

# Sistemas Inteligentes

## Tema 4 – Sistemas de Producción

*Curso 2020/2021*

Facultade de Informática  
**Universidade da Coruña**

# Objetivos

- Conocer la arquitectura típica de un sistema de producción, distinguiendo los distintos componentes.
- Diferenciar los sistemas de producción de las aproximaciones algorítmicas tradicionales.
- Percibir claramente la separación entre el conocimiento y los mecanismos de control del sistema, como una característica fundamental de estos sistemas.
- Estudiar la dinámica básica de los sistemas de producción, en función de la estrategia de control.
- Comprender las ventajas y limitaciones de utilización de los sistemas de producción.
- Valorar qué tipos de problemas son adecuados para resolver utilizando sistemas de producción.

- Definición
- Arquitectura
  - Base de conocimientos
  - Memoria activa
  - Motor de inferencias
- Dinámica de los sistemas de producción
  - Dirigidos por los datos
  - Dirigidos por las metas
- Ciclo básico de los sistemas de producción
  - Fase de decisión o selección de reglas
  - Fase de ejecución

- **Moret, V., Alonso, A., Cabrero, M. J., Guijarro, B., y Mosqueira, E. (2005).** Fundamentos de Inteligencia Artificial (2ªed). Servicio de Publicacións. Universidade da Coruña.
- **Borrajo, D., Juristo, N., Martínez, V. y Pazos, J. (1993).** Inteligencia Artificial: Métodos y Técnicas. Editorial Centro de Estudios Ramón Areces eds.
- **Negnevitsky,** Artificial Intelligence: a guide of intelligent systems.", Addison-Wesley, 2002
- **Mira, J., Delgado, A.E., Boticario, J.G., Díez, F.J. (1995).** Aspectos básicos de la Inteligencia Artificial. Sanz y Torres.
- **Galán, S.F., González, J. y Mira, J. (1998).** Problemas Resueltos de Inteligencia Artificial Aplicada. Búsqueda y Representación. Addison-Wesley.

## ■ Programas secuenciales:

- dependientes de los datos, condiciones iniciales, parámetros, respuestas de los usuarios, resultados de cálculos anteriores,...
- el flujo de control y la utilización de los datos especificados de manera rígida por el programa.
- el defecto específico de estos programas es su secuencialidad:
  - la bifurcación sólo se efectúa en puntos y caminos explícitamente previstos en el código del programa.
- en entornos cambiantes, la bifurcación será la norma, no la excepción, ya sea a partir de un nuevo estímulo, en función de sus propiedades y contenidos, o bien, en ausencia de nuevos estímulos, será a partir del contexto actual, historia o último estado.

- Programación basada en eventos:
  - adecuada para situaciones en las que continuamente están surgiendo nuevos estímulos, bien originados por el entorno externo o bien generados internamente por el programa.
  - el programa responde directamente a un amplio rango de sucesos, algunos imprevistos, en lugar de hacerlo a datos esperados.
  - no usan estructuras de control inflexibles y especificadas de antemano.
  - reconocen patrones en los datos, y seleccionan trozos de código en el sistema para que se activen.
  - Sistemas de inferencia dirigidos por patrones (SIDP)

- Los SIDP tienen estructura modular
- Cada módulo es responsable de detectar las distintas situaciones posibles, reconociendo patrones en los datos, y de responder de forma adecuada a las mismas.
- Los módulos se dividen funcionalmente en *antecedente* o lado izquierdo y *consecuente* o lado derecho
- El *antecedente* efectúa todos los accesos a los datos, para verificarlos y equipararlos con los patrones “plantilla” del módulo.
- El *consecuente* efectúa la escritura o modificación de datos.
- Tales módulos se denominan “reglas”, y los SIDP compuestos por conjuntos de reglas, Sistemas basados en Reglas (SBR) o Sistemas de Producción.

- Definición:
  - Sistemas inteligentes basados en reglas que operan frente a una base de hechos con mecanismos de emparejamiento formando parte explícita de su arquitectura.
  
- Se clasifican en dos categorías según su estructura de control:
  - Sistemas dirigidos por los datos
  - Sistemas dirigidos por los objetivos



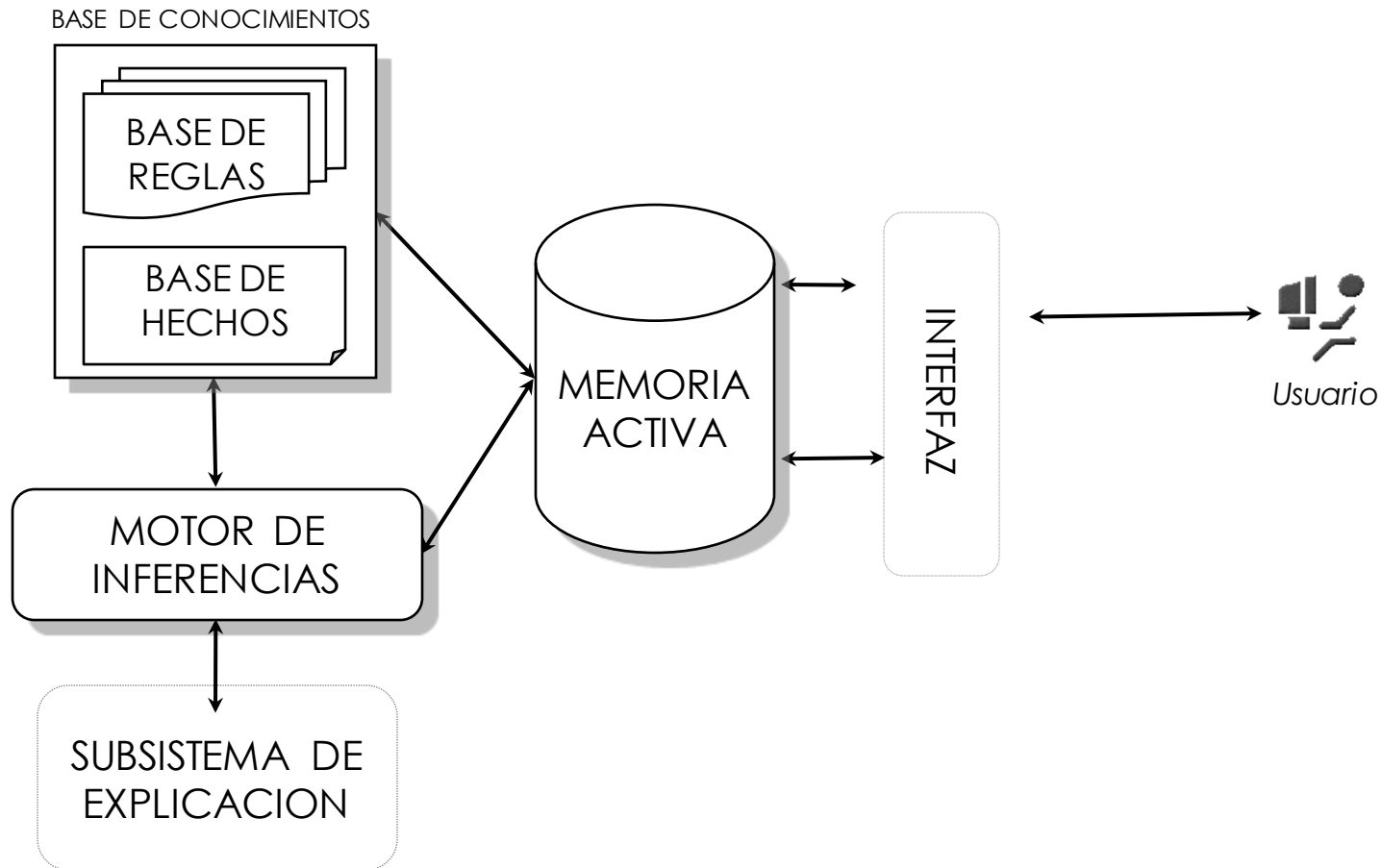
## ■ Sistemas dirigidos por los datos

- Las inferencias se obtienen cuando los antecedentes de alguna (o más de una) de sus reglas de producción se emparejan con, al menos, una parte de los hechos que describen el estado actual.
- Cuando esto ocurre, se dice que la regla en cuestión se ha activado, y está en condiciones de ser ejecutada. Su ejecución o no dependerá de la estrategia de exploración elegida.
- Son menos específicos, porque ejecutarán todas las reglas disponibles en función de la información introducida.

