

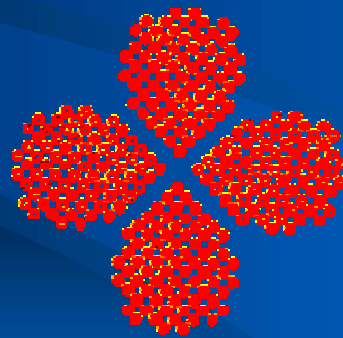
Sistemas Expertos

Introducción

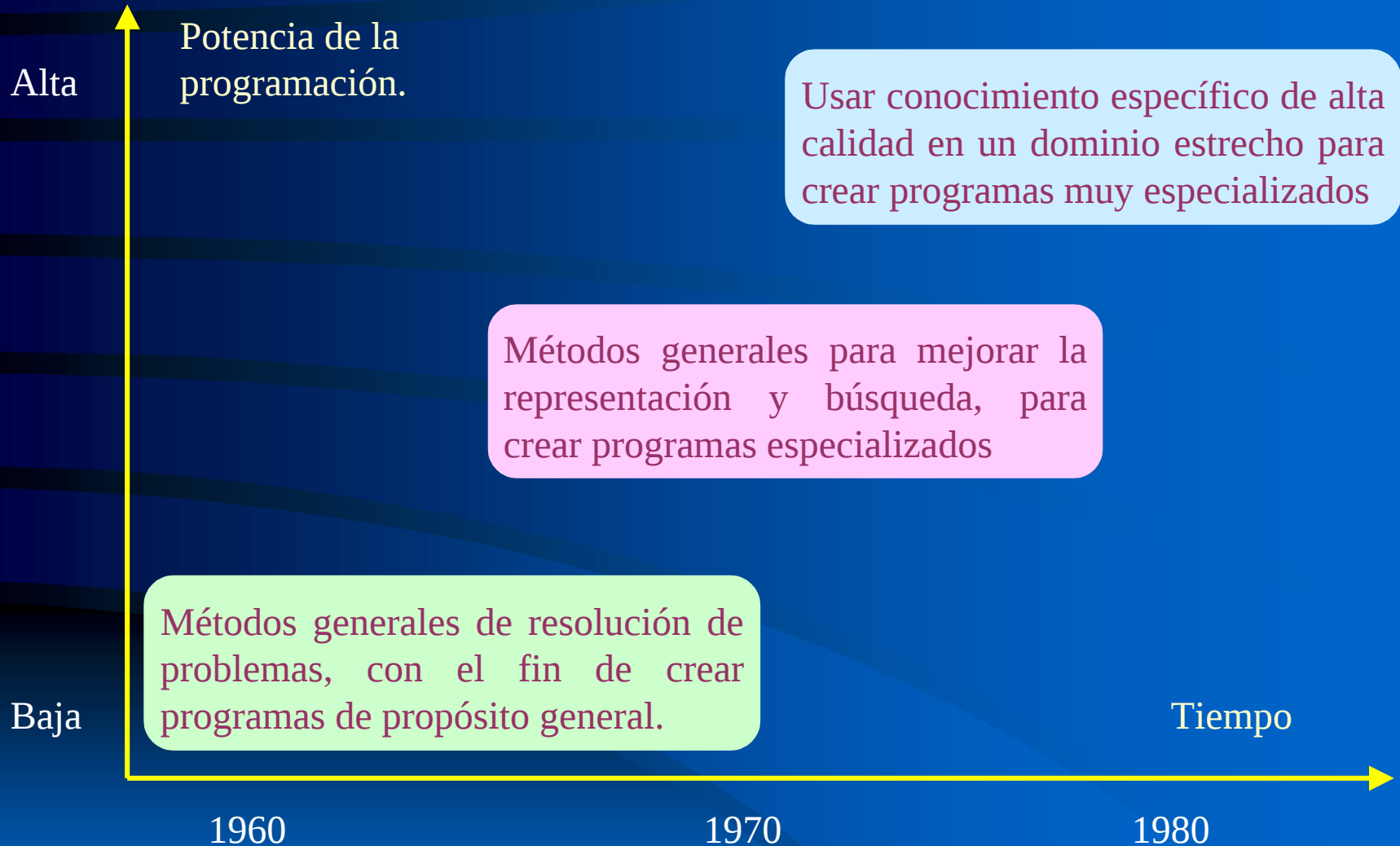
Arquitectura - Motor de Inferencias

Licenciatura en Ciencias de la Computación

Introducción a la Inteligencia Artificial



Evolución temporal de la investigación.



Sistemas basados en conocimiento

Definiciones.

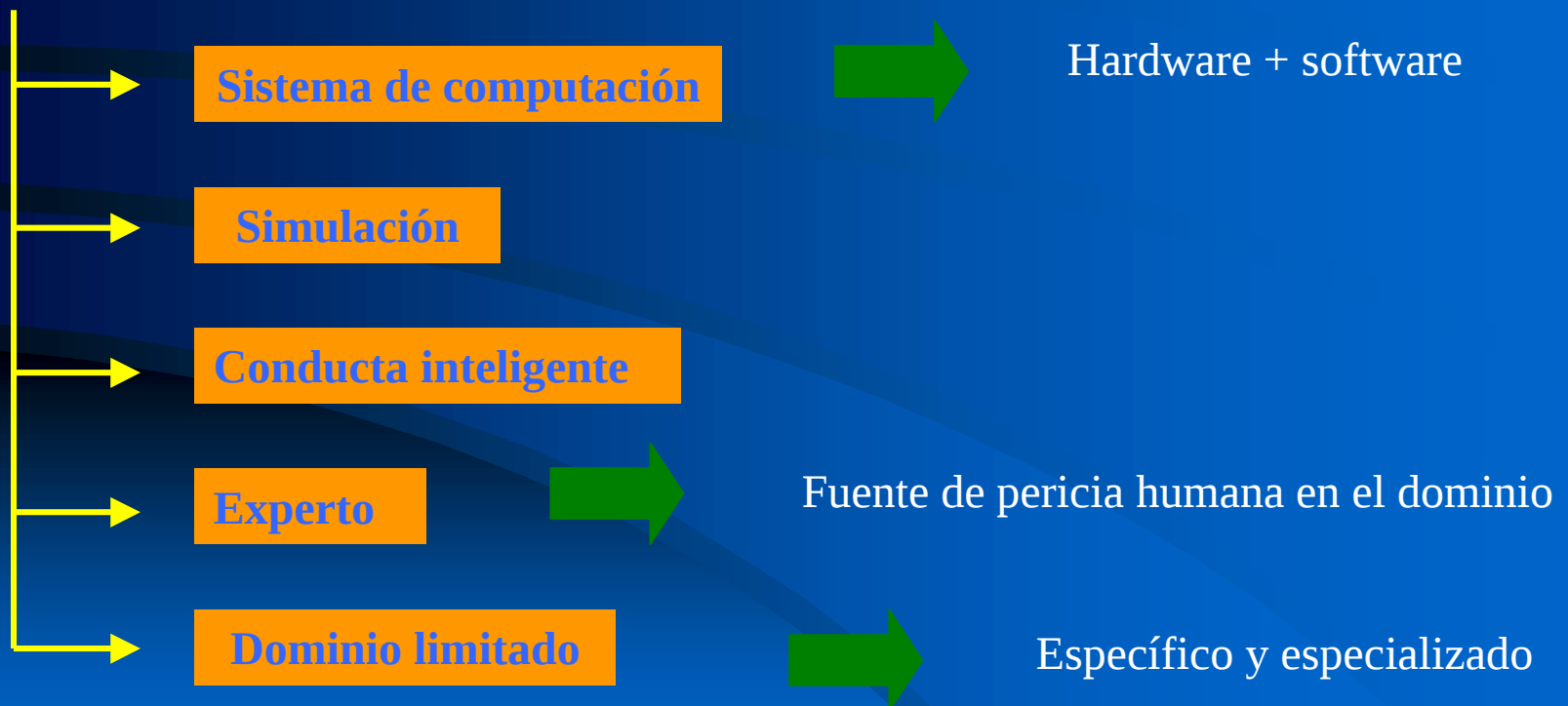
Sistemas que resuelven problemas aplicando una representación simbólica de la experiencia humana.
Jackson.

