Objetivos: (Fruta)

3) Volvemos a la regla 6

- ❑ Regla 6: La 1ª premisa (claseFruta = planta) no se encuentra en la MT ni puede derivarse pues ya sabemos que no se cumple. Falla
- ❑ Regla 7: Falla por el mismo motivo

Objetivo = Fruta

- ✗ R1: SI Forma = alargada Y Color = verde o amarillo ENTONCES Fruta = banana
- ✗ R2: SI Forma = redonda u ovalada Y Diámetro > 1.6 cm ENTONCES claseFruta = planta
- R3: SI Forma = redonda Y Diámetro < 1.6 cm ENTONCES claseFruta = árbol
- R4: SI numSemillas = 1 ENTONCES claseSemilla = hueso
- R5: SI numSemillas > 1 ENTONCES claseSemilla = múltiple
- ✗ R6: SI claseFruta = planta Y Color = verde ENTONCES Fruta = sandía
- ✗ R7: SI claseFruta = planta Y Color = amarillo ENTONCES Fruta = melón
- R8: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = albaricoque
- R9: SI claseFruta = árbol Y Color = naranja Y claseSemilla = múltiple ENTONCES Fruta = naranja
- R10: SI claseFruta = árbol Y Color = rojo Y claseSemilla = hueso ENTONCES Fruta = cereza
- R11: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = melocotón
- R12: SI claseFruta = árbol Y Color = rojo o amarillo o verde Y claseSemilla = múltiple ENTONCES Fruta = manzana
- R13: SI claseFruta = árbol Y Color = morado Y claseSemilla = hueso ENTONCES Fruta = ciruela

Objetivos: (Fruta)
Hechos: Forma = redonda,
Diámetro = 0.4
claseFruta = árbol

Ejemplo de encadenamiento regresivo

- ❑ Regla 8: La 1ª premisa (claseFruta = árbol) es satisfecha. Se continúa con la siguiente premisa (Color = naranja). Puesto que no existe un valor de color ni se puede derivar de otras reglas → c)
 - ❑ ¿Cuál es el valor de Color? rojo
 - Memoria de trabajo:
 ((Forma redonda) (Diámetro 0.4) (claseFruta árbol) (Color rojo))
 - Falla la regla 8
 - ❑ Regla 9: Falla por el mismo motivo
 - ❑ Regla 10: Se cumplen las dos primeras premisas (claseFruta = árbol), (Color = rojo). Se continúa con la siguiente premisa (claseSemilla = hueso). Puesto que no existe un valor para claseSemilla pero se puede derivar de las reglas 4 y 5 → b)
 - Objetivos: (claseSemilla, Fruta)
- 2) Reglas que pueden satisfacer el objetivo: 4, 5
- 3) Examinamos premisas

Objetivo = claseSemilla

- ✗ R1: SI Forma = alargada Y Color = verde o amarillo ENTONCES Fruta = banana
 - ✗ R2: SI Forma = redonda u ovalada Y Diámetro > 1.6 cm ENTONCES claseFruta = planta
 - R3: SI Forma = redonda Y Diámetro < 1.6 cm ENTONCES claseFruta = árbol
 - R4: SI numSemillas = 1 ENTONCES claseSemilla = hueso
 - R5: SI numSemillas > 1 ENTONCES claseSemilla = múltiple
 - ✗ R6: SI claseFruta = planta Y Color = verde ENTONCES Fruta = sandía
 - ✗ R7: SI claseFruta = planta Y Color = amarillo ENTONCES Fruta = melón
 - ✗ R8: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = albaricoque
 - ✗ R9: SI claseFruta = árbol Y Color = naranja Y claseSemilla = múltiple ENTONCES Fruta = naranja
 - R10: SI claseFruta = árbol Y Color = rojo Y claseSemilla = hueso ENTONCES Fruta = cereza
 - R11: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = melocotón
 - R12: SI claseFruta = árbol Y Color = rojo o amarillo o verde Y claseSemilla = múltiple ENTONCES Fruta = manzana
 - R13: SI claseFruta = árbol Y Color = morado Y claseSemilla = hueso ENTONCES Fruta = ciruela
- Objetivos: (claseSemilla, Fruta)
 Hechos: Forma = redonda,
 Diámetro = 0.4
 claseFruta = árbol
 Color = rojo