# Clasificación de Sistemas de Producción

- **Sistemas dirigidos por los objetivos**
  - Tanto antecedentes como consecuentes de las reglas deben ser considerados como aserciones sobre los datos. En este caso, la activación de las reglas tiene lugar por medio de un encadenamiento regresivo, y el emparejamiento se efectúa a través de las conclusiones de las reglas.
  - Para alcanzar una determinada meta hay que configurar un proceso evocativo en el que, de forma recursiva, se van estableciendo los antecedentes de las metas como submetas de orden inferior.
  - Son más específicos, porque la ejecución lleva implícito un proceso de búsqueda.

# Arquitectura de Sistemas de Producción

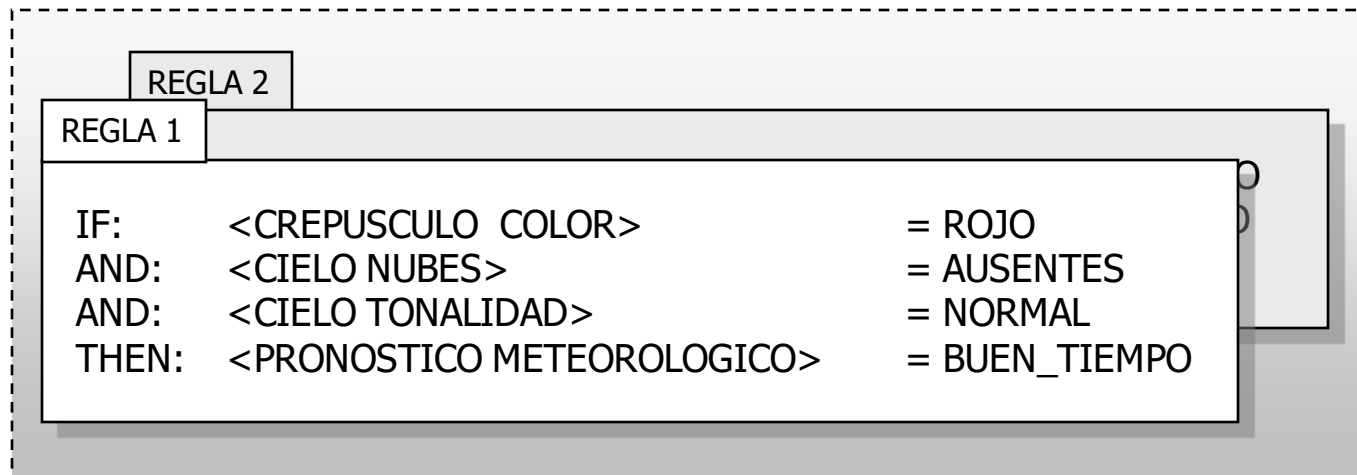


## ■ Características

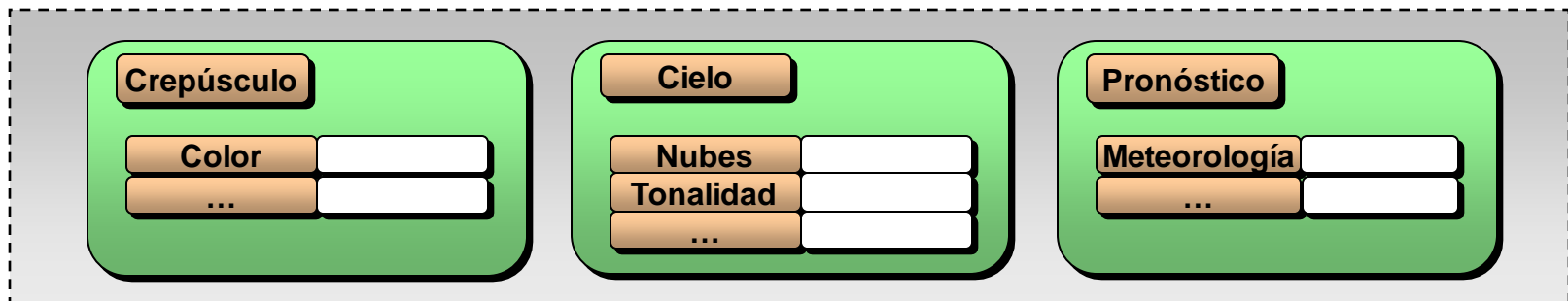
- Describe el universo de discurso o dominio en el cual el sistema de producción tiene que plantear soluciones.
- Está constituida por bases de hechos (BH) y por bases de reglas (BR).
- Las **bases de hechos** forman el esqueleto declarativo del sistema de producción, y su misión es la de articular a todos los hechos potencialmente relevantes del dominio.
- Las **bases de reglas** constituyen el esqueleto procedimental del sistema de producción, y a través de ellas se posibilita la construcción de los circuitos inferenciales que nos van a permitir obtener conclusiones válidas.
- La estructura de las bases de hechos y de las bases de reglas debe ser tal que ambas entidades puedan "comprenderse" entre sí.

