

# Lectura

## Inteligencia Artificial

Texto basado en las notas de Bareen Haval Sadiq Mzori y su trabajo Forward and Backward Chaining Techniques of Reasoning in Rule-Based Systems.

"El presente texto ha sido preparado de manera exclusiva para los alumnos del Curso de Inteligencia Artificial, que forma parte de la Plan de Estudio de la Escuela de Ciencia de Computación, según el artículo 44 de la Ley sobre el Derecho de Autor, D.L. N°822. Queda prohibida su difusión y reproducción por cualquier medio o procedimiento, total o parcialmente fuera del marco del presente curso".

---

## 1 Forward y Backward Chaining

Un sistema experto es un sistema informático que permite imitar la capacidad de toma de decisiones del ser humano. Están diseñados para resolver problemas intrincados mediante el razonamiento sobre el conocimiento. En general, hay tres componentes principales en un sistema experto basado en reglas: interfaz de usuario, base de conocimientos y motor de inferencia. Los sistemas expertos fueron las primeras formas activas de software de IA.

La interfaz de usuario desempeña el papel de puente entre el usuario y el sistema experto. Es responsabilidad de una persona crear la interfaz de usuario para un problema en particular. La interfaz de usuario permite al usuario enviar una consulta al sistema experto para recibir un consejo. Por tanto, la consulta y el consejo se consideran entrada y salida del sistema experto, respectivamente.

La base de conocimientos es una parte muy importante del sistema experto. Es simplemente una tecnología que almacena las reglas y los hechos, básicamente en forma de declaraciones IF ... THEN. Estas declaraciones son proporcionadas por expertos humanos. La base de conocimientos demuestra los hechos que el sistema necesita para resolver el problema y alcanzar la meta.

El motor de inferencia se encarga de proporcionar las reglas (lógicas) a la base de conocimiento y luego generar nuevos hechos usando estas reglas. Una de las técnicas utilizadas para alcanzar este objetivo son las reglas de producción y sus tipos: el encadenamiento hacia adelante (forward chaining) y el encadenamiento hacia atrás (backward chaining). Un punto importante es que el motor de inferencia necesita una memoria sólida para guardar todos los datos iniciales y nuevos proporcionados por el motor de interfaz.

Los sistemas basados en reglas se utilizan eficazmente en finanzas, derecho, medicina, negocios, manufactura, educación, informática y juegos de computadora.

Las reglas de producción son un conjunto de declaraciones representadas en forma de declaraciones IF ... THEN. Por ejemplo, IF A1 y B1 son verdaderos THEN C1 es verdadero, y se debe tomar la acción E1. Las condiciones A1 y B1 se denominan el lado izquierdo de la regla (LHS) y C1 y E1 se denominan el lado derecho de la regla (RHS). Entonces, si los elementos LHS corresponden a los hechos en la memoria de trabajo, entonces se activa la regla y se ejecuta el RHS. Otro caso es que la regla se puede negar, lo que significa que el LHS solo se ejecutará cuando no coincida ningún hecho en la memoria de trabajo. Las acciones pueden actualizar la memoria de trabajo agregando, modificando o eliminando los hechos.

El proceso de salida de una regla que activa otra regla se llama encadenamiento. La técnica de encadenamiento consiste en dividir la tarea en pequeños procedimientos y luego informar cada procedimiento dentro de la secuencia por sí mismo.

El encadenamiento hacia adelante es un razonamiento basado en datos, y comienza con los hechos conocidos e intenta hacer coincidir las reglas con estos hechos. Existe la posibilidad de que todas las reglas coincidan con la información (condiciones). En el encadenamiento hacia adelante, primero se prueban las reglas que buscan hechos coincidentes y luego se ejecuta la acción. En la siguiente etapa, la memoria de trabajo se actualiza con nuevos hechos y el proceso de comparación comienza de nuevo. Este proceso se está ejecutando hasta que no queden más reglas o se alcance el objetivo.

El encadenamiento hacia adelante es útil cuando hay mucha información disponible. Es útil implementar el encadenamiento directo si hay un número infinito de posibles soluciones, como problemas de configuración y planificación.