Ejemplo de encadenamiento regresivo

- ❑ Regla 4: La premisa (numSemillas = 1) no se encuentra en la memoria de trabajo. No hay reglas que deriven este valor → c)

❑ ¿Cuál es el valor de NumSemilla? 1

Memoria de trabajo:

((Forma redonda) (Diámetro 0.4) (claseFruta árbol) (Color rojo) (numSemilla 1))

La premisa se cumple → a) Se deriva (claseSemilla = hueso)

Memoria de trabajo:

((Forma redonda) (Diámetro 0.4) (claseFruta árbol) (Color rojo) (numSemilla 1)
(claseSemilla = hueso))

Objetivos: (Fruta)



3) Volvemos a la regla 10

- ❑ Regla 10: La premisa se cumple → a) Se deriva (Fruta = cereza)

Memoria de trabajo:

((Forma redonda) (Diámetro 0.4) (claseFruta árbol) (Color rojo) (numSemilla 1)
(claseSemilla hueso) (Fruta cereza))

Objetivos: ()



4) El intérprete se detiene (pila de objetivos vacía)

Objetivo = Fruta

- ✗ R1: SI Forma = alargada Y Color = verde o amarillo **ENTONCES** Fruta = banana
- ✗ R2: SI Forma = redonda u ovalada Y Diámetro > 1.6 cm **ENTONCES** claseFruta = planta
- R3: SI Forma = redonda Y Diámetro < 1.6 cm **ENTONCES** claseFruta = árbol
- R4: SI numSemillas = 1 **ENTONCES** claseSemilla = hueso
- R5: SI numSemillas > 1 **ENTONCES** claseSemilla = múltiple
- ✗ R6: SI claseFruta = planta Y Color = verde **ENTONCES** Fruta = sandía
- ✗ R7: SI claseFruta = planta Y Color = amarillo **ENTONCES** Fruta = melón
- ✗ R8: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso **ENTONCES** Fruta = albaricoque
- ✗ R9: SI claseFruta = árbol Y Color = naranja Y claseSemilla = múltiple **ENTONCES** Fruta = naranja
- R10: SI claseFruta = árbol Y Color = rojo Y claseSemilla = hueso **ENTONCES** Fruta = cereza
- R11: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso **ENTONCES** Fruta = melocotón
- R12: SI claseFruta = árbol Y Color = rojo o amarillo o verde Y claseSemilla = múltiple **ENTONCES** Fruta = manzana
- R13: SI claseFruta = árbol Y Color = morado Y claseSemilla = hueso **ENTONCES** Fruta = ciruela

Objetivos: (Fruta)

Hechos:

numSemilla = 1
claseSemilla = hueso



No quedan objetivos: fin

- ✗ R1: SI Forma = alargada Y Color = verde o amarillo ENTONCES Fruta = banana
 ✗ R2: SI Forma = redonda u ovalada Y Diámetro > 1.6 cm ENTONCES claseFruta = planta
 R3: SI Forma = redonda Y Diámetro < 1.6 cm ENTONCES claseFruta = árbol
 R4: SI numSemillas = 1 ENTONCES claseSemilla = hueso
 R5: SI numSemillas > 1 ENTONCES claseSemilla = múltiple
 ✗ R6: SI claseFruta = planta Y Color = verde ENTONCES Fruta = sandía
 ✗ R7: SI claseFruta = planta Y Color = amarillo ENTONCES Fruta = melón
 ✗ R8: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = albaricoque
 ✗ R9: SI claseFruta = árbol Y Color = naranja Y claseSemilla = múltiple ENTONCES Fruta = naranja
 R10: SI claseFruta = árbol Y Color = rojo Y claseSemilla = hueso ENTONCES Fruta = cereza
 R11: SI claseFruta = árbol Y Color = naranja Y claseSemilla = hueso ENTONCES Fruta = melocotón
 R12: SI claseFruta = árbol Y Color = rojo o amarillo o verde Y claseSemilla = múltiple ENTONCES Fruta = manzana
 R13: SI claseFruta = árbol Y Color = morado Y claseSemilla = hueso ENTONCES Fruta = ciruela
- Objetivos: ()
 Hechos:
 claseFruta = árbol
 Color = rojo
 claseSemilla = hueso
 Fruta = cereza