Sistemas que representan el conocimiento sobre el dominio en forma explícita y separada del resto del sistema.

Waterman.

Sistemas Expertos: Definición.

Es un sistema de cómputo capaz de simular la conducta inteligente de un experto humano en un dominio específico y especializado, con el objeto de resolver problemas.



Primeros Sistemas Expertos

MYCIN

Stanford (Buchanan - Shortliffe 1976): Diagnóstico y tratamiento de enfermedades infecciosas (aprox. 500 reglas). Introduce factores de certeza.

PROSPECTOR

SRI - US Geological Survey (Duda-Gashning-Hart, 1979): Evaluación de yacimientos minerales y petrolíferos. (aprox 1600 reglas). Probabilidades condicionales y Teorema de Bayes.

INTERNIST

Pittsburg (Pople-Myers, 1977): Diagnóstico en Medicina interna. CADUCEUS: Analiza relaciones causales.

CASNET

Rutgens (Kulikowsky-Weiss-Safir, 1977): Oftalmología. Aplica un modelo causal para el diagnóstico de glaucoma.

R1

Carnegie Mellon (Mc Dermott): Configuración de ordenadores VAX (aprox 500 reglas). Descompone el diseño en distintos niveles de abstracción.

MOLGEN

Stanford (Stefik): Diseño de experimentos genéticos. Utiliza abstracción.

Conocimiento
inexacto

Razonamiento experto
Modelos causales

Reducción espacio
de búsqueda

Características de los Sistemas Expertos.

Poderoso cuerpo de conocimiento especializado



Explícito y organizado para favorecer la toma de decisiones

Pericia de alto nivel



Gran habilidad en la resolución de problemas complejos



Sistema costo-efectivo de aceptación en el mercado.



Memoria institucional

Registro permanente de las mejores estrategias y métodos utilizados. Fundamental si hay relevos frecuentes.

Modelaje predictivo

Se pueden evaluar los resultados de nuevas estrategias agregando o quitando reglas.

Facilidad de entrenamiento

El SE puede funcionar como tutor si tiene la interfaz adecuada.

Habilidades que se esperan de un SE

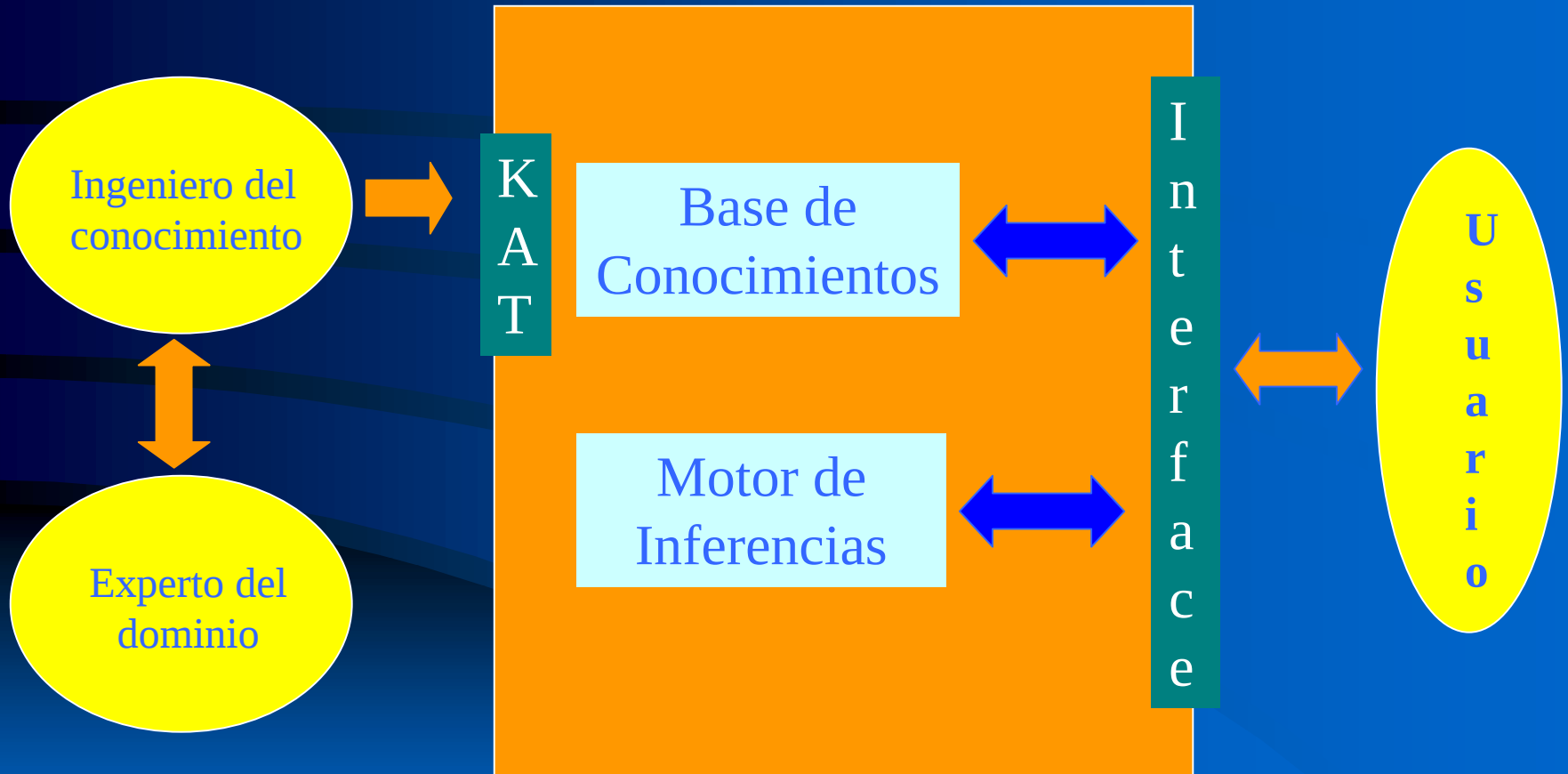
- ✓ Manipular con fluidez descripciones simbólicas.
- ✓ Buena performance, comparable por lo menos a la de expertos humanos.
- ✓ Interfaz amigable y adecuada a las necesidades de la aplicación.
- ✓ Habilidad para manipular conocimiento incompleto e impreciso.
- ✓ Buena capacidad de explicación para sus requerimientos a los usuarios.
- ✓ Posibilidad de justificar sus conclusiones.