Lo contrario de un encadenamiento hacia adelante es un encadenamiento hacia atrás, es decir, a diferencia del encadenamiento hacia adelante del razonamiento basado en datos, un encadenamiento hacia atrás es un método de razonamiento impulsado por objetivos. El encadenamiento hacia atrás comienza desde el objetivo (desde el final) que es una solución hipotética y el motor de inferencia intenta encontrar la evidencia coincidente. Cuando se encuentra, la condición se convierte en el subobjetivo y luego se buscan reglas para probar estos subobjetivos. Simplemente coincide con el RHS del objetivo. Este proceso continúa hasta que se prueban todos los subobjetivos y retrocede al paso anterior donde se eligió una regla. Si no hay una regla que se establezca en un subobjetivo individual, se elige otra regla.

El razonamiento de encadenamiento hacia atrás es bueno para los casos en los que no hay tantos hechos y la información (hechos) debe ser generada por el usuario. El razonamiento del encadenamiento hacia atrás también es eficaz para su aplicación en las tareas de diagnóstico.

## 1.1 Sistemas basados en reglas

El término conocimiento significa la comprensión de un dominio específico. La persona que tiene algún tipo de conocimiento y se lo pasa a otra persona se llama experto. El experto debe tener un amplio conocimiento de los hechos y las reglas y utilizar la experiencia en un dominio en particular y esta persona se denomina experto en el dominio. Cualquier sistema basado en reglas se puede crear en presencia de algunos componentes. En primer lugar, debe estar disponible un conjunto de hechos que se utilizan en la memoria de trabajo de un sistema. En segundo lugar, debe existir un conjunto de reglas que abarquen las posibles acciones a tomar. Finalmente, se debe usar la condición que determina si se encuentra una solución o si no hay solución en absoluto. Es muy difícil representar el proceso de pensamiento humano en forma de algoritmo, porque este proceso es un proceso interno en el cerebro de un humano. El experto puede expresar sus conocimientos como reglas para describir y luego resolver problemas. En inteligencia artificial, las reglas se utilizan para representar el conocimiento, y el conocimiento se puede representar mejor en forma de reglas lingüísticas IF-THEN. En la parte IF, la regla lógica es fácil de crear y comprender.

```
IF está lloviendo  
THEN lleve paraguas
```

Las reglas se componen de dos partes. La parte IF (lado izquierdo o LHS) se llama condición (antecedente) y la parte THEN (lado derecho o RHS) se llama acción (consecuente). Si la parte condicional de la regla (antecedente) es verdadera, entonces se afirma la parte de conclusión (consecuente). Esto puede ser un objetivo o un hecho nuevo que se agregará a la memoria de trabajo. La regla puede tener más de una condición conectada con (conjunción) AND, (disyunción) OR o una mezcla de ambos.

La parte de condición de una regla consiste en el objeto y su valor. El objeto se representa en forma lingüística y se vincula a su valor mediante un operador.

## 1.2 Forward Chaining

En un encadenamiento hacia adelante, el propósito del sistema es determinar los resultados de las premisas.

Este método comienza con el conjunto de hechos (los datos disponibles) y luego verifica si se cumplen las reglas dadas. Si es así (hay una coincidencia), la regla se ejecuta (se dispara). Este proceso continúa hasta que se encuentra la meta o objetivo (suponiendo que solo una respuesta es requerida), o no hay nuevos hechos que agregar.

Un aviso importante es que el hecho no es "nuevo" si se trata de un cambio de nombre de un hecho conocido. El cambio de nombre se refiere a que dos oraciones se renombran entre sí si son iguales, excepto que los nombres de las variables son diferentes.

Las reglas se expresan en una forma cercana al lenguaje humano, pero en realidad el sistema basado en reglas debe representarse de una manera que los procesos de máquina limitados puedan entender. No existe una regla de sintaxis estándar. El LHS de la regla consta del nombre del objeto seguido de los atributos y valores del objeto.