IAIC – Curso 2010-11

Tema 3.5 - 37

Implementación: fase de reconocimiento

- ☐ Encaje (*matching*) por ajuste o encaje de patrones
 - ☐ Qué reglas son aplicables en un cierto estado
- ☐ La situación es distinta dependiendo de si el enfoque es con encadenamiento hacia delante o hacia atrás
 - ☐ Encadenamiento hacia atrás → parte derecha de las reglas
 - ☐ Encadenamiento hacia adelante → parte izquierda de las reglas
- ☐ Existencia de variables → encaje con ligadura de variables
- ☐ Los mecanismos de encadenamiento utilizan diversas estrategias para agilizar el reconocimiento de reglas
 - ☐ Se pueden indexar las reglas para mejorar la eficiencia
 - ☐ Con encadenamiento hacia delante se indexan por el antecedente
 - ☐ Con encadenamiento hacia atrás se indexan por el consecuente
 - ☐ Tablas hash
 - ☐ Algoritmo de encaje más eficiente que el encaje uno a uno : RETE

IAIC – Curso 2010-11

Tema 3.5 - 38

Implementación: fase de reconocimiento

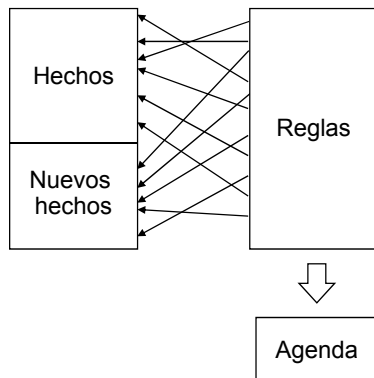
- ❑ El encaje uno a uno es muy poco eficiente:
 - ❑ Si en cada ciclo del proceso de inferencia llevado a cabo en el encadenamiento hacia delante hubiera que comprobar todas las reglas aplicables en función de la memoria de trabajo, el sistema sería muy poco eficiente
 - ❑ Teniendo en cuenta que en cada ciclo los cambios producidos en la base de hechos son los únicos que influyen en la modificación del conjunto conflicto, se toman dichas actualizaciones como base del cálculo de las reglas que pasarán a formar parte de este conjunto (*redundancia temporal*)
 - ❑ De esta forma, los nuevos hechos que son introducidos en la memoria de trabajo son los que determinan las nuevas reglas aplicables
 - ❑ Idea del algoritmo RETE: en lugar de buscar qué reglas satisfacen los hechos existentes en cada momento, son los nuevos hechos generados en cada ciclo los que buscan o determinan qué nuevas reglas se seleccionan

IAIC – Curso 2010-11

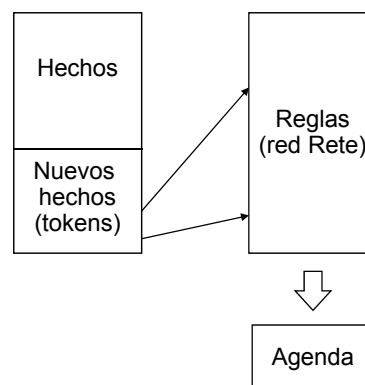
Tema 3.5 - 39

Implementación: fase de reconocimiento

- ❑ Método tradicional
 - ❑ Cada regla verifica si se cumplen sus condiciones a partir de la base de hechos total
- ❑ Algoritmo RETE
 - ❑ Los nuevos hechos buscan reglas



