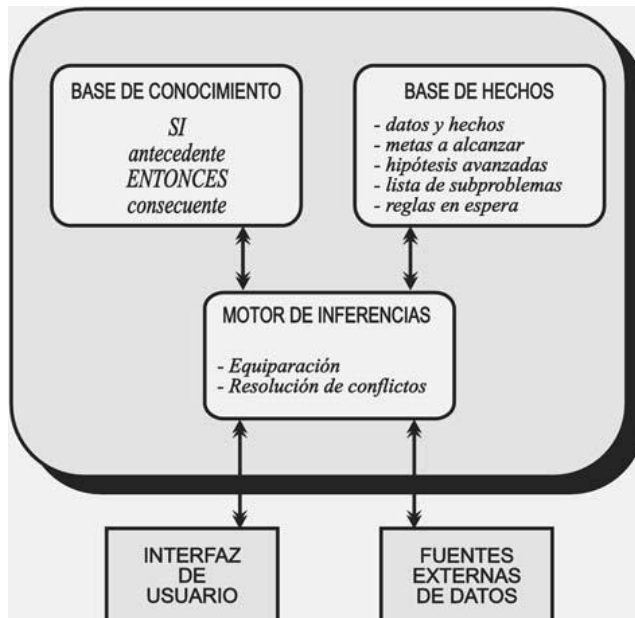


5.1 Introducción

Las reglas se utilizan para examinar un conjunto de datos y solicitar nueva información hasta llegar a un diagnóstico. Hoy en día las reglas, ampliadas con el uso de marcos y con algunos conceptos propios de las redes semánticas, constituyen el método estándar para la construcción de sistemas expertos.

5.2 Componentes básicos de los sistemas basados en reglas (SBRs).

Un sistema basado en reglas consta de **cinco elementos** básicos:



6.2.1 Base de Hechos

La **base de hechos (BH)** o **memoria de trabajo** contiene toda la información actual del **problema** o tarea a resolver, es decir, los datos y hechos establecidos hasta el momento en la tarea, las metas a alcanzar y las hipótesis avanzadas en el curso de una solución. En la BH es donde se guardan todos los cambios de información que se producen en el sistema durante cada proceso de inferencias. Es decir, en el sistema no se guarda más información que la que existe en la BH, que representa el estado actual del problema en curso de solución.

6.2.2 Base de conocimiento:

Contiene las reglas y, a veces, también algunas afirmaciones iniciales. Es específica del dominio en que el sistema puede considerarse experto (medicina, geología, diseño de circuitos...etc)

Una regla consta de dos partes: una izquierda, denominada condición o antecedente, y una derecha, llamada consecuente o de acción. Así, definiremos una regla como un par condición-acción. El antecedente contiene una lista de cláusulas a verificar y el consecuente una lista de acciones a ejecutar. Habitualmente las reglas se representan por medio de una flecha que conecta la condición con la acción:

Condición → Acción

Algunos ejemplos de reglas en distintos dominios de aplicación son los que se indican a continuación:

1. **Paciente menor de 10 años con manchas rojas y fiebre → Paciente con varicela**
2. **Coche no arranca → Revisar batería**
3. **El dólar baja → Comprar dólares**
4. **Envío urgente de un paquete → Incremento de tarifa base en 10 euros**

Las reglas operan sobre el espacio de trabajo de la BH. La condición expresa algún tipo de test sobre el contenido de la BH, que se puede verificar o no. Si se verifica el test de una regla, se puede ejecutar su parte de acción. La ejecución de una acción puede cambiar el contenido de la BH.

Las reglas son similares a los condicionales **If-Then** de la mayoría de los lenguajes de programación.