El algoritmo de encadenamiento directo se representa como la secuencia de los siguientes pasos:

- Los hechos iniciales son entradas del usuario que se configuran en la base de datos (memoria de trabajo)
- Se comprueba LHS de las reglas de producción
- Si la parte de condición lógica de una regla (parte IF) coincide, entonces la regla se activa
- Se ejecutan acciones del RHS
- Se retiran viejas condiciones/hechos
- Se ingresan nuevas condiciones/hechos
- Se realizan otras acciones de entrada-salida, unificaciones, etc.
- Se repite hasta que no se activen otras reglas.

### 1.3 Coincidencia en encadenamiento hacia adelante

La coincidencia de patrones en el proceso de encadenamiento también se conoce como unificación. Entonces, la aplicación del procedimiento de encadenamiento depende de la unificación. En la unificación se considera un conjunto de vinculaciones de variables. En un proceso de unificación se comparan dos oraciones atómicas y se devuelve un unificador si existe una de las oraciones dadas.

Los literales se muestran como una lista en la que el nombre del predicado es el primer elemento y el resto son argumentos. Los argumentos pueden ser un solo elemento o una lista de elementos. Para unificar dos literales, se debe examinar si sus primeros elementos son iguales. Si es así, continúe. De lo contrario, no se unificarán.

Un ejemplo simple del encadenamiento hacia adelante se da a continuación:

```
R1: IF A=5 AND B=7
    THEN C=10
R2: IF C=10 AND D=12
    THEN E=15
R3: IF E=15
    THEN F=18
```

Están presentes los siguientes hechos:  $A = 5$ ,  $B = 7$  y  $D = 12$ . Los valores de A y B ayudan a inferir el valor de C a partir de la regla 1, luego la regla 2 se dispara debido a la C inferida y al hecho conocido D (por lo que se infiere) y finalmente la regla 3 se dispara para inferir F. utilizando el encadenamiento hacia adelante se alcanza la meta F.

La solución del problema es hacer coincidir las reglas con los hechos una a una de manera exhaustiva, además de utilizar técnicas de indexado. El algoritmo que representa la solución es el algoritmo RETE.

El algoritmo RETE es muy útil para comparar hechos con los patrones de una regla. Escribir las reglas de una manera particular es más poderoso que escribirlas al azar y esto se realizará implementando el algoritmo RETE. A continuación se proporcionan un ejemplo de encadenamiento hacia adelante. El mecanismo de inferencia proporcionará una conclusión adecuada de la base de conocimientos.

R1: IF tienes hambre THEN haz comida  
R2: IF preparas comida THEN necesitas ingredientes  
R3: IF necesita ingredientes THEN, vaya al refrigerador  
R4: IF el refrigerador está lleno THEN, tome los ingredientes  
R5: IF tomate Y lechuga THEN haz ensalada  
R6: IF la carne THEN has parrillada  
R7: SI pollo THEN haz pollo frito  
R8: SI el refrigerador está vacío THEN duerma

Encuentra la conclusión, si los siguientes hechos están presentes: hambre, el refrigerador está lleno, carne.

En primer lugar, la regla R1 se dispara, luego las reglas R2, R3, R4 se disparan continuamente una tras otra y, finalmente, la regla R6 se dispara. Entonces la conclusión es hacer parrillada.

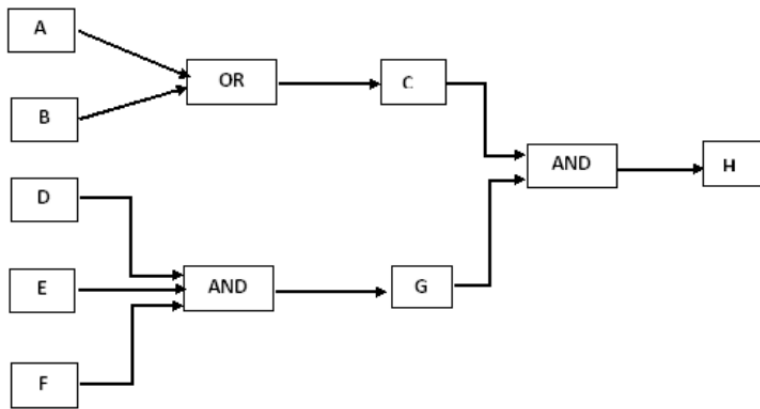
A continuación, se proporciona otra base de conocimientos que sirve para probar la conclusión.