Experiencia humana vs. SE.

PERICIA ARTIFICIAL	PERICIA HUMANA
Permanente	Perecedera
Siempre accesible	No siempre accesible
Fácil de documentar	No documentable
Estable, confiable	Variable, flexible ✓
Costo manejable	Costosa
Fragilidad de las máquinas	Fragilidad humana
Sin inspiración	Creativa ✓
Conocimiento técnico	Sentido común ✓
Estática	Dinámica (aprendizaje) ✓
Dominios estrechos	Dominios más extensos

✓ Los humanos son aún imprescindibles.

Estructura básica de un SE.



Estructura básica de un Sistema Experto

Base de Conocimientos



Su estructura de datos queda definida en términos del esquema de representación elegido para incorporar el conocimiento del dominio de trabajo.



Separados entre sí

Motor de Inferencias



Es la estructura de control de un SE, contiene el programa que gestiona la BC y otros mecanismos necesarios para administrar un sistema de naturaleza interactiva.

SE basados en reglas de producción

Ventajas:

- ↳ **Naturaleza modular:** El conocimiento es fácil de encapsular y expandir.
- ↳ **Explicaciones sencillas:** El registro de reglas disparadas permite presentar la cadena de razonamiento asociada.
- ↳ **Semejanza con el proceso cognitivo humano:** modelo natural del razonamiento humano.

Dificultades:

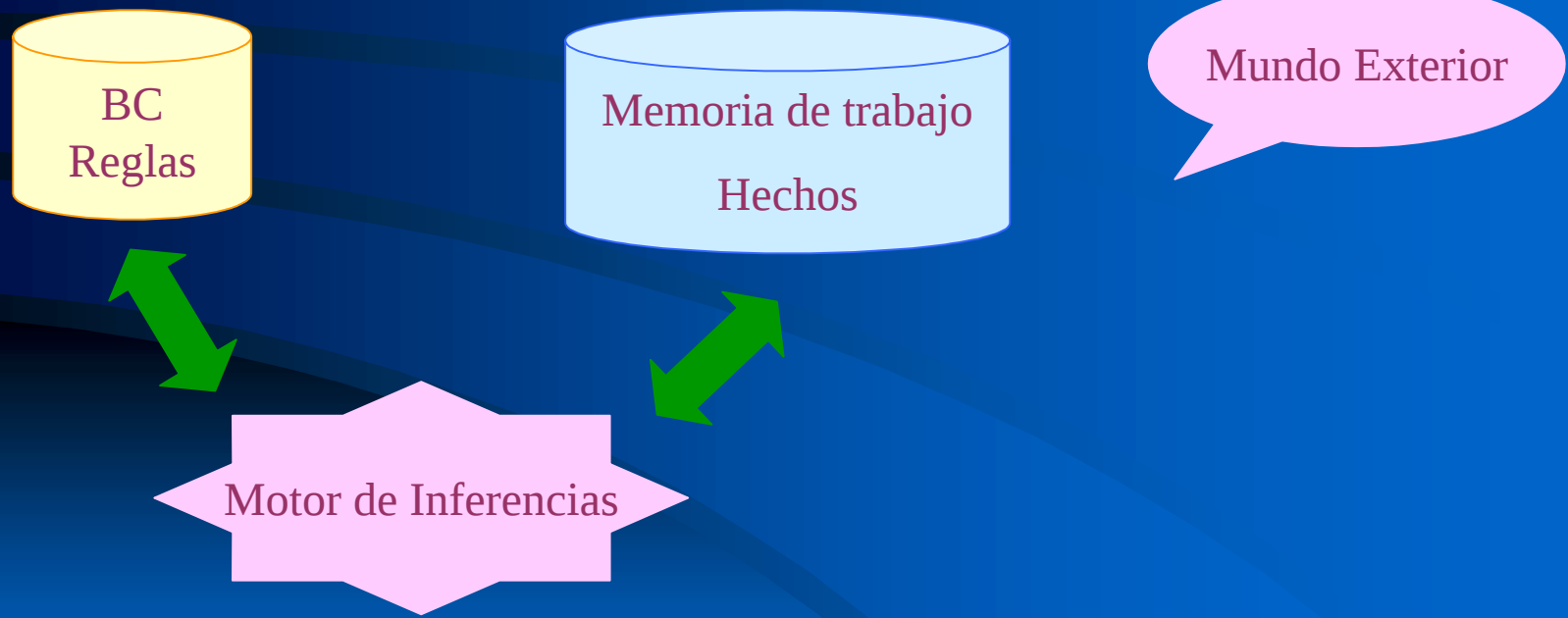
- ↳ Completitud y consistencia como responsabilidad del desarrollador.
- ↳ Se necesitan otras estructuras para lograr una visión global del conocimiento del dominio.

Se las combina con otros formalismos de representación.

SE basados en reglas de producción

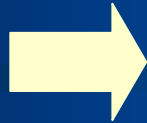
Sistemas de producción

Newell y Simon (1972): Al resolver problemas, las personas utilizan su **memoria a largo plazo** (permanente) que aplican a situaciones actuales contenidas en su **memoria a corto plazo** (transitoria). Esto puede generar modificaciones en la última.



SE basados en reglas de producción

Reglas de
producción



< CONDICION > < ACCION >



⇒ Son “gránulos” de conocimiento.

⇒ Reúnen información relativa a las condiciones de disparo y a los efectos resultantes del disparo.

⇒ Son estructuras bidireccionales.

SI ?X ES MAMIFERO Y ?X COME CARNE
ENTONCES ?X ES CARNIVORO.

SE basados en reglas de producción

Encadenamiento hacia atrás - Backward Chaining.

BH := CONOCIMIENTO INICIAL (HECHOS).