# Base de Conocimientos: Ejemplo

## BASE DE REGLAS

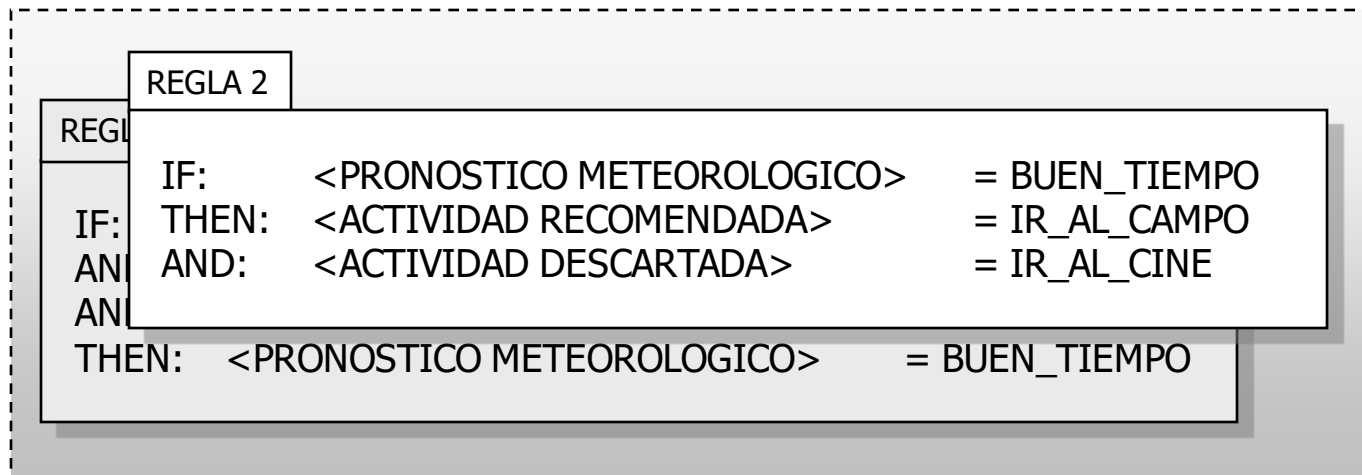


## BASE DE HECHOS



# Base de Conocimientos: Ejemplo

## BASE DE REGLAS



## BASE DE HECHOS

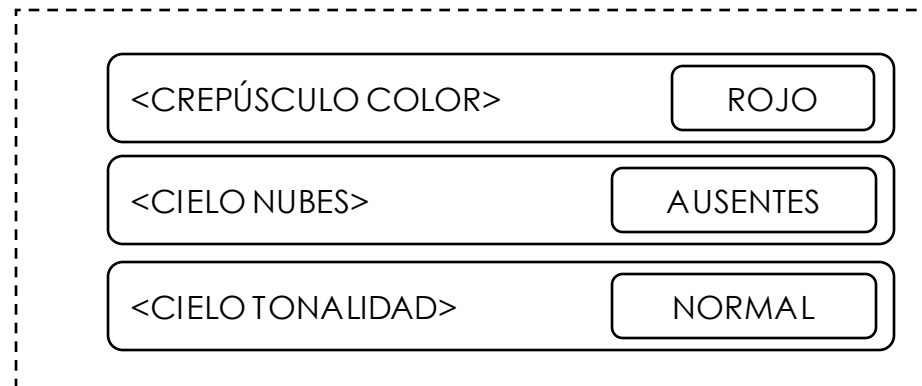


## Memoria Activa (MA)

- Es la estructura que contiene toda la información de naturaleza estática necesaria para resolver un problema concreto. Esta información incluye:
  - datos iniciales del problema
  - datos incorporados con posterioridad
  - hechos establecidos durante los procesos inferenciales
  - hipótesis de trabajo, metas o submetas que todavía no han sido establecidas
- Almacena todos los cambios de estado de nuestro sistema, de forma que representa siempre nuestro estado actual.
- Es la responsable de interaccionar con el mundo exterior, aceptando la entrada de información de naturaleza no inferencial.
- Es el foco permanente de atención de las reglas del sistema.

# Memoria Activa: Ejemplo

## MEMORIA ACTIVA (DATOS)



## MEMORIA ACTIVA (HIPÓTESIS)





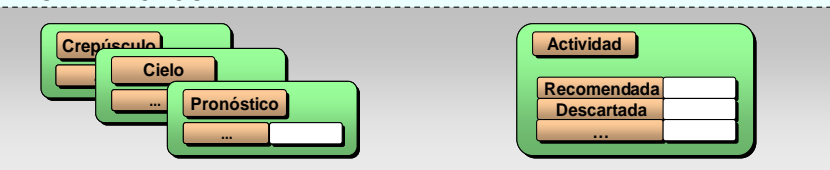
# Memoria Activa: Ejemplo

## BASE DE CONOCIMIENTOS

### BASE DE REGLAS

REGLA 2			
REG	IF:	<PRONOSTICO METEOROLOGICO>	= BUEN_TIEMPO
IF:	THEN:	<ACTIVIDAD RECOMENDADA>	= IR_AL_CAMPO
AN	AND:	<ACTIVIDAD DESCARTADA>	= IR_AL_CINE
AN	THEN:	<PRONOSTICO METEOROLOGICO>	= BUEN_TIEMPO

### BASE DE HECHOS



## MEMORIA ACTIVA (DATOS)

< CREPÚSCULO COLOR >	ROJO
< CIELO NUBES >	AUSENTES
< CIELO TONALIDAD >	NORMAL

## MEMORIA ACTIVA (HIPÓTESIS)

< ACTIVIDAD RECOMENDADA >	IR AL CAMPO
---------------------------	-------------

- Lanzamos un proceso inferencial dirigido por los objetivos (el *motor de inferencias* investiga las hipótesis de la *memoria activa* no establecidas <ACTIVIDAD RECOMENDADA> = IR AL CAMPO)

## Memoria Activa: Ejemplo

- Investigamos las reglas que concluyen algo sobre <ACTIVIDAD RECOMENDADA>, en este caso la Regla 2.

REGLA 2		
IF:	<PRONOSTICO METEOROLOGICO>	= BUEN_TIEMPO
THEN:	<ACTIVIDAD RECOMENDADA>	= IR_AL_CAMPO
AND:	<ACTIVIDAD DESCARTADA>	= IR_AL_CINE

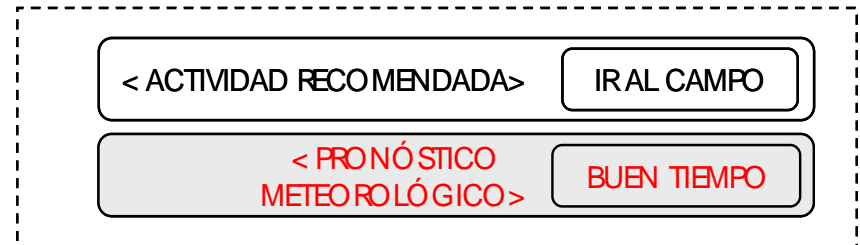
- Investigamos el antecedente de la Regla 2 para ver si con la información de la *memoria activa* se puede concluir algo

REGLA 2		
IF:	<PRONOSTICO METEOROLOGICO>	= BUEN_TIEMPO
THEN:	<ACTIVIDAD RECOMENDADA>	= IR_AL_CAMPO
AND:	<ACTIVIDAD DESCARTADA>	= IR_AL_CINE

# Memoria Activa: Ejemplo

- Como en la memoria activa no hay información sobre <PRONOSTICO METEOROLOGICO> se incluye como nueva hipótesis en la *memoria activa*
- La nueva hipótesis aparece como conclusión de Regla 1 y obliga a investigar su antecedente.
- La Regla 1 puede ejecutarse porque sus premisas aparecen en la *memoria activa* y sus valores asociados son los que emparejan.

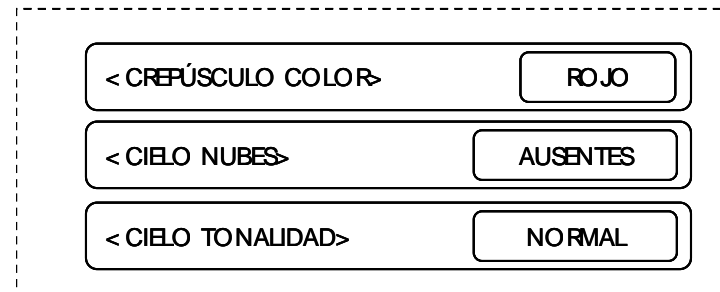
## MEMORIA ACTIVA (HIPÓTESIS)



## REGLA 1

IF:	<CREPUSCULO COLOR>	= ROJO
AND:	<CIELO NUBES>	= AUSENTES
AND:	<CIELO TONALIDAD>	= NORMAL
THEN:	<PRONOSTICO METEOROLOGICO>	= BUEN_TIEMPO

## MEMORIA ACTIVA (DATOS)



- Después de ejecutar la Regla 1 la submeta se confirma como hecho y se actualiza la *memoria activa*

## MEMORIA ACTIVA (DATOS)

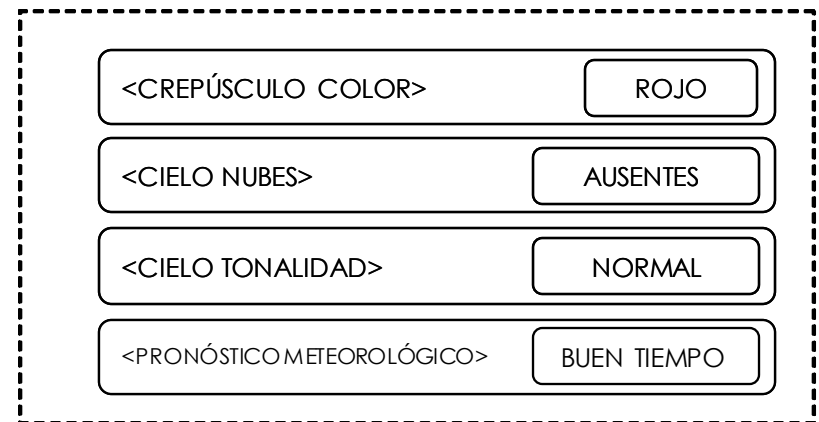
< CREPÚSCULO COLOR >	ROJO
< CIELO NUBES >	AUSENTES
< CIELO TONALIDAD >	NORMAL
< PRONÓSTICO METEOROLÓGICO >	BUEN TIEMPO

## MEMORIA ACTIVA (HIPÓTESIS)

< ACTIVIDAD RECOMENDADA >	IR AL CAMPO
---------------------------	-------------

- La Regla 2 está en condiciones de ser ejecutada ya que la *memoria activa* verifica completamente su antecedente.