R1: IF A OR B  
THEN C  
R2: IF D AND E AND F  
THEN G  
R3: IF C AND G  
THEN H

Se presentan los siguientes hechos: B, D, E, F.

Objetivo: demostrar H.

La estructura de un ejemplo de encadenamiento hacia adelante se muestra en la figura:

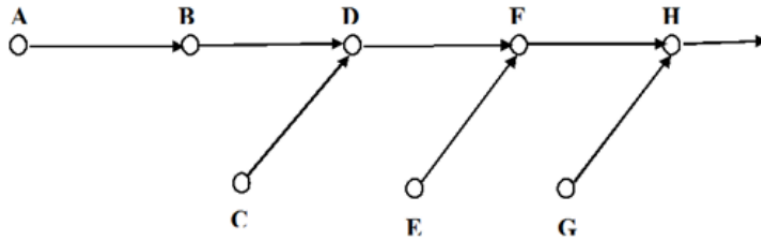


De la primera regla C es inferida, la segunda regla G se infiere, y finalmente, de la tercera regla H se infiere.

El encadenamiento hacia adelante también se puede representar en forma de gráfico con nodos y enlaces. Supongamos que están disponibles las siguientes reglas:

R1: IF A THEN B  
R2: IF B AND C THEN D  
R3: IF D AND E THEN F  
R4: IF F AND G THEN H

Para las reglas anteriores, el encadenamiento hacia adelante se ilustra en forma de grafo que comienza desde el nodo de la premisa hasta un nodo de salida del objetivo.



## 1.4 Resolución de conflictos en el encadenamiento hacia adelante

Si existe la posibilidad de activar dos o más reglas en las que las partes IF están satisfechas para ejecutarse al mismo tiempo (las reglas tienen la misma condición), este proceso se denomina conjunto de conflictos. En este caso, el motor de inferencia debe elegir la regla a disparar del conjunto. La elección de la regla que se debe activar cuando más de una regla puede activarse es una función de una estrategia de resolución de conflictos.

A continuación se ofrece un ejemplo sencillo con una base de conocimientos que provoca un conjunto de conflictos:

```
R1: IF hace calor
    THEN usa polo
R2: si hace calor
    THEN ve a nadar
R3: IF hace calor
    THEN poner crema solar
```

Si está presente el hecho de que hace calor, entonces se activarán las tres reglas, por lo que debemos averiguar qué acción se debe tomar.

Este problema se puede resolver utilizando las siguientes estrategias:

1) La regla con la prioridad más alta se activa primero si hay pocas reglas en la base de conocimientos. Las reglas deben organizarse en un orden adecuado en la base de conocimientos. A cada regla se le da prioridad y se selecciona la regla con la mayor prioridad.

2) Si una regla específica proporciona más información que una general, esta regla específica se activa y este método se denomina la estrategia de coincidencia más larga. Este método es más útil en casos no estándar.

3) Se activa la regla que se agregó más recientemente a la base de datos. Los nuevos elementos de la memoria de trabajo deben dispararse antes que los más antiguos. De esta manera, el sistema seguirá una sola cadena de razonamiento, en lugar de seguir esbozando nuevas conclusiones a partir de los datos antiguos.

La resolución de conflictos se puede definir como un conjunto de pares de los siguientes componentes:

<regla IF ... THEN , coincidencia de elemento de una memoria de trabajo>

Cuando se decide qué regla se activa primero, la resolución del conflicto asegura que la misma regla no se pueda ejecutar dos veces.

Esta estrategia ayuda a los sistemas de encadenamiento hacia adelante a obtener un rendimiento razonable, pero la construcción de las reglas es el punto más importante porque si no se escriben de manera adecuada, tendremos un control limitado sobre lo que sucederá con las condiciones previas definidas con precisión, así como cuando deben dispararse diferentes reglas.