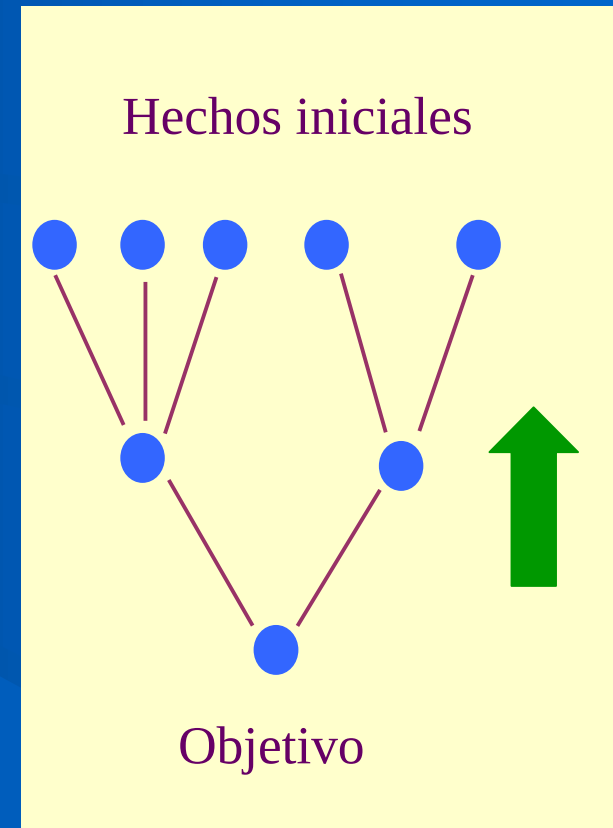
HASTA OBJETIVO O SIN REGLAS PARA DISPARAR.

REPITA

(1) ENCONTRAR K CONJUNTO DE REGLAS, CUYAS CONCLUSIONES PUEDEN UNIFICARSE CON LA HIPÓTESIS (CONJUNTO DE CONFLICTO).

(2) ELEGIR R DE K SEGÚN ESTRATEGIA DE SOLUCIÓN DE CONFLICTOS (POSIBLE BACKTRACKING).

(3) SI LA PREMISA DE R NO ESTÁ EN BH, TOMARLA COMO SUBOBJETIVO.



SE basados en reglas de producción

Backward Chaining: Ciclo base de un motor inductivo.

DETECCIÓN:

- ✓ SI EL OBJETIVO ES CONOCIDO ÉXITO.
- ✓ SINO, TOMAR LAS REGLAS QUE LO CONCLUYEN (CC).

ELECCIÓN:

- ✓ DECIDIR QUE REGLA APLICAR (RC)

APLICACIÓN:

- ✓ REEMPLAZAR EL OBJETIVO POR LA CONJUNCIÓN DE LAS CONDICIONES DE LA PREMISA ELEGIDA.

SE basados en reglas de producción

Ejemplo Backward Chaining: Primer paso

BASE DE REGLAS

R1: $p \wedge q \rightarrow s$

R2: $r \rightarrow t$

R3: $s \wedge t \rightarrow u$

R4: $s \wedge r \rightarrow v$



* OBJETIVO: v

* $v \in BH$?

* $v \notin BH$

* SIGUE



BH: $p \ q \ r$

SE basados en reglas de producción

Ejemplo Backward Chaining: Segundo paso

BASE DE REGLAS

R1: $p \wedge q \rightarrow s$

R2: $r \rightarrow t$

R3: $s \wedge t \rightarrow u$

R4: $s \wedge r \rightarrow v$



* MATCHING CON v

* $CC = \{ R4 \}$

* $s \in BH?$

* $s \notin BH$

* s SUBOBJETIVO

* SIGUE



BH: $p \ q \ r$

SE basados en reglas de producción

Ejemplo Backward Chaining: Tercer paso

BASE DE REGLAS

R1: $p \wedge q \rightarrow s$

R2: $r \rightarrow t$

R3: $s \wedge t \rightarrow u$

R4: $s \wedge r \rightarrow v$



BH: $p \ q \ r$



* MATCHING CON s

* $CC = \{ R1 \}$

* $p \in BH?$ SI.

* $q \in BH?$ SI.

* DISPARA R1.

* $BH \leftarrow s$

SE basados en reglas de producción

Ejemplo Backward Chaining: Cuarto paso

BASE DE REGLAS

R1: $p \wedge q \rightarrow s$

R2: $r \rightarrow t$

R3: $s \wedge t \rightarrow u$

R4: $s \wedge r \rightarrow v$



*CC = { R4 }

* $r \in \text{BH?}$ SI.

*DISPARA R4.

* $\text{BH} \leftarrow v$



BH: p q r s

SE basados en reglas de producción

Ejemplo Backward Chaining: Quinto paso

BASE DE REGLAS

R1: $p \wedge q \rightarrow s$

R2: $r \rightarrow t$

R3: $s \wedge t \rightarrow u$

R4: $s \wedge r \rightarrow v$



*OBJETIVO OK.

* FIN



BH: p q r s v

SE basados en reglas de producción

Ejemplo “Animales Carnívoros” para volcar en Kappa-PC.

* RCARNÍVORO1: SI EL ANIMAL MAMÍFERO COME CARNE
ENTONCES EL ANIMAL ES CARNÍVORO.

* RCARNÍVORO2: SI EL ANIMAL MAMÍFERO TIENE GARRAS Y DIENTES
AGUDOS
ENTONCES EL ANIMAL ES CARNÍVORO.

* RTIGRE: SI EL ANIMAL ES CARNÍVORO Y SU COLOR ES LEONADO Y
SU PELAJE TIENE FRANJAS NEGRAS
ENTONCES EL ANIMAL ES UN TIGRE.

* RLEOPARDO: SI EL ANIMAL ES CARNÍVORO Y SU COLOR ES LEONADO
Y SU PELAJE TIENE MANCHAS OSCURAS
ENTONCES EL ANIMAL ES UN LEOPARDO