MEMORIA ACTIVA (DATOS)



MEMORIA ACTIVA (HIPÓTESIS)



REGLA 2

IF:	<PRONOSTICO METEOROLOGICO>	= BUEN TIEMPO
THEN:	<ACTIVIDAD RECOMENDADA>	= IR AL CAMPO
AND:	<ACTIVIDAD DESCARTADA>	= IR AL CINE

- Se produce una nueva actualización de la *memoria activa*. Al no haber más hipótesis que confirmar o descartar el proceso termina.

MEMORIA ACTIVA (DATOS)

<CREPÚSCULO COLOR>	ROJO
<CIELO NUBES>	AUSENTES
<CIELO TONALIDAD>	NORMAL
<PRONÓSTICO METEOROLÓGICO>	BUEN TIEMPO
<ACTIVIDAD RECOMENDADA>	IR AL CAMPO
<ACTIVIDAD DESCARTADA>	IR AL CINE

- Cuando el proceso se detiene, la memoria activa contiene una descripción del estado final del problema, incluyendo datos, hechos e hipótesis
  - Los hechos y los datos de la memoria activa corresponden a entidades de la base de hechos, pero con valores concretos asociados.
  - Los valores de las trayectorias correspondientes a los hechos de la base se asignan a través de un proceso inferencial.
  - Los datos representan información que procede directamente del mundo exterior.
  - Las hipótesis son trayectorias completas (incluyendo valores), cuya veracidad se desea investigar.

# Motor de Inferencias (MI)

- Intérprete de reglas + Estrategia de control. Separación con el conocimiento
- Funciones:
  - Examinar la memoria activa y determinar qué reglas deben ejecutarse (estrategia de búsqueda + resolución de conflictos). Encontrar conexiones entre estados iniciales del problema y estados solución.
    - desde las premisas a las conclusiones
    - desde las soluciones a los datos iniciales
    - desde ambos simultáneamente
  - Controlar y organizar el proceso de ejecución de las reglas seleccionadas en el paso anterior
  - Actualizar la memoria activa cuando sea preciso (hechos, metas y submetas)
  - Asegurar el autoconocimiento del sistema (reglas activadas, reglas ejecutadas, últimos hechos incorporados a la memoria, prioridades de reglas...)



- El intérprete no es más que un programa secuencial cuya misión es determinar el siguiente paso a ejecutar.
- La estrategia de control es el mecanismo que examina la memoria activa y determina qué regla disparar, a través de los llamados ciclos básicos del sistema, y en función de ciertos parámetros como:
  - criterios de activación elegidos,
  - estrategias de búsqueda implementadas y
  - dirección de tránsito por el espacio de estados

## ■ Base de reglas

BR = [R1, R2, R3, R4, R5, R6, R7]

R1:	IF (X and Y)	THEN	Z
R2:	IF (C and D)	THEN	G
R3:	IF (E and V)	THEN	H
R4:	IF (A and B)	THEN	C
R5:	IF (F or G)	THEN	X
R6:	IF (Z and B)	THEN	V
R7:	IF (E and C)	THEN	F

## Memoria Activa

M0 = <A, B, D, Y, E, (H)>

Datos del problema o  
hechos previamente  
demostrados

Hipótesis sin confirmar

- Consideraremos dos casos con diferentes restricciones en el motor de inferencias

- Restricciones del motor de inferencias:
  - Encadenamiento progresivo
  - Activación de todas las reglas que emparejen con la memoria activa
  - Búsqueda en profundidad con ejecución de la primera regla de las que hayan sido activadas más recientemente
  - No ejecutar dos veces la misma regla
  - Terminar cuando H sea un hecho demostrado e incorporado a la memoria activa

## ■ Funcionamiento

- El emparejador examina los antecedentes de las reglas y selecciona aquellas que se corresponden con hechos y datos de la memoria activa.
- Se ejecutará la primera de las reglas activadas, y el resultado se incorporará a la memoria activa.
- Se comprobará si en la memoria activa aparece H como hecho demostrado.
- En caso contrario se repetirá un nuevo ciclo hasta:
  - encontrar la solución
  - todas las reglas han sido ejecutadas y no se ha llegado a la solución

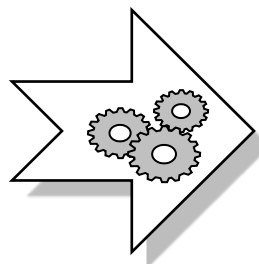
# Ejemplo dinámico 1: Ciclo 1

## Base de Reglas

R1: IF (X and Y) THEN Z  
R2: IF (C and D) THEN G  
R3: IF (E and V) THEN H  
**R4: IF (A and B) THEN C**  
R5: IF (F or G) THEN X  
R6: IF (Z and B) THEN V  
R7: IF (E and C) THEN F

## Memoria Activa

M0 = <A, B, D, Y, E, (H)>

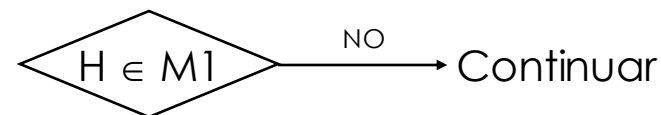


Reglas activadas Reglas ejecutadas

R4	Ciclo 1	R4	Ciclo 1
----	---------	----	---------

## Memoria Activa

M1 = <A, B, D, Y, E, (H), C>



# Ejemplo dinámico 1: Ciclo 2

## Base de Reglas

R1: IF (X and Y) THEN Z  
**R2: IF (C and D) THEN G**  
 R3: IF (E and V) THEN H  
 R4: IF (A and B) THEN C ✓  
 R5: IF (F or G) THEN X  
 R6: IF (Z and B) THEN V  
**R7: IF (E and C) THEN F**

## Memoria Activa

M1 =  
 <A, B, D, Y, E, (H), C>

## Reglas activadas

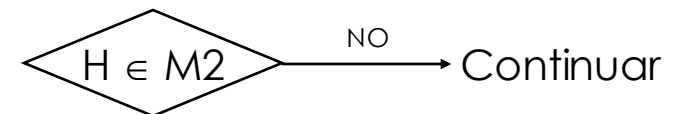
R2	Ciclo 2
R7	Ciclo 2
	Ciclo 1

## Reglas ejecutadas

R2	Ciclo 2
R4	Ciclo 1

## Memoria Activa

M2 =  
 <A, B, D, Y, E, (H), C, G>



# Ejemplo dinámico 1: Ciclo 3

## Base de Reglas

R1:	IF (X and Y)	THEN	Z
R2:	IF (C and D)	THEN	G ✓
R3:	IF (E and V)	THEN	H
R4:	IF (A and B)	THEN	C ✓
<b>R5:</b>	<b>IF (F or G)</b>	<b>THEN</b>	<b>X</b>
R6:	IF (Z and B)	THEN	V
R7:	IF (E and C)	THEN	F ✓