Para controlar el rendimiento del sistema, se utilizan los elementos especiales de la memoria de trabajo. Por ejemplo, podríamos decidir que hay algunas etapas básicas de procesamiento al realizar algunas tareas y ciertas reglas deberían activarse en una etapa determinada, podríamos tener un elemento especial de

memoria de trabajo y agregar a las condiciones previas de todas las reglas relevantes. Cuando se completa esta etapa, se elimina el elemento de memoria de trabajo.

Hay dos tipos de resolución de conflictos: resolución de conflictos generales y resolución de conflictos específicos de problemas.

El proceso de selección de reglas es muy importante, porque una selección inconsciente de reglas hace que se activen varias reglas y muchas de estas reglas pueden no ser útiles para alcanzar el objetivo. Como resultado, al usuario se le hacen muchas preguntas innecesarias. Por lo tanto, se degradará el rendimiento del sistema.

En los sistemas basados en reglas, el conocimiento se considera mediante meta-reglas. Las meta-reglas se utilizan para hacer que el proceso de selección de reglas sea más eficiente. Las meta-reglas se utilizan para determinar la estrategia de resolución de conflictos. Representan el conocimiento sobre cómo funcionará el sistema.

Las meta-reglas pueden definir la siguiente propiedad: si el conocimiento de un experto es más confiable y probable en comparación con las reglas de otro experto, esto se denomina meta conocimiento. Un metaconocimiento también se puede interpretar como un conocimiento sobre el conocimiento.

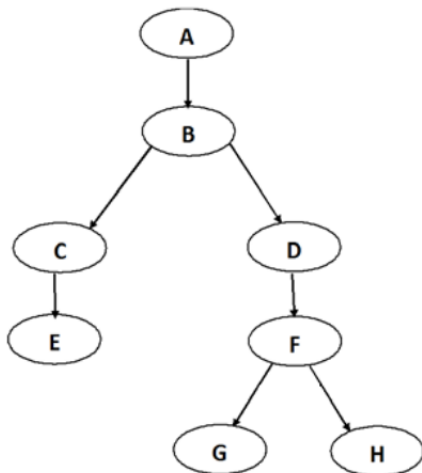
Las meta-reglas tienen mayor prioridad. Después de hacer coincidir las reglas y los hechos y usar la resolución de conflictos, el sistema elegirá una regla para encontrar el objetivo y en la última etapa, esta regla se activará. Este proceso puede causar una modificación y se le harán preguntas al usuario además de los nuevos cálculos y nuevas acciones. Después de todo, el objetivo se le presentará al usuario.

## 1.5 Búsqueda en profundidad en el encadenamiento hacia adelante

A veces, las reglas del encadenamiento hacia adelante se pueden representar en forma de árbol desde la raíz hasta la profundidad. A continuación se dan las siguientes reglas:

R1: IF A THEN B  
R2: IF B THEN C AND D  
R3: IF C THEN E  
R4: IF D THEN F  
R5: IF F THEN G AND H

La búsqueda en profundidad en un encadenamiento hacia adelante para las reglas anteriores se representa en la figura:



El encadenamiento hacia adelante generalmente tiene reglas que son inútiles en el sistema cuando intenta encontrar el objetivo. Consideremos el siguiente conjunto de reglas:

IF A AND B THEN D  
IF A AND C THEN E

```
IF A AND F THEN G
IF D AND E THEN H
IF D THEN J
IF E AND F AND H THEN K
```

Los siguientes hechos (entradas) están presentes: A, B y C. El objetivo del razonamiento es J.

A continuación se dan dos soluciones del proceso de encadenamiento hacia adelante: el razonamiento subóptimo y el razonamiento óptimo.

#### **Razonamiento subóptimo:**

```
IF A AND B THEN D
IF A AND C THEN E
IF D AND E THEN I
IF D THEN J
```

#### **Razonamiento óptimo:**

```
IF A AND B THEN D
IF D THEN J
```

Como puede verse, la solución subóptima no es eficiente y empeorará si aumenta el número de reglas. Las heurísticas se utilizan para resolver este problema. En general, las heurísticas se clasifican en formas débiles y fuertes. Las heurísticas débiles prefieren conjuntos de reglas simples que tienen un pequeño número de condiciones (parte IF) y también analizan la diferencia sintáctica entre el estado actual y el estado objetivo, lo que se denomina análisis means-end. Las heurísticas fuertes se conocen como meta-reglas, destinadas a ser utilizadas para un dominio específico del conocimiento y normalmente son más útiles que las heurísticas débiles.

