

# Métodos de Inteligencia Artificial

---

L. Enrique Sucar (INAOE)

[esucar@inaoep.mx](mailto:esucar@inaoep.mx)

[ccc.inaoep.mx/esucar](http://ccc.inaoep.mx/esucar)

Tecnologías de Información

UPAEP

# Reglas de Producción

- Introducción
- Partes principales
- Tipos de razonamiento
- Explicación
- Resolución de conflictos
- Apareamiento de patrones

# Introducción

Normalmente se asocia la inteligencia con “regularidades” y el comportamiento inteligente parece que ejecuta reglas.

Newell y Simon 70's proponen los sistemas de producción como un modelo psicológico del comportamiento humano.

## **Introducción**

**En este modelo, parte del conocimiento humano se representa en forma de producciones o reglas de producción.**

**Se asemeja al proceso de memoria humano: memoria a corto plazo (deducciones intermedias) y memoria a largo plazo (producciones).**

# Introducción

Normalmente las reglas de producción se ven como un formalismo en el cual representar el conocimiento y es el formalismo más usado en los sistemas expertos.

Originalmente las producciones eran reglas gramaticales para manipular cadenas de símbolos.

# Reglas de producción

Manipulan estructuras de símbolos, como listas o vectores.

Se tiene:

- Un conjunto **N** de nombres de objetos en el dominio
- Un conjunto **P** de propiedades que representan atributos de los objetos

- Un conjunto **V** de valores que los atributos pueden tener

Generalmente se usa una tripleta: (**objeto atributo valor**).

Por ejemplo:

- (juan edad 25)
- (juan edad X)
- (turbina1 capacidad 50)

## Estructura de una regla:

$$P_1 \dots, P_m \longrightarrow Q_1 \dots, Q_n$$

Que significa:

IF las condiciones  $P_1$  y  $P_2$  y... y  $P_m$  se cumplen THEN realiza las acciones (o concluye)  $Q_1$  y... y  $Q_n$

**IF Animal es\_un carnívoro AND**

**Animal color café AND**

**Animal tiene rayas**

**THEN Animal es tigre**

## Propiedades de las reglas:

- **Modular:** cada regla define un pequeño y relativamente independiente pedazo de conocimiento
- **Incremental:** nuevas reglas pueden ser añadidas a la base de conocimiento relativamente independiente de las demás

- **Modificable:** como consecuencia de ser modular, las reglas pueden ser modificadas o se pueden añadir/quitar reglas fácilmente
- **Transparencia:** habilidad de explicar sus decisiones y soluciones

# Partes principales

## Un sistema de producción tiene:

- Un conjunto de reglas (base de conocimiento)
- Un interprete de reglas o máquina de inferencia (que decide qué regla aplicar, controla la actividad del sistema)
- Una memoria de trabajo (que guarda los datos, metas, y resultados intermedios)

# Memoria de Trabajo

Guarda inferencias/aseveraciones temporalmente.

Es la estructura de datos que es observada y manipulada/cambiada por las reglas.

## **Memoria de Trabajo**

**Los datos de la memoria de trabajo son los que permiten cumplir las condiciones de las reglas y dispararlas (las reglas verifican la existencia de elementos en la memoria de trabajo para disparar).**

**Las acciones de las reglas: modifican, añaden o quitan elementos de la memoria de trabajo (o producen efectos secundarios).**

# Máquina de Inferencia

El interprete o máquina de inferencia realiza el ciclo *reconoce-actua*:

1. Aparea las condiciones (o acciones) de las reglas con los elementos de la memoria de trabajo

**2.**

Si existe más de una regla que puede disparar, escoge una (resolución de conflictos)

**3.**

Aplica la regla (ejecuta las acciones/conclusiones) que puede involucrar cambios la memoria de trabajo

# **Máquina de Inferencia**

**Generalmente el ciclo empieza con unos datos iniciales y para cuando no hay reglas aplicables (o por una regla).**

**El sentido/importancia/contribución de cada regla depende de su contribución dentro de todas las reglas para solucionar el problema.**