## Memoria Activa

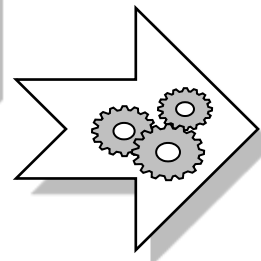
M2 =  
<A, B, D, Y, E, (H), C, G>

## Reglas activadas

R5	Ciclo 3
R7	Ciclo 2
	Ciclo 1

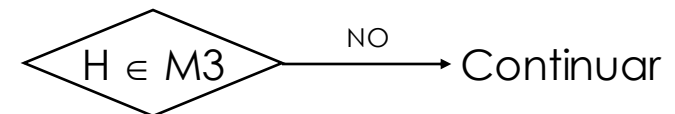
## Reglas ejecutadas

R5	Ciclo 3
R2	Ciclo 2
R4	Ciclo 1



## Memoria Activa

M3 =  
<A, B, D, Y, E, (H), C, G, X>



# Ejemplo dinámico 1: Ciclo 4

## Base de Reglas

<b>R1:</b>	<b>IF (X and Y) THEN</b>	<b>Z</b>
R2:	IF (C and D) THEN	G ✓
R3:	IF (E and V) THEN	H
R4:	IF (A and B) THEN	C ✓
R5:	IF (F or G) THEN	X ✓
R6:	IF (Z and B) THEN	V
R7:	IF (E and C) THEN	F ✓

## Memoria Activa

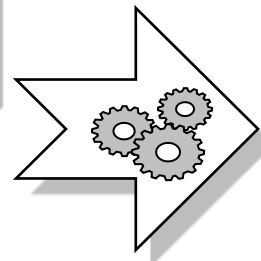
M3 =  
<A, B, D, Y, E, (H), C, G, X>

## Reglas activadas

<b>R1</b>	<b>Ciclo 4</b>
	<b>Ciclo 3</b>
<b>R7</b>	<b>Ciclo 2</b>
	<b>Ciclo 1</b>

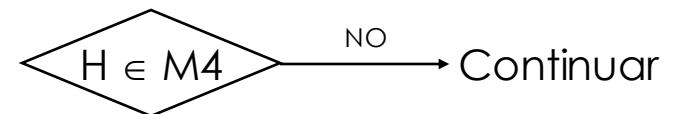
## Reglas ejecutadas

<b>R1</b>	<b>Ciclo 4</b>
<b>R5</b>	<b>Ciclo 3</b>
<b>R2</b>	<b>Ciclo 2</b>
<b>R4</b>	<b>Ciclo 1</b>



## Memoria Activa

M4 =  
<A, B, D, Y, E, (H), C, G, X, Z>





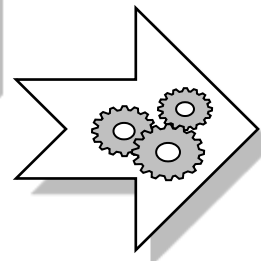
# Ejemplo dinámico 1: Ciclo 5

## Base de Reglas

R1:	IF (X and Y)	THEN	Z ✓
R2:	IF (C and D)	THEN	G ✓
R3:	IF (E and V)	THEN	H
R4:	IF (A and B)	THEN	C ✓
R5:	IF (F or G)	THEN	X ✓
<b>R6:</b>	<b>IF (Z and B)</b>	<b>THEN</b>	<b>V</b>
R7:	IF (E and C)	THEN	F ✓

## Memoria Activa

M4 =  
<A, B, D, Y, E, (H), C, G, X, Z>



## Reglas activadas

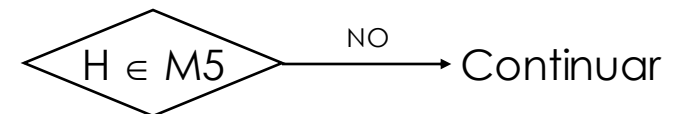
R6	Ciclo 5
	Ciclo 4
	Ciclo 3
R7	Ciclo 2
	Ciclo 1

## Reglas ejecutadas

R6	Ciclo 5
R1	Ciclo 4
R5	Ciclo 3
R2	Ciclo 2
R4	Ciclo 1

## Memoria Activa

M5 =  
<A, B, D, Y, E, (H), C, G, X, Z, V>



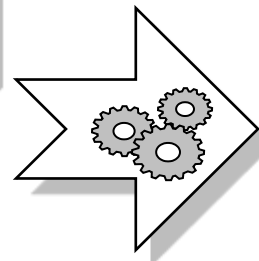
# Ejemplo dinámico 1: Ciclo 6

## Base de Reglas

R1:	IF (X and Y)	THEN	Z ✓
R2:	IF (C and D)	THEN	G ✓
<b>R3:</b>	<b>IF (E and V)</b>	<b>THEN</b>	<b>H</b>
R4:	IF (A and B)	THEN	C ✓
R5:	IF (F or G)	THEN	X ✓
R6:	IF (Z and B)	THEN	V ✓
R7:	IF (E and C)	THEN	F ✓

## Memoria Activa

M5 =  
<A, B, D, Y, E, (H), C, G, X, Z, V>



## Reglas activadas

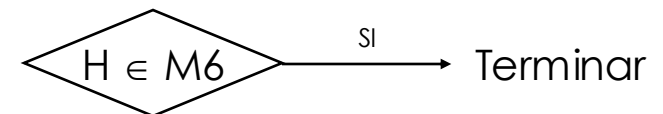
R3	Ciclo 6
	Ciclo 5
	Ciclo 4
	Ciclo 3
R7	Ciclo 2
	Ciclo 1

## Reglas ejecutadas

R3	Ciclo 6
R6	Ciclo 5
R1	Ciclo 4
R5	Ciclo 3
R2	Ciclo 2
R4	Ciclo 1

## Memoria Activa

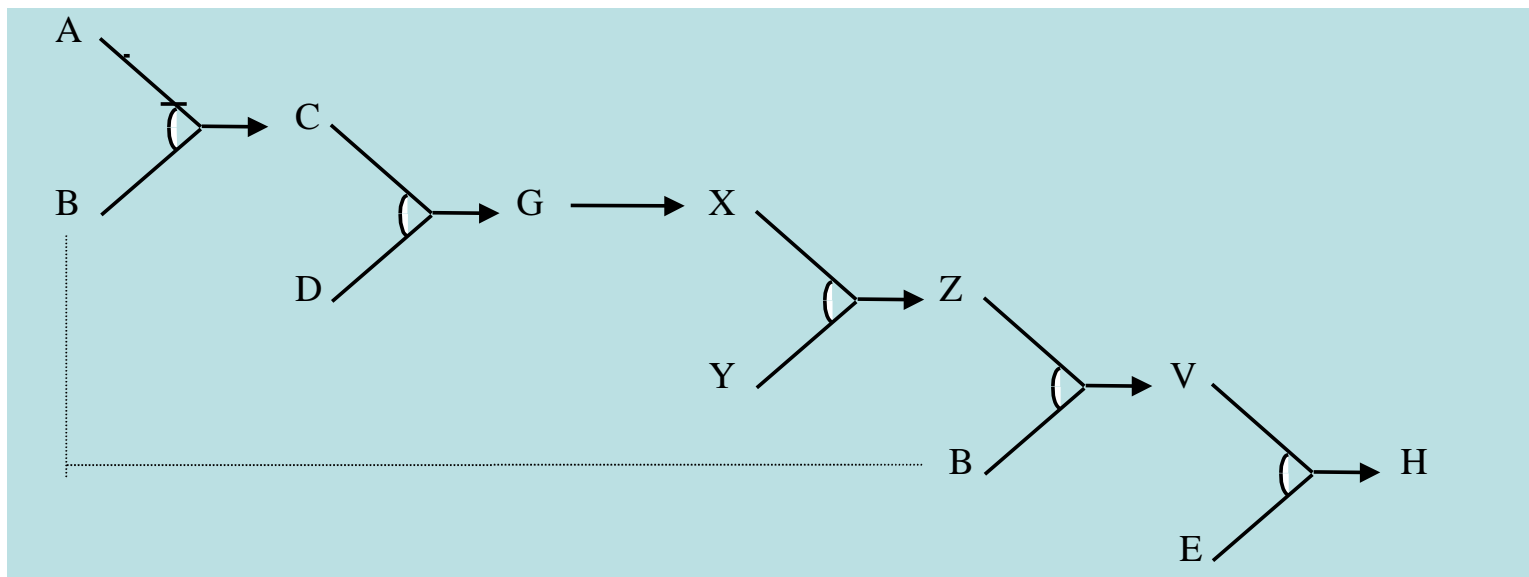
M6 =  
<A, B, D, Y, E, C, G, X, Z, V, H>