6.2.3 Motor de inferencia o intérprete de reglas

Es el elemento central del sistema; se encarga de coordinar la información procedente de todos los demás y de enviar los resultados de la inferencia al lugar oportuno

El motor de inferencias (MI), la estrategia de control o el intérprete de reglas es el **mecanismo que sirve para examinar la BH y decidir qué reglas se deben disparar**. El esquema general de funcionamiento se puede ver en el Algoritmo 3.1:

Algoritmo 3.1 Algoritmo genérico de funcionamiento del motor de inferencias.

```
1: BH = HechosIniciales;  
2: mientras NoVerificaCondiciónFinalización(BH) o NoseEjecutaAccióndeParada hacer  
3:   ConjuntoConflicto = Equiparar(BC,BH);  
4:   R=Resolver(ConjuntoConflicto);  
5:   NuevosHechos = Aplicar(R,BH);  
6:   Actualizar(BH,NuevosHechos);  
7: fin mientras
```

6.2.4 Interfaz de usuario:

Se encarga de solicitar al usuario la información necesaria y de mostrarle los resultados de la inferencia.

6.2.5 Fuentes externas de datos:

Es necesaria para tener acceso eficiente a toda la información relativa a los casos anteriores. Pueden ser base de datos.

6.3 Inferencia

Los SBR actuales usan, fundamentalmente, **dos modos de inferencia**:

1. **Encadenamiento hacia delante**
2. **Encadenamiento hacia atrás.**

También es posible encontrar SBR que usan modos de razonamiento mixtos.

Encadenamiento hacia delante

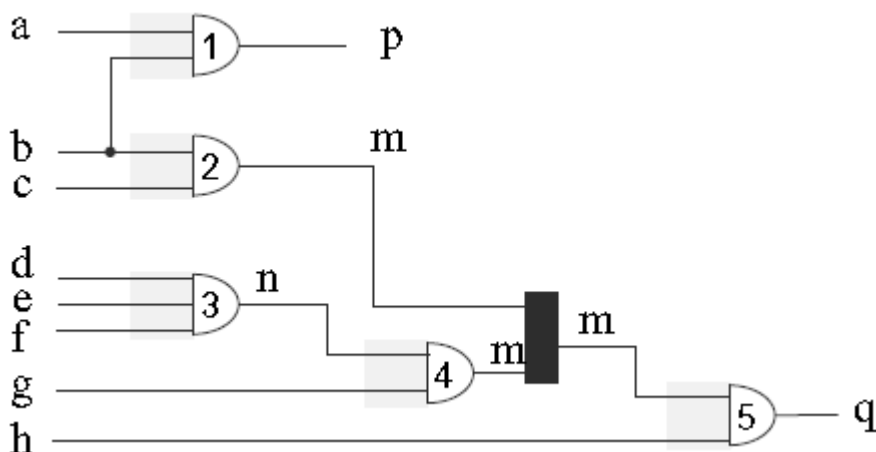
El encadenamiento hacia delante o razonamiento progresivo comienza con una colección de hechos o afirmaciones de partida, y aplica las reglas de la BC repetidas veces hasta que no se generen nuevos hechos.

Una regla se puede aplicar si todas las cláusulas del antecedente de la regla se verifican, teniendo en cuenta el contenido de la BH.

Cada vez que se aplica una regla se almacenan en la BH los resultados de las acciones del consecuente de dicha regla. El encadenamiento hacia atrás o razonamiento regresivo parte de un conjunto de hipótesis e intenta verificar estas hipótesis usando datos de la BH o datos externos (obtenidos, por ejemplo, a partir del usuario).

Si el número de reglas no es muy grande, uno puede obtener un grafo dirigido con las reglas (llamado red de inferencia), en el cual las cláusulas del antecedente son las entradas a los nodos y las acciones del consecuente son las salidas de los nodos que, a su vez, pueden ser las cláusulas del antecedente de otros nodos. En esta red, el antecedente que no es consecuente de ninguna otra regla de la red, se convierte en los hechos de partida; y el consecuente, que no es antecedente de ninguna otra regla, se convierte en la meta a alcanzar por el sistema.

La siguiente Figura muestra un ejemplo de red de inferencia con un conjunto de reglas y hechos simbólicos.