IAIC – Curso 2010-11

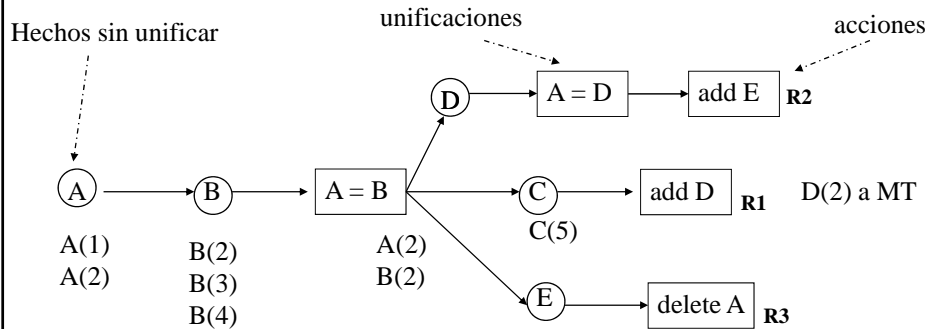


Tema 3.5 - 40

Ejemplo de RETE

- ❑ R1 - $A(x), B(x), C(y) \Rightarrow \text{add } D(x)$
- ❑ R2 - $A(x), B(x), D(x) \Rightarrow \text{add } E(x)$
- ❑ R3 - $A(x), B(x), E(y) \Rightarrow \text{delete } A(x)$
- ❑ MT - $\{A(1), A(2), B(2), B(3), B(4), C(5)\}$

--- Añadir regla a Red --
 Si existe nodo de premisa:
 conectar
 Sino: crearlo y conectar



Qué pasa si añado R4 - $A(x), D(x) \Rightarrow \text{add } M(x)$

IAIC - Curso 2010-11

Tema 3.5 - 41

Algoritmo RETE (Forgy, 1982)

- ❑ Realiza un encaje de patrones eficiente (muchos a muchos)
 - ❑ entre la base de reglas y la base de hechos
- ❑ Evita reevaluar condiciones ya evaluadas
- ❑ Aprovecha la **redundancia temporal**:
 - ❑ Los posibles cambios en la agenda entre ciclo son pocos
- ❑ Aprovecha la **similitud estructural**:
 - ❑ Muchas reglas comparten condiciones
- ❑ Construye un grafo de dependencias estableciendo las reglas que comparten condiciones
 - ❑ En un ciclo de operación, las nuevas reglas que se añaden a la agenda son aquéllas que dependían de condiciones que acaban de hacerse ciertas al aplicar la regla anterior

IAIC - Curso 2010-11

Tema 3.5 - 42

Estrategias de control o resolución de conflictos - I

- ❑ Estrategias de resolución del conjunto conflicto (agenda)
 - ❑ Qué regla del conjunto conflicto que será disparada en cada ciclo
- ❑ Dos enfoques de estrategias
 - ❑ **Control global:** control independiente del dominio de aplicación
 - ❑ Estrategias implementadas en el intérprete (optimización)
 - ❑ No son modificables por el programador
 - ❑ **Control local:** control dependiente del dominio de aplicación
 - ❑ El programador da instrucciones explícitas: establecer prioridades
- ❑ Objetivos, que el sistema se comporte de forma
 - ❑ Sensible: responde a cambios reflejados en la memoria de trabajo
 - ❑ Estable: haya continuidad en la línea de razonamiento
- ❑ **Refracción:** una regla sólo se dispara una vez con los mismos hechos
- ❑ La estrategia ha de tener un equilibrio entre sistemática y eficiencia