Número de ciclos del motor de inferencias = 6

# Ejemplo dinámico1

- Circuito inferencial generado:



- Restricciones del motor de inferencias:
  - Encadenamiento regresivo
  - Activación de todas las reglas que emparejen con la memoria activa
  - Búsqueda en anchura exhaustiva
  - No ejecutar dos veces la misma regla
  - Terminar cuando H sea un hecho demostrado e incorporado a la memoria activa

# Ejemplo dinámico 2

## ■ Funcionamiento

- El emparejador examinará las conclusiones de las reglas y seleccionará aquéllas que correspondan a hipótesis de la memoria activa.
- De este modo se van a ir generando subhipótesis que irán incorporándose a la memoria activa. La generación sucesiva de submetas se denomina retropropagación.
- Una vez que la retropropagación ha concluido, comienza la fase de ejecución.
- En la fase de ejecución las correspondientes conclusiones son inferidas a través de un proceso "hacia adelante", y por lo tanto la hipótesis inicial puede ser verificada.
- Esta situación ocurre cuando una regla es directamente ejecutable, y por lo tanto su conclusión puede ser establecida.

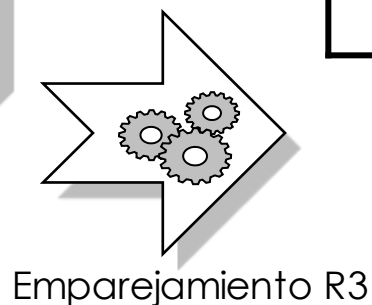
## Ejemplo dinámico 2: Ciclo 1

## Base de Reglas

R1: IF (X and Y) THEN Z  
 R2: IF (C and D) THEN G  
**R3: IF (E and V) THEN H**  
 R4: IF (A and B) THEN C  
 R5: IF (F or G) THEN X  
 R6: IF (Z and B) THEN V  
 R7: IF (E and C) THEN F

## Memoria Activa

$M0 = \langle A, B, D, Y, E, (H) \rangle$



## Reglas seleccionadas

Ciclo	1
Regla	R3

$$\left\{ \begin{array}{l} E \in M0 \\ \underline{V \notin M0} \end{array} \right\}$$

## Reglas ejecutadas

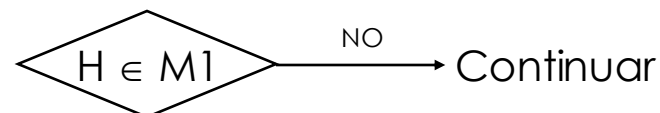
	Ciclo 1
--	---------

## Hipótesis

V	Ciclo 1
H	Ciclo 0

## Memoria Activa

$M1 = \langle A, B, D, Y, E, (H), (V) \rangle$



## Ejemplo dinámico 2: Ciclo 2

### Base de Reglas

R1: IF (X and Y) THEN Z  
 R2: IF (C and D) THEN G  
 R3: IF (E and V) THEN H ◀  
 R4: IF (A and B) THEN C  
 R5: IF (F or G) THEN X  
**R6: IF (Z and B) THEN V**  
 R7: IF (E and C) THEN F

### Memoria Activa

M1 = <A, B, D, Y, E, (H), (V)>

### Reglas seleccionadas

Ciclo	2	1	$\left\{ \begin{array}{l} B \in M1 \\ \underline{Z} \notin M1 \end{array} \right\}$
Regla	R6	R3	

### Reglas ejecutadas

	Ciclo 2
	Ciclo 1

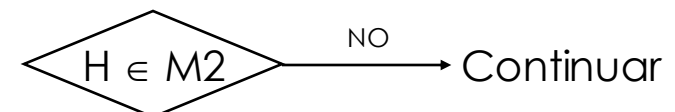
### Hipótesis

Z	Ciclo 2
V	Ciclo 1
H	Ciclo 0

### Memoria Activa

M2 = <A, B, D, Y, E, (H), (V), (Z)>

Emparejamiento  
R6



## Ejemplo dinámico 2: Ciclo 3

### Base de Reglas

<b>R1:</b>	<b>IF (X and Y) THEN</b>	<b>Z</b>
R2:	IF (C and D) THEN	G
R3:	IF (E and V) THEN	H ◀
R4:	IF (A and B) THEN	C
R5:	IF (F or G) THEN	X
R6:	IF (Z and B) THEN	V ◀
R7:	IF (E and C) THEN	F

### Memoria Activa

M2 = <A, B, D, Y, E, (H), (V), (Z)>

### Reglas seleccionadas

<b>Ciclo</b>	<b>3</b>	<b>2</b>	<b>1</b>
<b>Regla</b>	<b>R1</b>	<b>R6</b>	<b>R3</b>

$$\left\{ \begin{array}{l} Y \in M2 \\ \underline{X \notin M2} \end{array} \right\}$$

### Reglas ejecutadas

	<b>Ciclo 3</b>
	<b>Ciclo 2</b>
	<b>Ciclo 1</b>

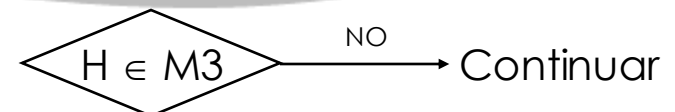
### Hipótesis

<b>X</b>	<b>Ciclo 3</b>
<b>Z</b>	<b>Ciclo 2</b>
<b>V</b>	<b>Ciclo 1</b>
<b>H</b>	<b>Ciclo 0</b>

Emparejamiento  
R1

### Memoria Activa

M3 = <A, B, D, Y, E, (H), (V), (Z), (X)>





## Ejemplo dinámico 2: Ciclo 4

### Base de Reglas

R1: IF (X and Y) THEN Z ◀  
 R2: IF (C and D) THEN G  
 R3: IF (E and V) THEN H ◀  
 R4: IF (A and B) THEN C  
**R5: IF (F or G) THEN X**  
 R6: IF (Z and B) THEN V ◀  
 R7: IF (E and C) THEN F