SE basados en reglas de producción

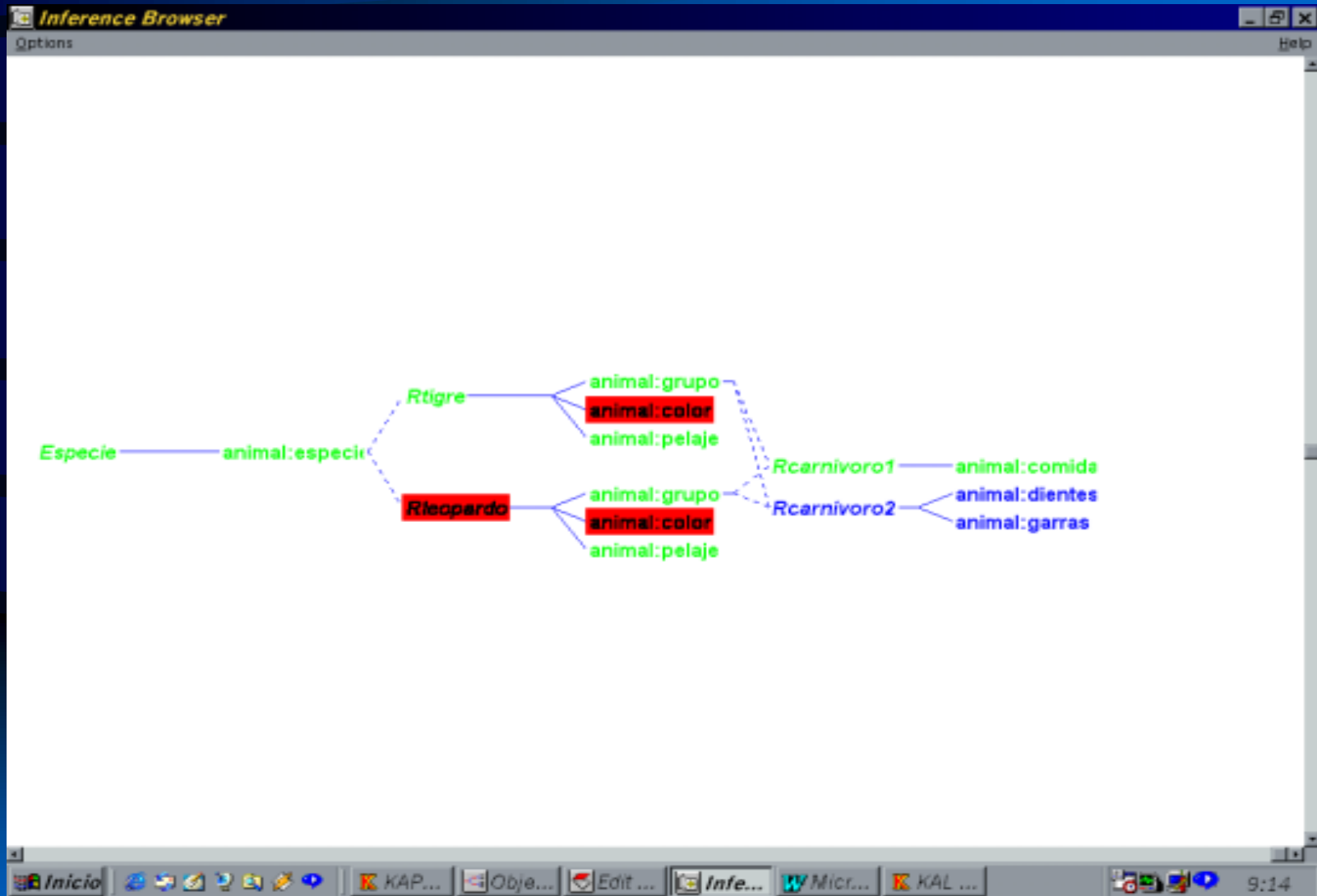
Backward Chaininig en Kappa-PC.



FASES

- ⇒ **EXPANDING**: EVALÚA LOS IF Y ABRE EL ÁRBOL CONSIDERANDO NUEVOS HECHOS.
- ⇒ **COLLAPSING**: TESTEA SI VERIFICA EL GOAL.
- ⇒ **ASKING**: PIDE INFORMACIÓN AL USUARIO.

Ejemplo “Animales Carnívoros” en Backward Chaining.



SE basados en reglas de producción

Encadenamiento hacia adelante - Forward Chaining.

BH := CONOCIMIENTO INICIAL (HECHOS).

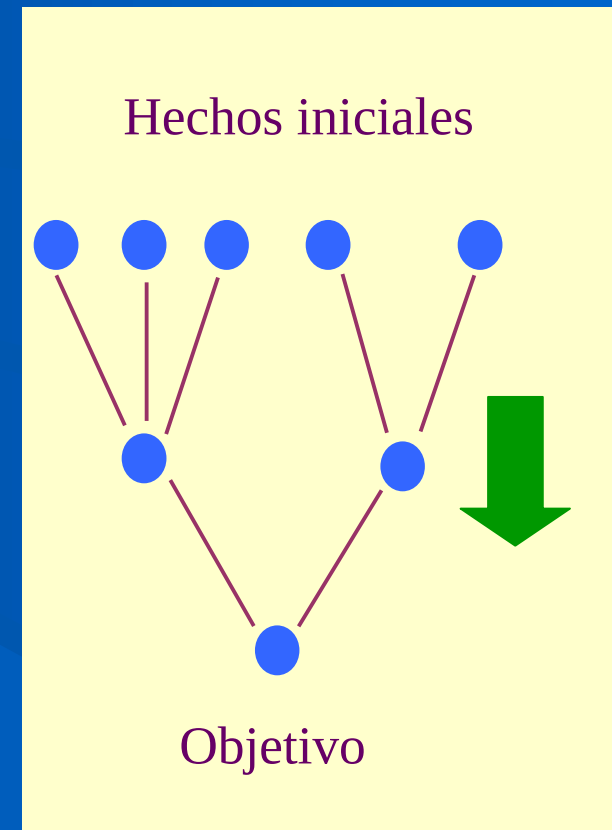
HASTA OBJETIVO O SIN REGLAS PARA DISPARAR.

REPITA:

(1) ENCONTRAR K CONJUNTO DE REGLAS CUYAS PREMISAS CUMPLEN CON BH (CONJUNTO DE CONFLICTO-CC).

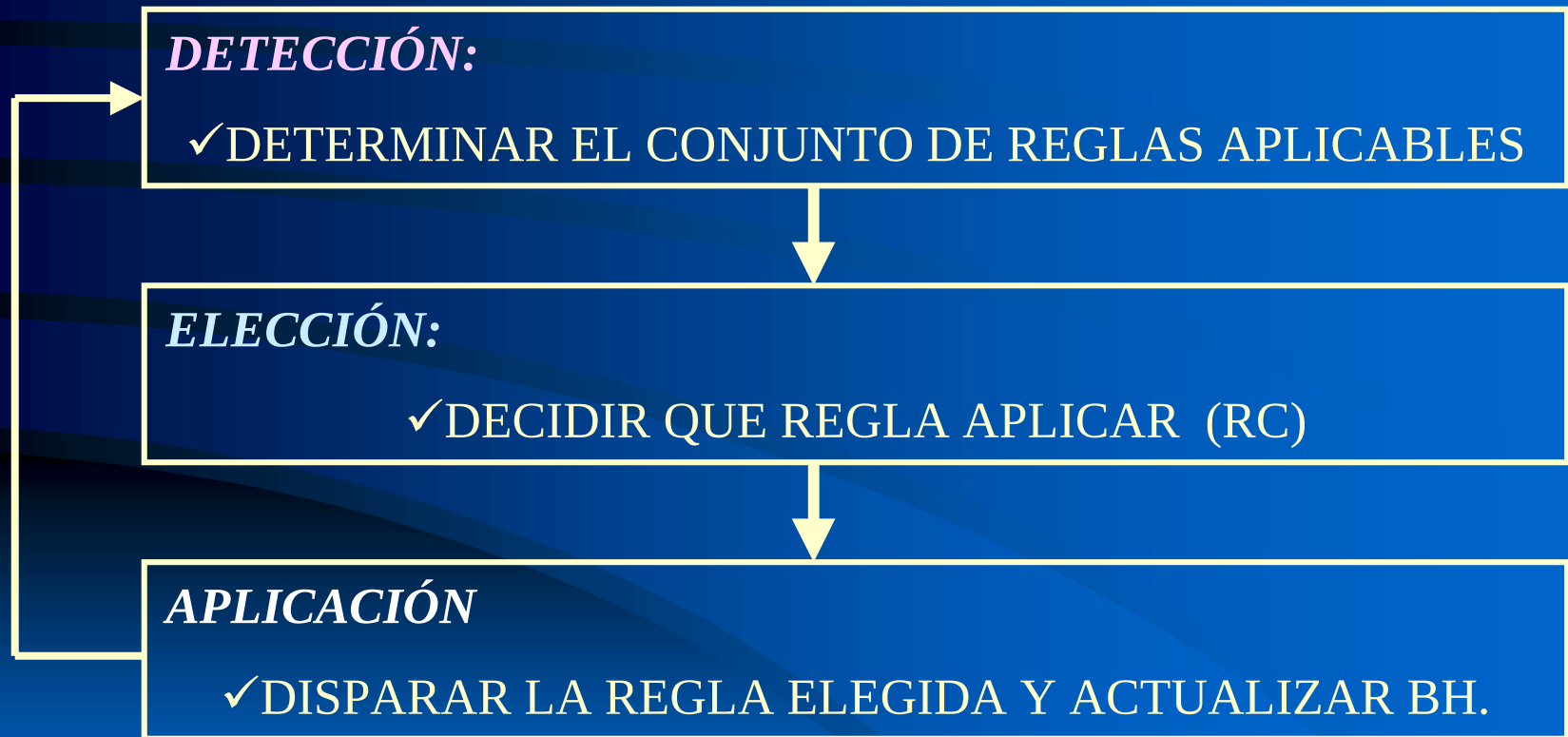
(2) ELEGIR R DE K SEGÚN ESTRATEGIA DE SOLUCIÓN DE CONFLICTOS (RC).