Estrategias de control o resolución de conflictos – I I

1. **Orden** en la base de reglas
 - ❑ Se selecciona la primera regla aplicable,
 - ❑ Asume orden lineal explícito en la BR (lo cual puede ser peligroso)
2. **Prioridades** asociadas a las reglas (*conocimiento heurístico*)
 - ❑ Se selecciona la regla de mayor prioridad: (`declare (salience 50)`)
 - ❑ La prioridad la establece el experto del dominio
3. **Especificidad:** más prioridad reglas con más condiciones
4. **Novedad:** prioridad reglas según
 - a) Antigüedad (en MT) de sus hechos
 - b) Momento de activación de la regla
 - (set-strategy depth) : activa ahora la más reciente activada
 - (set-strategy breadth) : activa ahora la más antigua activada
5. **Arbitrariedad**
 - ❑ Cuando no existen criterios adicionales disponibles (*no sistemática*)

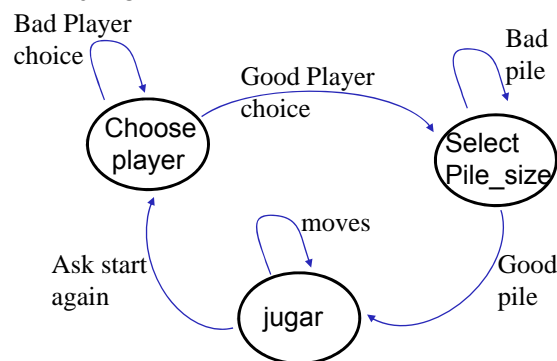
Diseñar las Bases de Reglas: usando Autómatas

1.- Un problema se divide en estados

- ❑ Un Módulo representa un estado del problema
- ❑ Qué posibles estados puede tener un problema?

2.- Definir las relaciones entre estados

- ❑ Qué provoca la transición de un estado a otro?
- ❑ Tres **fases** del juego del sticks.clp



IAIC – Curso 2010-11

Tema 3.5 - 45

Diseñar las Bases de Reglas: Módulos

- ❑ Cuando un problema es grande conviene
 - ❑ Dividirlo en subproblemas: cada uno es un módulo
- ❑ Un Módulo es un espacio de nombres
 - ❑ Base de reglas dividida en bloques o módulos: Mejor diseño
 - ❑ Ejemplo: módulo de averías mecánicas y módulo de averías eléctricas
 - ❑ Más eficiente la selección de reglas y el razonamiento
 - ❑ Ejemplo
 - ❑ `(defmodule etapaCinco)`
 - ❑ `(defrule etapaCinco::PasoUno ...)`
 - ❑ `(focus etapaCinco) ... (run)`
- ❑ En ejecución de Jess lo hace por módulos
 - ❑ En `runJessEnShellUnix.txt` se usa por defecto "module MAIN"
- ❑ Mas detalles en `materialesIntroduccionJess`

IAIC – Curso 2010-11

Tema 3.5 - 46

Ventajas de los Sistemas de Reglas - I

- ❑ Separación de conocimiento y control
 - ❑ Simplifica el desarrollo de sistemas expertos
- ❑ Carácter totalmente modular de la base de conocimiento
- ❑ Uniformidad
 - ❑ Conocimiento representado con una sintaxis simple y homogénea
- ❑ Naturalidad para representar cierto tipo de conocimiento
- ❑ Control dirigido por patrones
 - ❑ Los programas en IA necesitan flexibilidad
 - ❑ Las reglas pueden “dispararse” en cualquier secuencia
 - ❑ Estado → reglas aplicables → camino de la solución