Las mejores meta-reglas son las heurísticas fuertes automáticas. Aprenden dinámicamente a seleccionar la regla en cada etapa mientras el sistema está en el modo de ejecución.

## **1.6 Ventajas y desventajas del encadenamiento hacia adelante**

Hay algunas ventajas del encadenamiento hacia adelante que se detallan a continuación:

- Funciona muy bien cuando un problema comienza naturalmente mediante la recopilación de datos y la búsqueda de información que se pueda recopilar para usar en pasos futuros.
- Tiene la capacidad de proporcionar una gran cantidad de datos a partir de los pocos datos o hechos iniciales disponibles.
- Es una técnica muy popular para la implementación en sistemas expertos y sistemas que utilizan reglas de producción en la base de conocimientos. Para el sistema experto que necesita interrupción, control, monitoreo y planificación, el encadenamiento hacia adelante es la mejor opción.
- Cuando hay pocos hechos y estados iniciales, es muy útil aplicar el encadenamiento hacia adelante.

También hay algunas desventajas de un encadenamiento hacia adelante y se detallan a continuación:

- El motor de inferencia generará nueva información sin saber qué información se utilizará para alcanzar el objetivo.
- Se le puede pedir al usuario que ingrese una gran cantidad de entradas sin saber qué entrada es relevante para la conclusión.
- Se pueden disparar varias reglas que no tienen nada para alcanzar el objetivo.
- Puede producir diferentes conclusiones que son las causas de un alto costo del proceso de encadenamiento.

## 1.7 Principio básico del encadenamiento hacia atrás

El encadenamiento hacia atrás funciona en sentido inverso al encadenamiento hacia adelante y comienza desde el objetivo e intenta encontrar datos para demostrar su objetivo. Por lo tanto, también se le llama razonamiento impulsado por objetivos. Después de comenzar desde el objetivo dado, se realiza la búsqueda de partes THEN de las reglas dadas (parte de acción) (RHS) y si se encuentra la regla y su parte IF (condición) coincide con los datos en la base de datos, entonces la regla es ejecutada (lanzada). De lo contrario, si la condición no coincide con los datos (hechos) en la base de datos, el motor de inferencia establece la regla que está trabajando en una pila y crea un nuevo subobjetivo para probar la condición en la regla actual. La base de conocimientos sigue buscando reglas para demostrar el subobjetivo. El proceso de apilar las reglas se repite hasta que la base de conocimientos no tiene reglas para probar el subobjetivo.

Considera las siguientes reglas:

R1: IF A AND B THEN C

R2: IF C THEN E

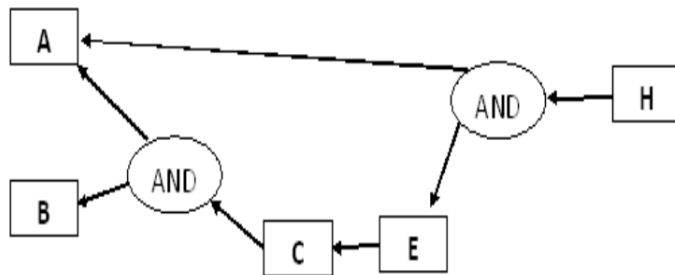
R3: IF A AND E THEN H

Hechos: A, B

Objetivo: Demuestra H

Al principio, el sistema busca una regla que compruebe el objetivo, es decir, busca el RHS de las reglas, que es la regla 3. Luego se considera el LHS de la regla que es la parte IF que contiene la condición de la regla. En esta etapa, el sistema intenta satisfacer la condición y si los hechos coinciden con la condición de esta regla, la regla se activa. De lo contrario, el sistema establece un subobjetivo que contiene la nueva regla y trata de demostrarlo. Entonces, en la regla 3 el sistema mira el LHS y encuentra A y E en la base de datos, pero solo tenemos A. Por lo tanto, el sistema ahora hará que probar E sea su nuevo subobjetivo. E se demuestra en la regla número 2, el LHS de la regla número 2 es C. El nuevo subobjetivo es demostrar C. El sistema agrega el antiguo subobjetivo en una pila. C se prueba en la regla número 1. La condición en la regla número 1 puede coincidir con los hechos en la base de datos. Ahora el sistema puede probar los subobjetivos en la pila y el objetivo está probado.

La estructura de un encadenamiento hacia atrás para el problema anterior se representa en la figura siguiente:



El concepto de encadenamiento hacia atrás se utiliza para construir muchos tipos de sistemas expertos y especialmente los sistemas interactivos simulan la conversión entre usuarios y una persona experta. Usando la técnica de encadenamiento hacia atrás, el sistema sabrá qué preguntar y cuándo hacer la pregunta adecuada.

Los sistemas que utilizan el encadenamiento hacia atrás también se conocen como sistemas dirigidos por objetivos. Tienen una lista de objetivos y tratan de ejecutarlos. La lista agrega dinámicamente nuevos objetivos a la parte superior de la lista y empuja los viejos objetivos hacia abajo en la lista. La idea es que el sistema funcione en el objetivo superior de la lista una vez, después de que se ejecute y se elimine de la lista, y el siguiente objetivo se convierte en el objetivo superior e intente ejecutarlo. Este procedimiento se realiza hasta que no haya ningún objetivo que ejecutar en la lista.



En el encadenamiento hacia atrás, el sistema comienza con las reglas de alto nivel y se puede expandir a cualquier nivel necesario para llegar a la decisión agregando bloques de reglas que cubren detalles específicos de la decisión. El sistema puede reutilizar estas reglas en varios lugares. Si en el futuro se necesitan algunos criterios nuevos para demostrar un objetivo principal, se pueden agregar reglas adicionales al bloque de reglas y la llamada al sistema y probarlas, lo que facilita la adición de reglas y la expansión de un sistema.

El encadenamiento hacia atrás permite que el sistema le pida al usuario información sobre las reglas que tiene. A veces, el usuario no está en condiciones de responder estas preguntas o no tiene la información requerida. En este caso, agregar algunas reglas más al sistema ayudará al proceso de ejecución, y es posible que se soliciten estas reglas. información y el usuario podría responder.

## 1.8 Estrategia de resolución en el encadenamiento hacia atrás

Se aplica una estrategia de resolución en el encadenamiento hacia atrás para combinar subobjetivos después de buscar algunos axiomas con el fin de hacer una deducción del objetivo.

¿Cómo funciona una estrategia de resolución en el encadenamiento hacia atrás? Suponga que están disponibles las siguientes reglas:

```
R1: IF A AND B AND C
    THEN D
R2: IF D
    THEN E
```

Se dan los axiomas de que A, B y C son verdaderos. El objetivo es deducir E si un subobjetivo D es verdadero.

Para demostrar que E es verdadero, debemos asegurarnos de que el subobjetivo D sea verdadero. Se demuestra que el subobjetivo D es cierto ya que los hechos A, B y C son ciertos. Entonces E se deduce del hecho de que D es verdadera.

El encadenamiento hacia atrás funciona siempre que no se tenga en cuenta ningún subobjetivo. Si se cumplen todos los subobjetivos, se alcanza la meta.