### Memoria Activa

M3 = <A, B, D, Y, E, (H), (V), (Z), (X)>

### Reglas seleccionadas

Ciclo	4	3	2	1
Regla	R5	R1	R6	R3

$$\left\{ \begin{array}{l} F \notin M3 \\ G \notin M3 \end{array} \right\}$$

### Reglas ejecutadas

	Ciclo 4
	Ciclo 3
	Ciclo 2
	Ciclo 1

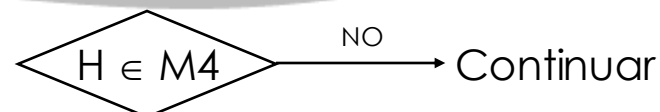
### Hipótesis

F G	Ciclo 4
X	Ciclo 3
Z	Ciclo 2
V	Ciclo 1
H	Ciclo 0

Emparejamiento  
R5

### Memoria Activa

M4 = <A, B, D, Y, E, (H),  
(V), (Z), (X), (F), (G)>



## Ejemplo dinámico 2: Ciclo 5

### Base de Reglas

R1:	IF (X and Y)	THEN	Z ◀
<b>R2:</b>	<b>IF (C and D)</b>	<b>THEN</b>	<b>G</b>
R3:	IF (E and V)	THEN	H ◀
R4:	IF (A and B)	THEN	C
R5:	IF (F or G)	THEN	X ◀
R6:	IF (Z and B)	THEN	V ◀
<b>R7:</b>	<b>IF (E and C)</b>	<b>THEN</b>	<b>F</b>

### Memoria Activa

M4 = <A, B, D, Y, E, (H),  
(V), (Z), (X), (F), (G)>

### Reglas seleccionadas

Ciclo	5	4	3	2	1
Regla	R2 R7	R5	R1	R6	R3

$$\left\{ \begin{array}{l} C \notin M4 \\ D \in M4 \\ E \in M4 \end{array} \right\}$$

### Hipótesis

C	Ciclo 5
F G	Ciclo 4
X	Ciclo 3
Z	Ciclo 2
V	Ciclo 1
H	Ciclo 0

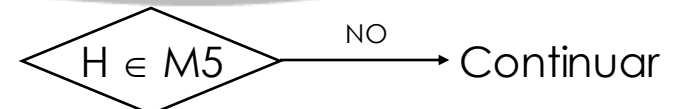
### Reglas ejecutadas

	Ciclo 5
	...
	Ciclo 1

Emparejamiento  
R2, R7

### Memoria Activa

M5 = <A, B, D, Y, E, (H),  
(V), (Z), (X), (F), (G), (C)>



## Ejemplo dinámico 2: Ciclo 6

### Base de Reglas

R1:	IF (X and Y)	THEN	Z ◀
R2:	IF (C and D)	THEN	G ◀
R3:	IF (E and V)	THEN	H ◀
<b>R4:</b>	<b>IF (A and B)</b>	<b>THEN</b>	<b>C</b>
R5:	IF (F or G)	THEN	X ◀
R6:	IF (Z and B)	THEN	V ◀
R7:	IF (E and C)	THEN	F ◀

### Memoria Activa

M5 = <A, B, D, Y, E, (H),  
(V), (Z), (X), (F), (G), (C)>

### Reglas seleccionadas

Ciclo	5	4	3	2	1
Regla	R2 R7	R5	R1	R6	R3

$\left\{ \begin{array}{l} A \in M5 \\ B \in M5 \end{array} \right\}$

### Reglas ejecutadas

R4	Ciclo 6
	Ciclo 5
	...
	Ciclo 1

### Hipótesis

F G	Ciclo 4
X	Ciclo 3
Z	Ciclo 2
V	Ciclo 1
H	Ciclo 0

Emparejamiento  
Ejecución R4

### Memoria Activa

M6 = <A, B, D, Y, E, (H),  
(V), (Z), (X), (F), (G), C>

$\Diamond H \in M6 \xrightarrow{\text{NO}} \text{Continuar}$

## Ejemplo dinámico 2: Ciclo 7

### Base de Reglas

R1:	IF (X and Y)	THEN	Z
R2:	IF (C and D)	THEN	G
R3:	IF (E and V)	THEN	H
R4:	IF (A and B)	THEN	C
R5:	IF (F or G)	THEN	X
R6:	IF (Z and B)	THEN	V
R7:	IF (E and C)	THEN	F

### Memoria Activa

M6 = <A, B, D, Y, E, (H),  
(V), (Z), (X), (F), (G), C>

### Reglas seleccionadas

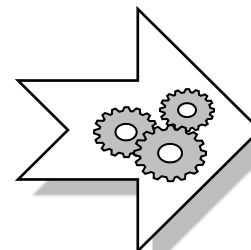
Ciclo	4	3	2	1
Regla	R5	R1	R6	R3

### Reglas ejecutadas

R2 R7	Ciclo 7
R4	Ciclo 6
	Ciclo 5
	...
	Ciclo 1

### Hipótesis

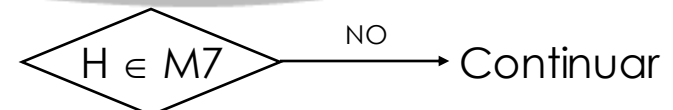
X	Ciclo 3
Z	Ciclo 2
V	Ciclo 1
H	Ciclo 0



Ejecución  
R2, R7

### Memoria Activa

M7 = <A, B, D, Y, E, (H),  
(V), (Z), (X), F, G, C>



## Ejemplo dinámico 2: Ciclo 8

## Base de Reglas

R1:	IF (X and Y)	THEN	Z
R2:	IF (C and D)	THEN	G
R3:	IF (E and V)	THEN	H
R4:	IF (A and B)	THEN	C
R5:	IF (F or G)	THEN	X
R6:	IF (Z and B)	THEN	V
R7:	IF (E and C)	THEN	F

## Memoria Activa

M7 = <A, B, D, Y, E, (H),  
(V), (Z), (X), F, G, C>

## Reglas seleccionadas

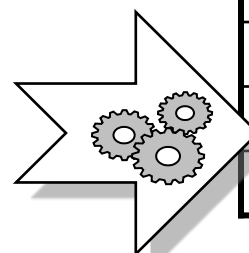
Ciclo	3	2	1
Regla	R1	R6	R3

## Reglas ejecutadas

R5	Ciclo 8
R2 R7	Ciclo 7
R4	Ciclo 6
	Ciclo 5
	...
	Ciclo 1

## Hipótesis

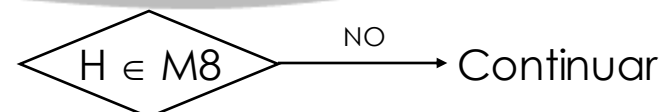
Z	Ciclo 2
V	Ciclo 1
H	Ciclo 0



Ejecución  
R5

## Memoria Activa

M8 = <A, B, D, Y, E, (H),  
(V), (Z), (X), F, G, C>



# Ejemplo dinámico 2: Ciclo 9

## Base de Reglas

R1:	IF (X and Y)	THEN	Z
R2:	IF (C and D)	THEN	G
R3:	IF (E and V)	THEN	H
R4:	IF (A and B)	THEN	C
R5:	IF (F or G)	THEN	X
R6:	IF (Z and B)	THEN	V
R7:	IF (E and C)	THEN	F

## Memoria Activa

M8 = <A, B, D, Y, E, (H),  
(V), (Z), X, F, G, C>

## Reglas ejecutadas

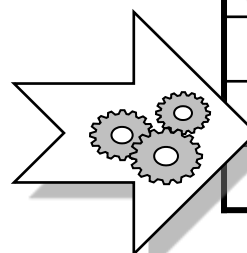
R1	Ciclo 9
R5	Ciclo 8
R2 R7	Ciclo 7
R4	Ciclo 6
	Ciclo 5
	...
	Ciclo 1