Ventajas de los Sistemas de Reglas - II

- ❑ Independencia del lenguaje
 - ❑ El modelo es independiente de la representación elegida para reglas y hechos, siempre que el lenguaje soporte encaje de patrones
- ❑ Correspondencia natural con búsqueda en espacios de estados
 - ❑ Nodos: estados sucesivos de la MT, operadores: reglas
- ❑ Modelo plausible del mecanismo humano de resolución de problemas
 - ❑ Razonamiento basado en reglas: decidir qué hacer o qué se puede deducir cuando se dan una serie de condiciones
 - ❑ Uno de los primeros usos de los sistemas de producción (*Newell & Simon*). Siguen usándose para estudiar el comportamiento humano
- ❑ Trazabilidad del proceso de razonamiento (ver runJessEnShellUnix.txt)
 - ❑ Se pueden comunicar qué condiciones se conocían en un momento determinado, por qué se ha escogido una regla y qué se ha deducido tras aplicarla
 - ❑ Ayuda a la depuración

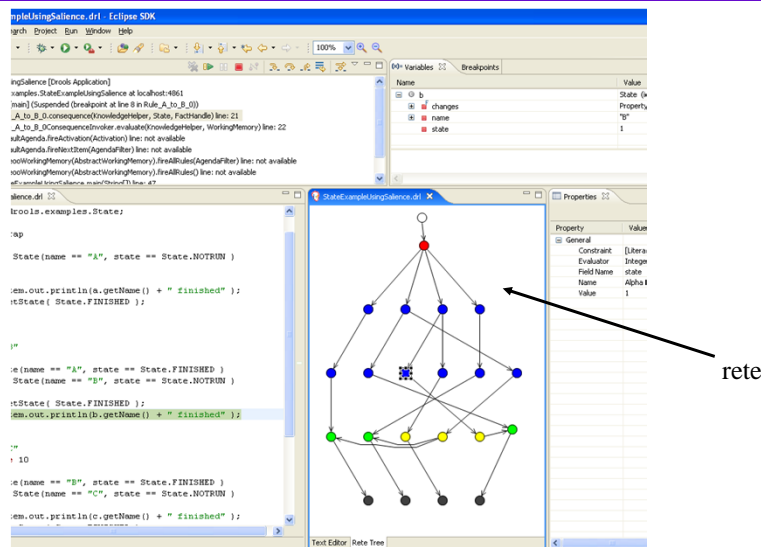
Inconvenientes de los Sistemas de Reglas

- ❑ Ineficiencia (se mejora con algoritmos como RETE)
 - ❑ El proceso de reconocimiento de patrones puede ser ineficiente
 - ❑ El precio a pagar por la modularidad, flexibilidad y uniformidad
- ❑ Dificultad de verificación de la consistencia de la BC
 - ❑ ¿Conocimiento contradictorio al modificar la BC?
- ❑ Opacidad y dificultad de depuración
 - ❑ Es difícil examinar una BC y determinar qué acciones van a ocurrir (depende del motor de inferencia)
 - ❑ No hay un flujo de control claro
 - ❑ La división del conocimiento en reglas hace que cada regla individual sea fácilmente tratable, pero se pierde la visión global
 - ❑ Las reglas representan pasos muy pequeños en la resolución del problema sin que haya una jerarquía de reglas
- ❑ Dificultad en cubrir todo el conocimiento
 - ❑ Cuello de botella de la adquisición de conocimiento
- ❑ Herramientas de desarrollo intentan suplir con utilidades