# Ejemplo: sistema para créditos

## Base de Conocimiento

- Si hist-credito & solvente & refer  
Entonces otorga-prest
- Si tarjetas-credito > 2 Entonces  
hist-credito
- Si hipoteca-casa OR prest-carro  
Entonces hist-credito
- Si ingresos > pago-mensual x 3  
Entonces solvente
- Si ref-1 & ref-2 & ref-3  
Entonces refer

## Memoria de Trabajo

- ingresos=1000
- pago-mensual=200
- hipoteca-casa
- ref-Maria
- ref-Juan
- ref-Pedro

**hist-credito**

**solvente**

**refer**

**otorga-prest**

# Máquina de Inferencia

Existen diferentes estrategias de razonamiento: a nivel global las reglas pueden ser llevadas en un ***encadenamiento hacia adelante*** o en un ***encadenamiento hacia atrás***.

# Estrategias de Razonamiento

**Encadenamiento hacia adelante  
(forward chaining/ data driven/  
event driven/ bottom-up) parte de  
hechos para cumplir condiciones y  
ejecutar acciones (creando nuevos  
hechos).**

# Estrategias de Razonamiento

**Encadenamiento hacia atrás  
(backward chaining/ goal driven/  
expectation driven/ top-down) parte  
de los estados meta y trata de  
cumplir las condiciones necesarias  
para llegar a ellos.**

# Estrategias de Razonamiento

Se pueden tener estrategias que hacen combinación de ambos: oportunístico.

v.g.,

**P1:**    \$ -> a\$a

**P2:**    \$ -> b\$b

**P3:**    \$ -> c\$c

## Estrategias de Razonamiento

Podemos dado **c** usar las reglas P1, P1, P3, P2, P3 y llegar a: **cbcaacaacbc**.

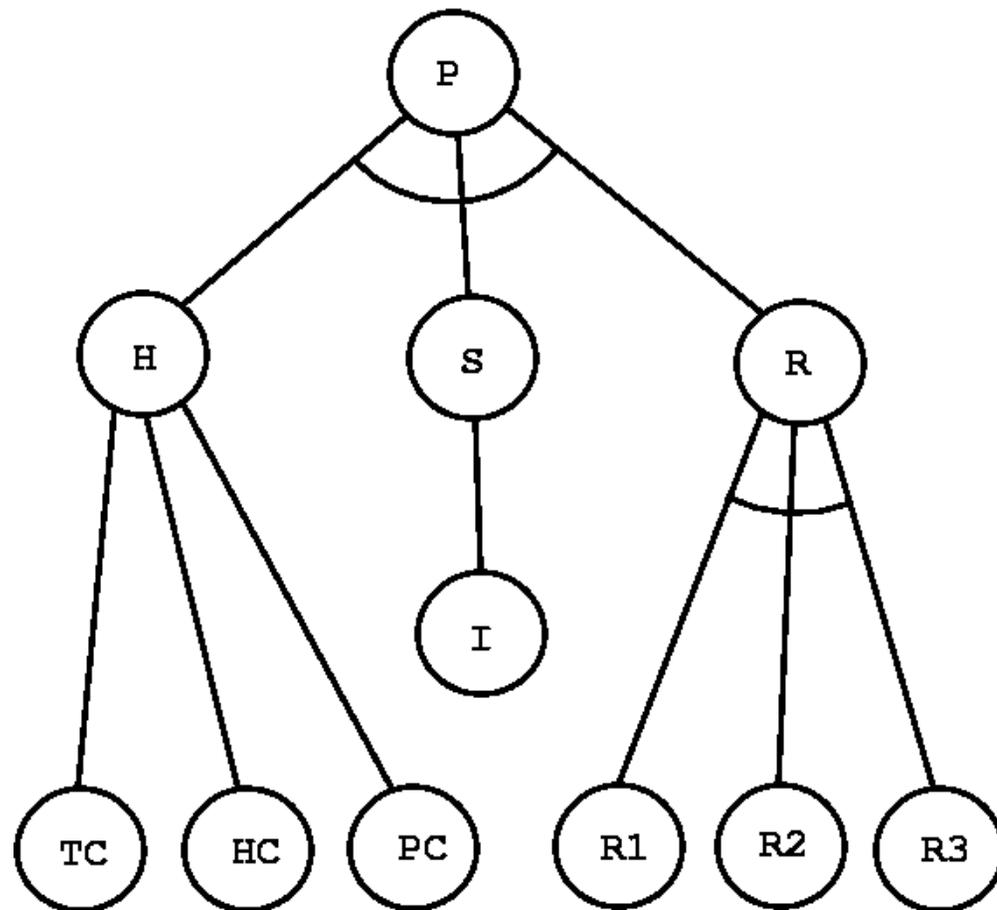
Otra forma es tomar **cbcaacaacbc** y ver que reglas se aplican hasta llegar a algo conocido (**c**).