## Reglas seleccionadas

Ciclo	2	1
Regla	R6	R3

## Hipótesis

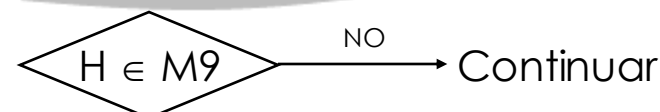
V	Ciclo 1
H	Ciclo 0



Ejecución  
R1

## Memoria Activa

M9 = <A, B, D, Y, E, (H),  
(V), Z, X, F, G, C>



## Ejemplo dinámico 2: Ciclo 10

### Base de Reglas

R1:	IF (X and Y)	THEN	Z
R2:	IF (C and D)	THEN	G
R3:	IF (E and V)	THEN	H
R4:	IF (A and B)	THEN	C
R5:	IF (F or G)	THEN	X
R6:	IF (Z and B)	THEN	V
R7:	IF (E and C)	THEN	F

### Memoria Activa

M9 = <A, B, D, Y, E, (H),  
(V), Z, X, F, G, C>

### Reglas ejecutadas

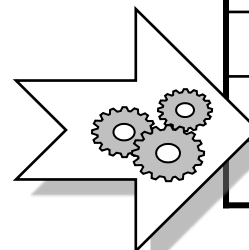
R6	Ciclo 10
R1	Ciclo 9
R5	Ciclo 8
R2 R7	Ciclo 7
R4	Ciclo 6
	Ciclo 5
	...
	Ciclo 1

### Reglas seleccionadas

Ciclo	1
Regla	R3

### Hipótesis

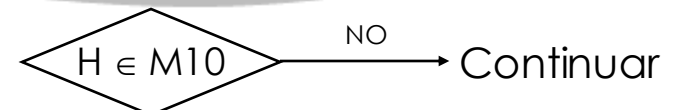
H	Ciclo 0
---	---------



Ejecución  
R6

### Memoria Activa

M10 = <A, B, D, Y, E, (H),  
V, Z, X, F, G, C>



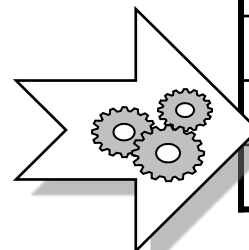
## Ejemplo dinámico 2: Ciclo 11

### Base de Reglas

R1:	IF (X and Y)	THEN	Z
R2:	IF (C and D)	THEN	G
R3:	IF (E and V)	THEN	H
R4:	IF (A and B)	THEN	C
R5:	IF (F or G)	THEN	X
R6:	IF (Z and B)	THEN	V
R7:	IF (E and C)	THEN	F

### Memoria Activa

M10 = <A, B, D, Y, E, (H),  
V, Z, X, F, G, C>



Ejecución  
R3

Reglas ejecutadas    Reglas seleccionadas

R3	Ciclo 11
R6	Ciclo 10
R1	Ciclo 9
R5	Ciclo 8
R2 R7	Ciclo 7
R4	Ciclo 6
	Ciclo 5
	...
	Ciclo 1

Ciclo	
Regla	

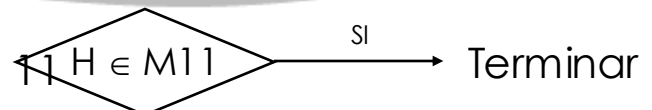
Hipótesis

--	--

### Memoria Activa

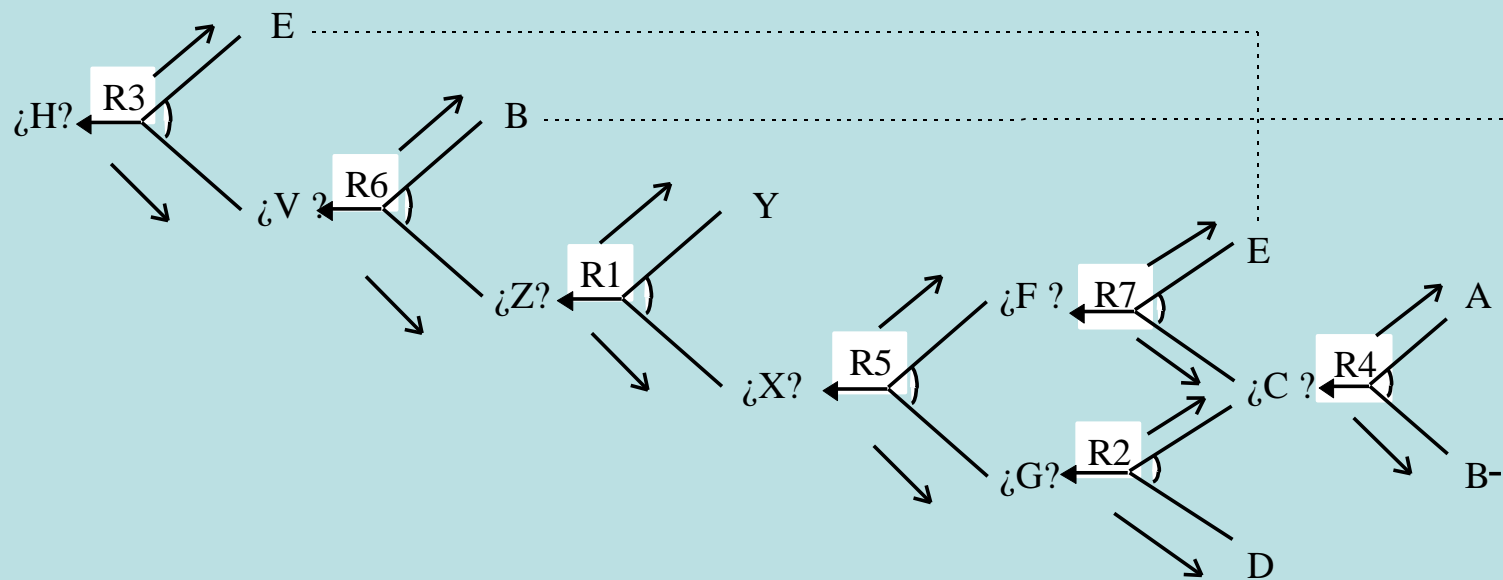
M11 = <A, B, D, Y, E, H,  
V, Z, X, F, G, C>

Número de ciclos del motor de inferencias = 11





## Circuito inferencial generado



PROCESO EVOCATIVO: FASE DE RETROPROPAGACIÓN

CASCADA INFERENCIAL: FASE DE EJECUCIÓN

- El ciclo básico está constituido por:
  - Fase de **decisión** o **selección de reglas**
    - Restricción
    - Equiparación
    - Resolución de conflictos
  - Fase de **acción** o **ejecución** de las reglas seleccionadas

### ■ Restricción.

- Trata de simplificar el proceso de equiparación.
- Elimina del foco de atención del motor de inferencias aquellas reglas que claramente no tienen nada que ver con el estado actual representado en la memoria activa del sistema.
- Ejemplo de restricción estática: los sistemas de producción suelen dividirse en varias bases de reglas y varias bases de hechos.
- Alternativa dinámica: emplea metaconocimiento (conocimiento sobre conocimiento).

- **Equiparación** (emparejamiento)
  - Se tratará de identificar qué reglas son potencialmente relevantes en el contexto del problema que queremos resolver
  - El resultado final es la obtención del denominado *conjunto conflicto* que, incluye todas las reglas potencialmente útiles en la resolución de nuestro problema

- **Resolución de conflictos:** decidir qué regla aplicar
  - La decisión final está fuertemente condicionada por la estrategia genérica de búsqueda.
  - Técnicas:
    - Uso de metarreglas. Por ejemplo: Las reglas proporcionadas por expertos tienen mayor prioridad que las de los novatos
    - Asignar prioridades a las reglas (en función del orden en la base de reglas o de valores numéricos directamente)
    - Usar la regla más específica puesto que procesa más información que una regla general
    - Elegir la regla que use los datos más recientes, con el fin de seguir una línea de razonamiento estable y sensible con la nueva información.

- Actualización de la memoria activa (nuevos hechos y/o hipótesis)
- Marcaje de las estructuras utilizadas
- Verificación de si continuar o no el proceso cíclico