(3) DISPARAR R Y ACTUALIZAR BH. (RECORDAR R).



SE basados en reglas de producción

Forward Chaining: Ciclo base de un motor deductivo.



SE basados en reglas de producción

Ejemplo Forward Chaining: Primer paso

BASE DE REGLAS

R1: $p \wedge q \rightarrow s$

R2: $r \rightarrow t$

R3: $s \wedge t \rightarrow u$

R4: $s \wedge r \rightarrow v$



BH: $p \ q \ r$



*CC = { R1, R2 }

* $R1 \leftarrow RC$

*DISPARA R1

* $BH \leftarrow s$

* R1 APLICADA

SE basados en reglas de producción

Ejemplo Forward Chaining: Segundo paso

BASE DE REGLAS

(R1: $p \wedge q \rightarrow s$)

R2: $r \rightarrow t$

R3: $s \wedge t \rightarrow u$

R4: $s \wedge r \rightarrow v$



BH: p q r s



*CC = { R2, R4 }

* R2 $\leftarrow$ RC

* DISPARA R2

* BH $\leftarrow$ t

* R2 APLICADA

SE basados en reglas de producción

Ejemplo Forward Chaining: Tercer paso

BASE DE REGLAS

(R1: $p \wedge q \rightarrow s$)

(R2: $r \rightarrow t$)

R3: $s \wedge t \rightarrow u$

R4: $s \wedge r \rightarrow v$



*CC = { R3, R4 }

* R3 $\leftarrow$ RC

* DISPARA R3

* BH $\leftarrow$ u

* R3 APLICADA



BH: p q r s t

SE basados en reglas de producción

Ejemplo Forward Chaining: Cuarto paso

BASE DE REGLAS

(R1: $p \wedge q \rightarrow s$)

(R2: $r \rightarrow t$)

(R3: $s \wedge t \rightarrow u$)

R4: $s \wedge r \rightarrow v$



BH: p q r s t u



*CC = {R4}

* R4 $\leftarrow$ RC

* DISPARA R4

* BH $\leftarrow$ v

* R4 APLICADA

SE basados en reglas de producción

Ejemplo Forward Chaining: Quinto paso

BASE DE REGLAS

(R1: $p \wedge q \rightarrow s$)

(R2: $r \rightarrow t$)

(R3: $s \wedge t \rightarrow u$)

(R4: $s \wedge r \rightarrow v$)



*CC = { }

* FIN



BH: p q r s t u v

SE basados en reglas de producción

Forward Chaininig en Kappa-PC.



ESTRATEGIAS:

* SELECTIVE

* DEPTH FIRST

* BREATH FIRST

* BEST FIRST

Ciclo de Forward Chaining en Kappa-PC.

1) Se evalúa el hecho actual en la AGENDA.



⇒ $CC = \{ \text{REGLAS QUE MENCIONAN EL HECHO EN SUS PREMISAS} \} \rightarrow \text{LISTA DE REGLAS (LR)}.$

⇒ Se quita el hecho de la AGENDA.

2) Se evalúan las REGLAS de LR en orden, al disparar una:



⇒ Se agregan (si corresponde) hechos en la AGENDA.

⇒ Se quita la REGLA de la LISTA DE REGLAS.

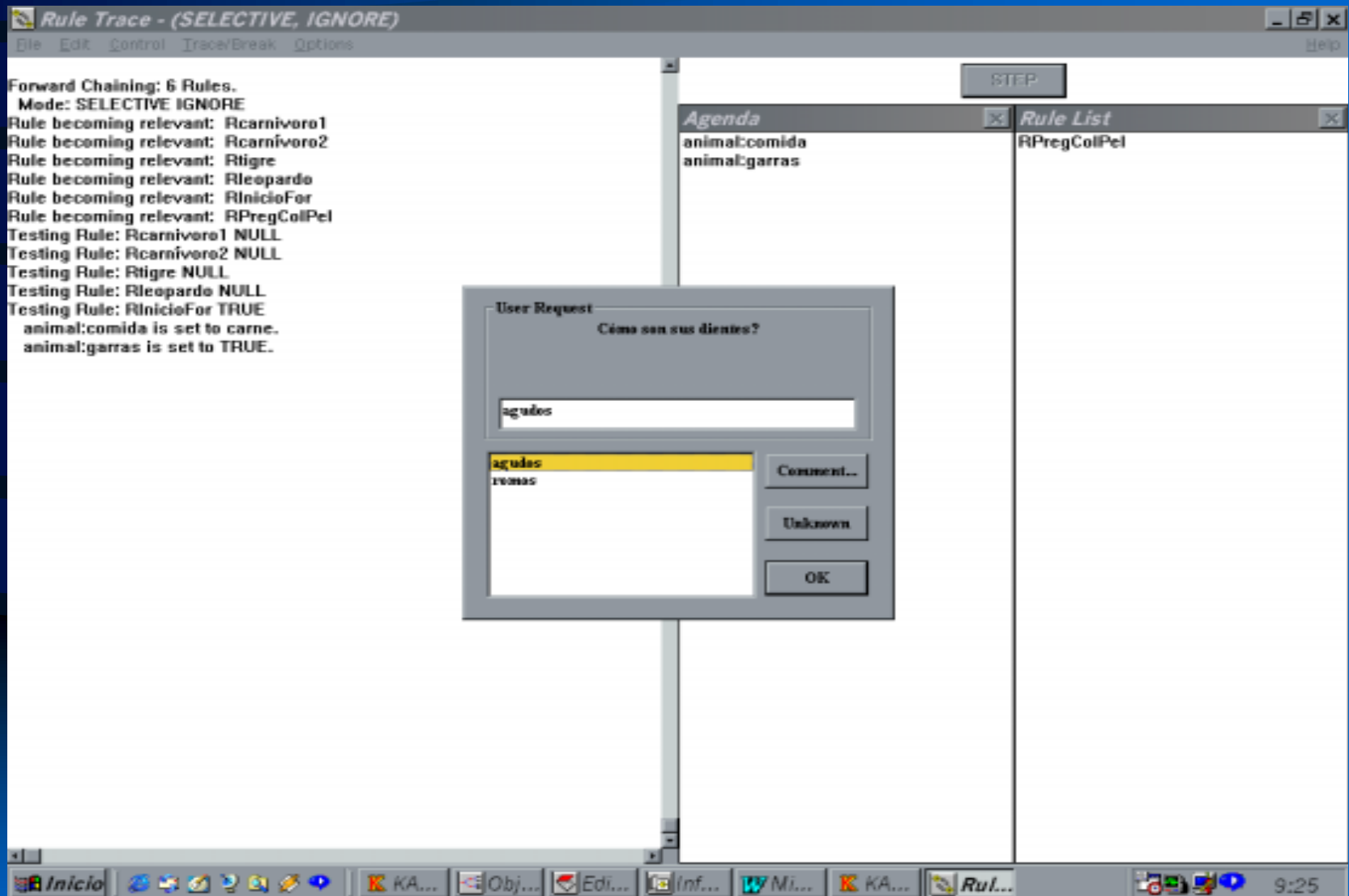
3) Se evalúa finalización.