## Árbol AND/OR

El proceso de inferencia se puede ver como un espacio de búsqueda AND/OR, con nodos AND siendo todas las condiciones/ acciones que se tienen que cumplir y los nodos OR siendo las posibles reglas a disparar/ considerar.

# Ejemplo:

Árbol AND/OR



## ¿Cuándo usamos cada uno?

Depende del propósito y la forma del espacio de búsqueda.

Si el propósito es de cubrir todo lo que se pueda deducir de un conjunto de hechos, el árbol se “achica”, tenemos claras las entradas pero no las conclusiones, entonces encadenamiento hacia adelante.

## ¿Cuándo usamos cada uno?

Si el propósito es verificar/negar una conclusión, el árbol se “ensancha”, tenemos claras las metas pero no las entradas, entonces encadenamiento hacia atrás.

# Ejemplo de encadenamientos

Forward:

- $HC \rightarrow H$
- $I \rightarrow S$
- $R1, R2, R3 \rightarrow R$
- $H, S, R \rightarrow P$

Backward:

- $P \leftarrow H, S, R$
- $H \leftarrow HC$
- $S \leftarrow I$
- $R \leftarrow R1, R2, R3$

# Explicación

Explicaciones sirven para:

- **Mostrar deficiencias de las reglas**
- **Clarificar suposiciones de la máquina**
- **Explicar situaciones no anticipadas**
- **Hacer sentir al usuario más seguro**
- **Hacer ver la máquina más “inteligente”**

# Explicación

Con las reglas podemos resolver preguntas:

- **HOW**, e.g., ¿Cómo supiste N?
- **WHY**, e.g., ¿Por qué quieres saber A?

# Ejemplo de explicaciones

## Reglas:

R1: If una persona tiene \$30,000 y tiene grado de licenciatura  
Then debe de invertir a plazo fijo

R2: If una persona gana mas de \$120,000 al año y tiene licenciatura  
Then debe de invertir en acciones

R3: If una persona es menor de 30 y esta invirtiendo a plazo fijo  
Then debe invertir en acciones

R4: If una persona es menor de 30  
Then tiene nivel licenciatura

R5: If una persona quiere invertir en acciones  
Then debe de invertir en Telmex

# Ejemplo de explicaciones

- **Hechos:**  
Tiene \$30,000 y tiene 25 años  
Quiere saber si debe de invertir en acciones
- **Sistema:** ¿Cuál es tu ingreso anual?
- **Usuario :** ¿Porqué?
- **Sistema:** Quiero saber si es mayor a \$120,000, porque como sé que tienes nivel de licenciatura, si ganas más de \$120,000, te recomiendo invertir en acciones.

# Porqué (WHY)

- La explicación se basa en utilizar inferencia hacia adelante, en el ejemplo:
- R4: If una persona es menor de 30  
Then tiene nivel licenciatura
- R2: If una persona gana mas de \$120,000 al año y tiene licenciatura  
Then debe de invertir en acciones
- De aquí el sistema deduce que necesita saber el ingreso para decidir si invierte en acciones

# Ejemplo de explicaciones

- **Hechos:**  
Tiene \$30,000 y tiene 25 años  
Quiere saber si debe de invertir en Telmex
- **Sistema:** Invierte en Telmex
- **Usuario :** ¿Cómo?
- **Sistema:** Como tienes \$30,000 y eres menor de 30, (R4) tienes nivel licenciatura, (R1) debes invertir a plazo fijo. Por otro lado si inviertes a plazo fijo (R3) debes invertir en acciones. Si quieres invertir en acciones te recomiendo (R5) que inviertas en Telmex.