IAIC – Curso 2010-11

Tema 3.5 - 49

Sistemas de Reglas: Drools



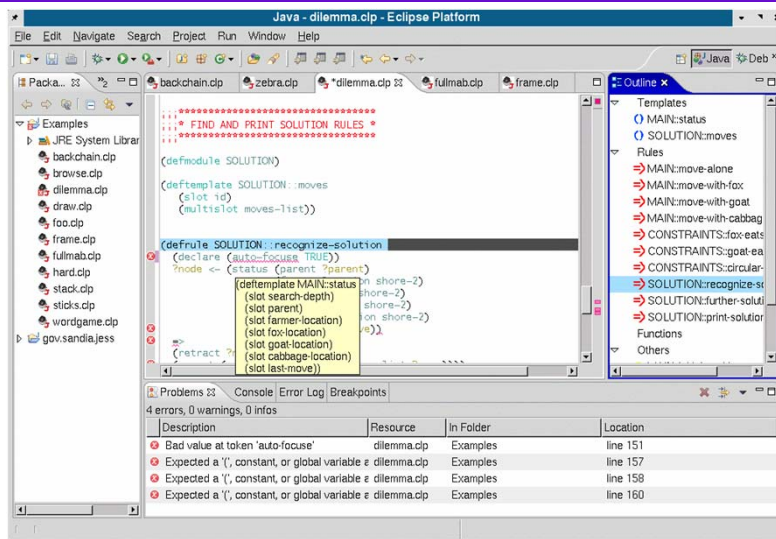
<http://www.jboss.org/drools/>

Características: <http://www.jboss.org/drools/featuresandscreenshots.html>

IAIC – Curso 2010-11

Tema 3.5 - 50

Sistemas de Reglas: JessDE, plugin para Eclipse

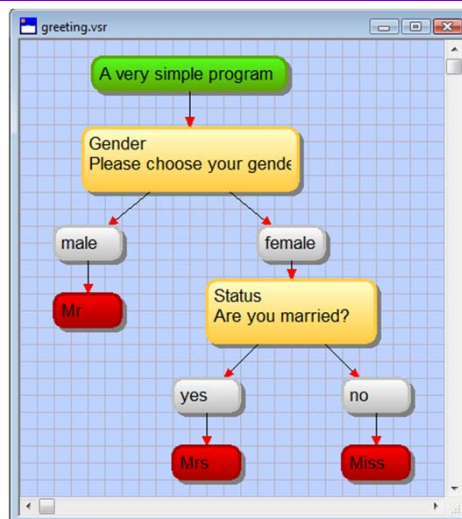


<http://www.jessrules.com/>
Demo: <http://www.jessrules.com/jessdemo/>

IAIC – Curso 2010-11

Tema 3.5 - 51

Sistemas de Reglas: VisiRules (LPA)



<http://www.lpa.co.uk/vsr.htm>
Demos online : http://www.lpa.co.uk/vrs_dem.htm

IAIC – Curso 2010-11

Tema 3.5 - 52

Materiales en el Campus Virtual: materialesIntroduccionJess.rar

- ☐ cadenas.CLP
- ☐ ejemploletrasversion1-H5Ej3.clp
- ☐ ejemploTrozoJuegoJess.txt
- ☐ empezarConJessAqui.docx
- ☐ IdeasConstruirBReglas.doc
- ☐ Jess71p2.zip
- ☐ runCadenas.txt
- ☐ runJessEnShellUnix.txt
- ☐ sticks.clp
- ☐ tutorialBreveJessConJava.docx

Bibliografía

- ☐ **Fernández, S., González, J., Mira, J.**
Problemas resueltos de Inteligencia artificial aplicada
UNED. Pearson, 1998 (2005).
- ☐ **Rich, E. y Knight, K.**
Artificial Intelligence.
McGraw-Hill, 1991, 2ª edición.
- ☐ **Russell, S. y Norvig, P.**
Inteligencia Artificial: Un Enfoque Moderno.
Prentice Hall, 2004, 2ª edición.
- ☐ **Luger, G.F.**
Artificial Intelligence.
Addison-Wesley, 2005, 5ª edición.
- ☐ **Nilsson, J.**
Artificial Intelligence: A New Synthesis.
Prentice Hall, 2004, 2ª edición.

Bibliografía

- ❑ **Jackson, P.**
Introduction to Expert Systems.
Addison-Wesley, 1999.
- ❑ **Gonzalez, A. J. y Dankel, D. D.**
The Engineering of Knowledge Based Systems:
Theory and Practice.
Prentice Hall, 1993.
- ❑ **Giarratano, J. y Riley, G.**
Sistemas Expertos. Principios y Programación
Thomson, 2001.