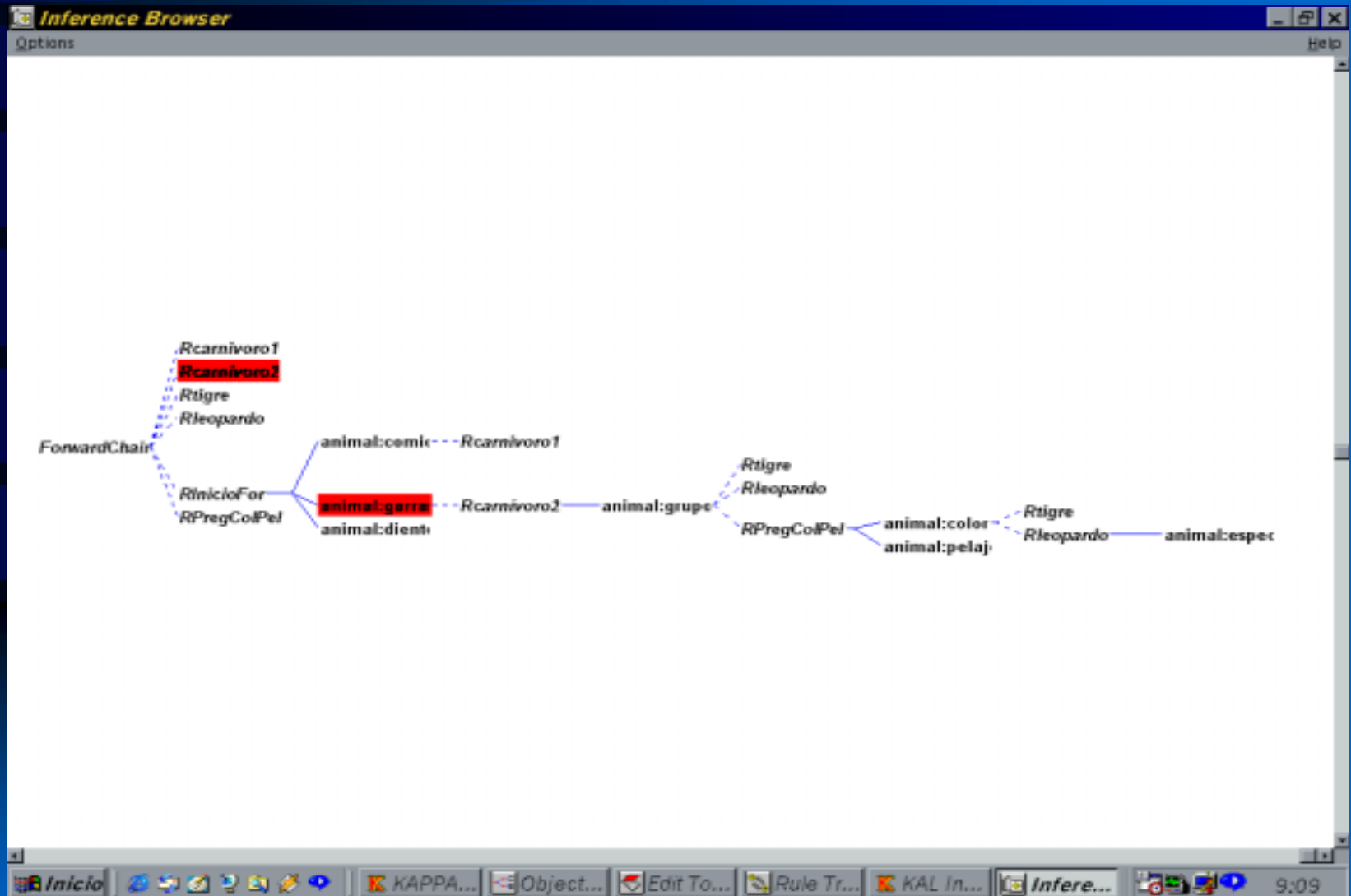
⇒ Si hay GOAL y es TRUE $\rightarrow$ FIN.

⇒ Si no es así $\rightarrow$ Otro Ciclo.

Forward Chaining en Kappa-PC.



Forward Chaining en Kappa-PC.