# Cómo (HOW)

- La explicación se basa en hacer inferencia hacia atrás, determinando las reglas que se aplicaron para llegar a la conclusión, hasta llegar a los datos de entrada.

- **En el ejemplo:**

R5: If una persona quiere invertir en acciones  
Then debe de invertir en Telmex

R3: If una persona es menor de 30 y esta invirtiendo a plazo fijo  
Then debe invertir en acciones

R1: If una persona tiene \$30,000 y tiene grado de licenciatura  
Then debe de invertir a plazo fijo

R4: If una persona es menor de 30  
Then tiene nivel licenciatura

Datos: menor de 30, tiene \$30,000

## **Resolución de Conflictos**

**En razonamiento hacia adelante pueden existir más de una regla que puede disparar.**

**En razonamiento hacia atrás pueden existir más de una regla que cumple con las metas.**

**Lo que se necesita es una estrategia de resolución de conflictos que guíe (decida cual de las posibles reglas disparar) y evite que el proceso sea exponencial.**

## Tipos de Control

Para controlar ésto se utilizan: **control global** (independiente del dominio) y **control local** (dependiente del dominio).

El control local puede ser por medio de meta-reglas (reglas que controlan reglas).

Puntos: **sensibilidad** (responder rápidamente a cambios en el medio) y **estabilidad** (mostrar cierta continuidad en la línea de razonamiento).

## Estrategias globales

Las estrategias globales más utilizadas son:

- No se permite disparar una regla más de una vez con los mismos datos (*refractorines*)
- Preferir reglas que utilizan datos más recientes. La idea es de seguir una línea de razonamiento (*recency*)

## Estrategias globales

- Preferir reglas que son más específicas, i.e., reglas que tienen una mayor cantidad de condiciones y por lo tanto son más difíciles de disparar (*specificity*)

## **Meta-reglas**

**Algunos sistemas permiten tener reglas que razonan en cuanto a qué reglas disparar (meta-reglas).**

**Las meta-reglas tiene el rol de dirigir el razonamiento (más que realizar el razonamiento).**

# Ejemplos de meta-reglas

## Para una aplicación:

IF Edad Paciente > 65

THEN Aplica reglas de enfermedades de la vejez

## Para una clase de aplicaciones:

IF “Edad” en cierto intervalo

THEN Aplica reglas para dicho intervalo

# Apareamiento de Patrones

Normalmente existen variables en las reglas y se tiene que hacer un “apareamiento de patrones”.

**Notación : ?x aparea un elemento, !x  
aparea varios elementos, v.g.,**

**If Persona nombre ?x edad ?y**

**And ?y > 12**

**And ?y < 20**

**Then ?x es\_un adolescente**

**Hecho: Persona nombre Juan edad 25.**

# Apareamiento de Patrones

El apareo de condiciones en general es más fácil en encadenamiento hacia adelante que en encadenamiento hacia atrás:

En Fwd: Datos hechos (sin variables) apareamos con condiciones (con variables) y producimos nuevos hechos (sin variables).

# **Apareamiento de Patrones**

**En Bwd: Hipótesis (con variables) apareamos consecuencias (con variables) y producimos nuevas hipótesis (con variables). Se acerca más a unificación.**

## **Encadenamiento hacia atrás**

**Pasos de encadenamiento hacia atrás: el proceso trata de establecer valores para las variables de las reglas**

# Encadenamiento hacia adelante

Pasos de encadenamiento hacia adelante:

- **Infiere**
  - **Selecciona reglas**
    - **Resuelve conflictos**
    - **Aplica regla**

## **Ventajas de Reglas:**

- **Permiten representar el conocimiento en forma adecuada para las computadoras**
- **Modularizan pedazos de conocimiento**
- **Permiten el desarrollo incremental**
- **Las decisiones son entendibles y explicables**
- **Abren nuevas posibilidades computacionales (paralelismo)**
- **Representación homogénea de conocimiento**
- **Permiten interacciones no planeadas y útiles**