## 1.9 Algoritmo de encadenamiento hacia atrás

El algoritmo de encadenamiento hacia atrás se puede representar de la siguiente forma:

- Se selecciona la regla que coincide con el objetivo.

```
IF la condición (parte IF) está vacía, solicite información al usuario.
ELSE
WHILE no termina, AND tenemos las reglas seleccionadas DO
```

- Agrega las condiciones de las reglas:

```
IF la condición no se cumple, THEN ponga la condición como un objetivo a resolver
END WHILE.
```

## 1.10 Búsqueda en profundidad en el encadenamiento hacia atrás

Suponga que se dan las siguientes reglas y en la siguiente figura se ilustra un encadenamiento hacia atrás en profundidad para estas reglas.

```
R1: IF A AND B AND C THEN D
R2: IF E AND F THEN A
R3: IF G AND H THEN B
```

```
R4: IF I OR J THEN C
\end{itemize}
```

```
\vspace{0.2cm}
```

```
\includegraphics[scale=0.35]{Z6}
```

```
\vspace{0.2cm}
```

Eche un vistazo a las siguientes reglas:

```
\vspace{0.2cm}
```

```
\begin{verbatim}
```

```
R1: IF el cliente tiene la prioridad
    THEN el cliente obtiene un descuento del 20%
R2: IF el cliente realiza una compra por más de $ 200
    THEN el cliente tiene prioridad
R3: IF el cliente es miembro del club de fans
    THEN el cliente tiene prioridad
```

Hecho (s) dados:

El cliente realiza una compra por más de \$ 200.

Objetivo: ¿El cliente obtendrá un 20% de descuento?

El sistema busca una regla que coincida con el objetivo, que es R1.

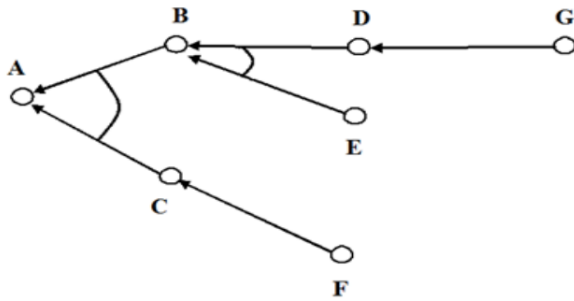
Luego, el sistema busca la condición de R1 y trata de completar y esta condición se convierte en el nuevo subobjetivo (objetivo máximo). En esta etapa, el sistema buscará información para satisfacer la condición.

Esto es, encontrar una regla que su parte THEN coincida con el nuevo objetivo máximo. No hay información pero la parte THEN de R2 coincide con el subobjetivo. La condición de R2 se convierte en el nuevo subobjetivo y en la base de datos hay un dato que cumple la condición (realiza una compra por más de 200 dólares). Entonces, la regla R2 se activa y el sistema la elimina de la lista de subobjetivos y el antiguo subobjetivo se convierte en el nuevo subobjetivo, que es R1. En esta etapa, se cumple la condición en R1 y R1 se dispara. La decisión final es que el cliente obtenga un descuento del 20%.

Al igual que el encadenamiento hacia adelante, el encadenamiento hacia atrás también se puede representar en forma de grafo. Suponga que están disponibles las siguientes reglas:

```
R1: IF A THEN B AND C
R2: IF B THEN D AND E
R3: IF C THEN F
R4: IF D THEN G
```

En la figura se muestra la representación gráfica del encadenamiento hacia atrás desde el nodo de salida del objetivo al nodo de premisa inicial:



### 1.11 Ventajas y desventajas del encadenamiento hacia atrás

A continuación se dan algunas ventajas del encadenamiento hacia atrás:

- El sistema dejará de procesar una vez que la variable tenga su valor.
- El sistema que utiliza el encadenamiento hacia atrás intenta establecer objetivos en el orden en que llegan a la base de conocimientos.
- Se dirige la búsqueda en encadenamiento hacia atrás.
- Durante la búsqueda, el encadenamiento hacia atrás considera aquellas partes de la base de conocimiento que están directamente relacionadas con el problema considerado o el encadenamiento hacia atrás nunca realiza inferencias innecesarias.
- El encadenamiento hacia atrás es una herramienta excelente para tipos específicos de problemas, como el diagnóstico y la depuración.
- Comparado con el encadenamiento hacia adelante, se solicitan pocos datos, pero se buscan muchas reglas.

También hay algunas desventajas del encadenamiento hacia atrás:

- Se debe conocer la meta para realizar el proceso de encadenamiento hacia atrás
- El proceso de implementación del encadenamiento hacia atrás es difícil.