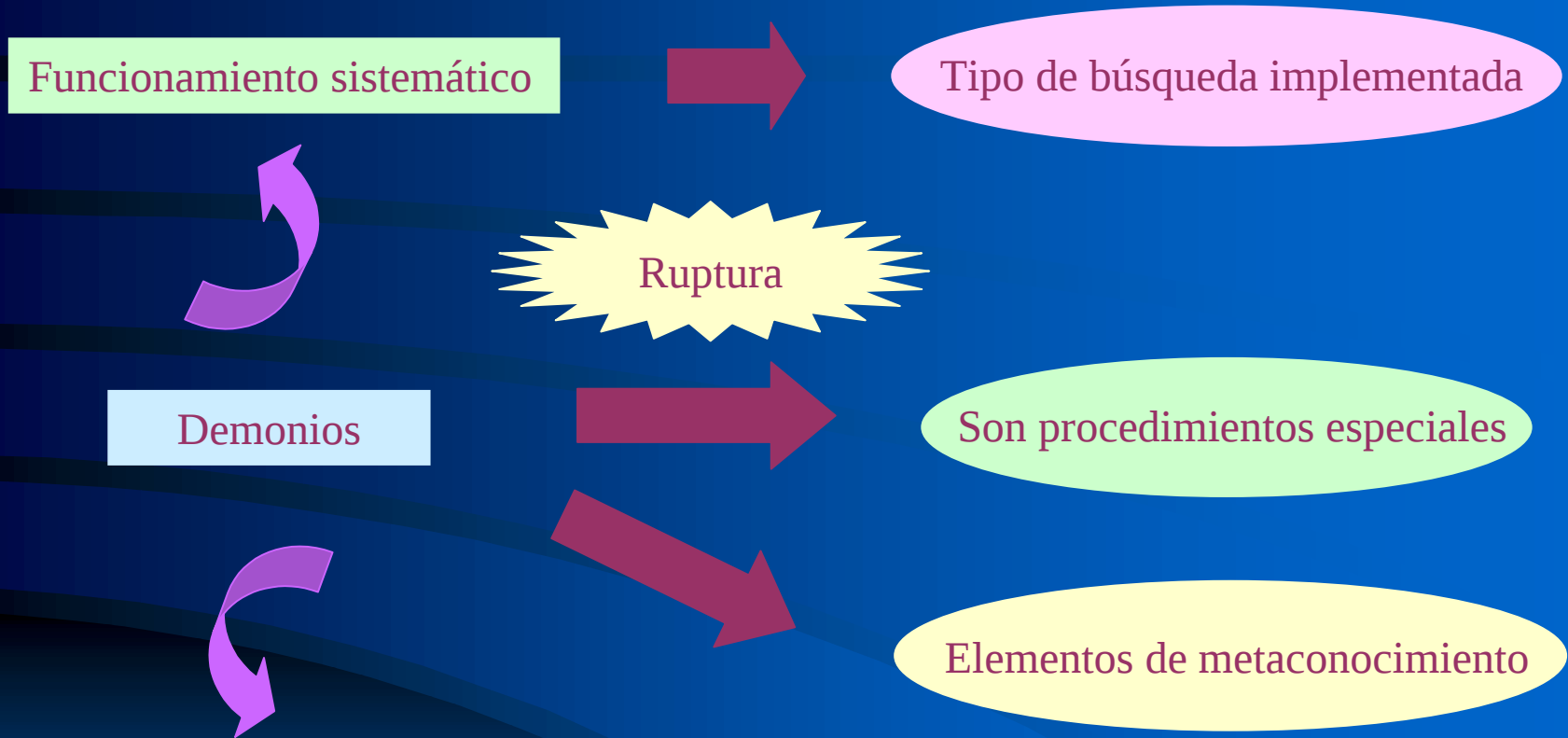
Forward Chaining en Kappa-PC.

- * Para lanzarlo deben garantizarse hechos en la agenda (Assert inicial o Forward con [NOASSERT]).
- * Se debe asegurar la posterior introducción de hechos en la agenda durante la inferencia.
- * Las estrategias permiten recorrer el árbol de búsqueda de diferentes formas.
- * Las prioridades en las reglas permiten personalizar las estrategias.

Forward Chaininig en Kappa-PC.

- * Los objetivos no son imprescindibles pero permiten “cortar” la búsqueda.
- * La búsqueda puede polarizarse lanzando el Forward con diferentes grupos de reglas.
- * Pueden introducirse “patrones” para inferir sobre distintos objetos.

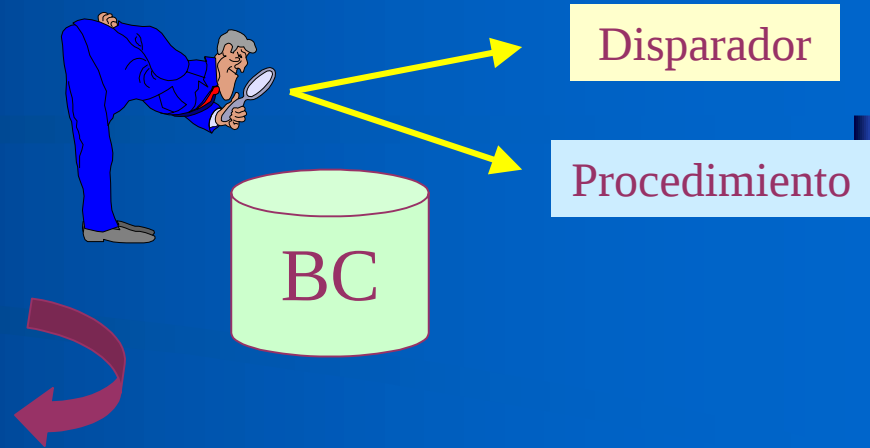
Motor de Inferencias: Estructuras de control



Se invocan a partir de cierto conocimiento deducido durante la ejecución de la aplicación.

Estructuras de control: Demonios

El demonio es una estructura de control que vigila constantemente el comportamiento del SE y se activa cuando encuentra determinadas condiciones en la BC.



✿ Difieren de los procedimientos tradicionales, en que estos últimos responden a un llamado específico que los identifica y pone en acción.

Personal Consultant-xx

Forward Chaining es un demonio, manejado mediante la propiedad ANTECEDENT de las reglas.

Kappa-PC

Monitores: Demonios que se activan a partir de cómo maneja el sistema los objetos con los que se los asocia.

Demonios en KAPPA-PC.

↪ ***If Needed:*** Cuando el sistema necesita el slot pero su valor no se conoce. Pensado para asignar valores por defecto → Debe retornar ese valor.

↪ ***When Accesed:*** Cuando el sistema accede al slot sin importar si tiene valor asignado. Lleva como argumentos (slot, valor) y debe retornar valor.

↪ ***Before Change:*** Antes de que se cambie el valor del slot. Lleva como argumentos (slot, valor) y debe retornar valor.

↪ ***After Change:*** Después de cambiar el valor del slot.

Sistemas expertos: Metaconocimiento

- ✓ Es el conocimiento estratégico vinculado a la utilización del conocimiento del dominio del Sistema Experto.
- ✓ Es conocimiento que predica sobre el conocimiento involucrado en el sistema.

Métodos de implementación:

Meta-reglas

- ✓ Activación o no de distintos grupos de reglas específicas.
- ✓ Ordenación de las reglas dentro de los grupos.

Funciones

- ✓ Definición de la búsqueda o no de conceptos asociados.

✓ PC-PLUS → FINDOUT - NOFINDOUT

- ✓ Posibilidad de asignarles “pesos” numéricos.

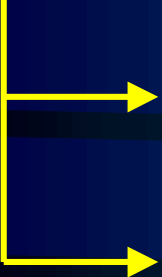
Orden de las reglas

✓ PC-PLUS → UTILITY.

✓ KAPPA-PC → PRIORITY

Sistemas expertos: Metaconocimiento

Enriquece y aporta a la performance de un Sistema Experto y por sus características incluye:



✓ Elementos en el Motor de Inferencias: Demonios.

✓ Elementos en la Base de Conocimientos: Metareglas, PRIORITY, etc.

Ejemplos de Metareglas en SEXP-1

IF Litología_Principal = Arenisca THEN Tratar sólo el grupo de reglas asociadas.

IF Litología_Principal = Roca_Ignea THEN NoAsk (Lista de valores)