## **Desventajas de Reglas:**

- **No hay fundamento para decidir qué problemas tiene solución**
- **Problemas de verificación / consistencia / completez de conocimiento**
- **Escalamiento sin perder entendimiento /eficiencia**
- **Permiten interacciones no planeadas y no deseadas**
- **No tienen acceso al razonamiento que hay detrás de las reglas**

- **Bases de reglas grandes son difíciles de mantener y desarrollar (requiere una partición de las reglas, pero el formalismo no lo permite hacer directamente)**
- **No son adecuadas cuando existe incertidumbre ya que se pierde la modularidad**

## **RETE (red)**

**Del tiempo de ejecución el 90% se consume en el proceso de apareo.**

**El algoritmo Rete se basa en dos observaciones (suposiciones):**

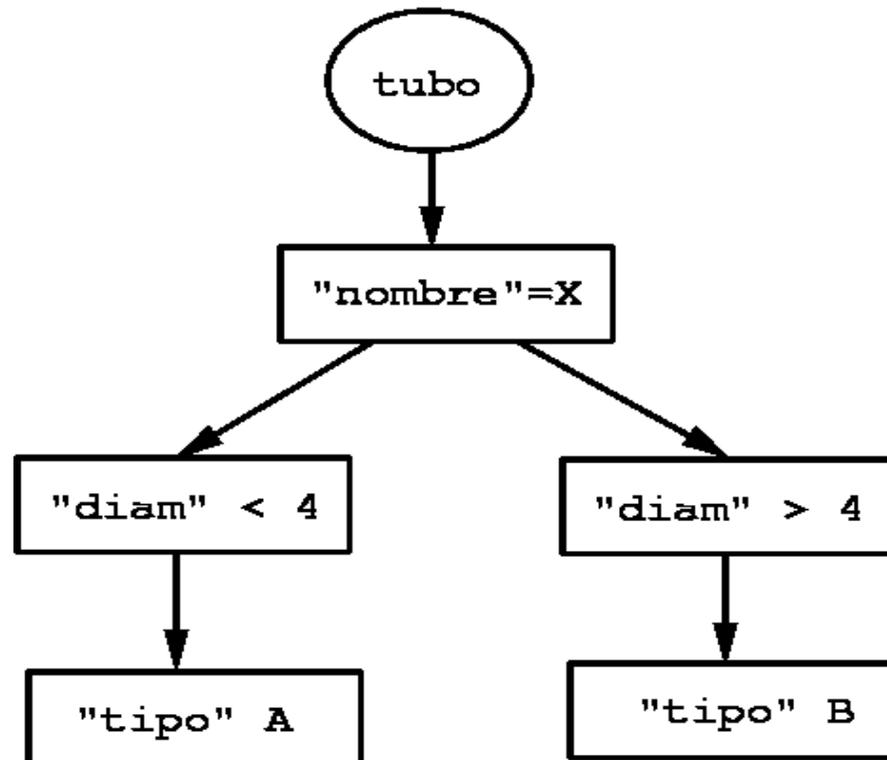
- 1. La memoria de trabajo es muy grande y cambia poco entre cada ciclo.  
Esto implica que mucha información sigue presente en el siguiente ciclo y se puede utilizar**
- 2. Las condiciones de muchas reglas son similares**

# RETE - ejemplo

```
IF tubo "nombre"=X, "diam" < 4  
THEN tubo "tipo" A
```

```
IF tubo "nombre"=X, "diam" > 4  
THEN tubo "tipo" B
```

## Grafo "RETE"



# Herramientas de Desarrollo

Existen herramientas de desarrollo para construir sistemas basados en reglas. Éstas las podemos clasificar en 3 niveles:

1. *Shells*: Exsys, Nexpert, ...
2. Lenguajes de reglas: OPS, Clips, ...
3. Lenguajes simbólicos: Lisp, Prolog

# Tarea

- Leer Capítulos 9 (9.3) y 12 (12.2) de Russell
- Práctica 1 (en la página del curso)