En este esquema se representan los hechos o elementos de la BH por medio de las letras que van desde la a hasta la q; aunque, en general, los antecedentes suelen ser más complejos. En la figura se muestra una red con las siguientes cinco reglas, representadas gráficamente mediante puertas de implicación:

- Regla 1: $a \& b \rightarrow p$
- Regla 2: $b \& c \rightarrow m$
- Regla 3: $d \& e \& f \rightarrow n$
- Regla 4: $n \& g \rightarrow m$
- Regla 5: $h \& m \rightarrow q$

Las reglas 2 y 4 tienen el mismo consecuente, m, que se corresponde con el caso en que se llega a una misma conclusión, siguiendo dos líneas de razonamiento independientes.

Esto se representa por medio de un rectángulo negro. A partir de esta red de inferencia, hay dos posibles formas de razonamiento, que consisten en:

- Buscar el conjunto de objetivos que se verifican a partir de un conjunto de entradas. Es lo que se conoce como encadenamiento hacia delante. En este tipo de razonamiento, la inferencia progresa en la red de izquierda a derecha.
- Determinar si se verifica un objetivo dado con ciertas entradas. Es lo que se conoce como encadenamiento hacia atrás. Aquí, la inferencia progresa en la red de derecha a izquierda.

6.3.1 Encadenamiento hacia delante

Como puede verse, la particularidad de este algoritmo es la etapa 3 de equiparación, en donde se seleccionan las reglas cuyos antecedentes se verifican, teniendo en cuenta el contenido de la BH.

Algoritmo 3.2 Algoritmo de encadenamiento hacia delante.

```

1: BH = HechosIniciales, ConjuntoConflicto = ExtraeCualquierRegla(BC);
2: mientras NoContenida(Meta,BH) y NoVacio(ConjuntoConflicto) hacer
3:   ConjuntoConflicto = Equiparar(Antecedente(BC),BH);
4:   si NoVacio(ConjuntoConflicto) entonces
5:     R=Resolver(ConjuntoConflicto);
6:     NuevosHechos = Aplicar(R,BH);
7:     Actualizar(BH,NuevosHechos);
8:   fin si
9: fin mientras
10: si Contenida(Meta,BH) entonces
11:   devolver "éxito";
12: fin si

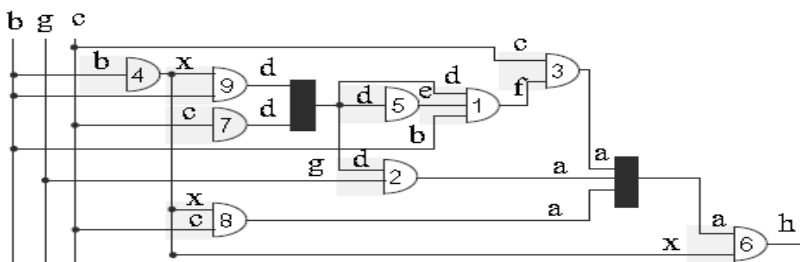
```

6.3.1.1 Ejemplo de aplicación del encadenamiento hacia delante.

Supongamos que disponemos de la siguiente BC:

- Regla 1: $b \& d \& e \rightarrow f$
- Regla 2: $d \& g \rightarrow a$
- Regla 3: $c \& f \rightarrow a$
- Regla 4: $b \rightarrow x$
- Regla 5: $d \rightarrow e$
- Regla 6: $a \& x \rightarrow h$
- Regla 7: $c \rightarrow d$
- Regla 8: $x \& c \rightarrow a$
- Regla 9: $x \& b \rightarrow d$

La red de inferencia de este conjunto de reglas puede verse en la siguiente Figura



Supongamos que partimos de los siguientes contenidos en la BH:

BH: {b, c}

Meta a alcanzar: h

Supongamos que una vez que se dispara una regla ya no se puede volver a disparar. Esta suposición es correcta en este ejemplo particular (pero no en general), ya que la ejecución repetida de una regla no modifica en nada la BH (en cada repetición se genera un hecho que ya está contenido en la BH desde la primera vez que se ha disparado la regla). Con esta suposición evitamos que una regla

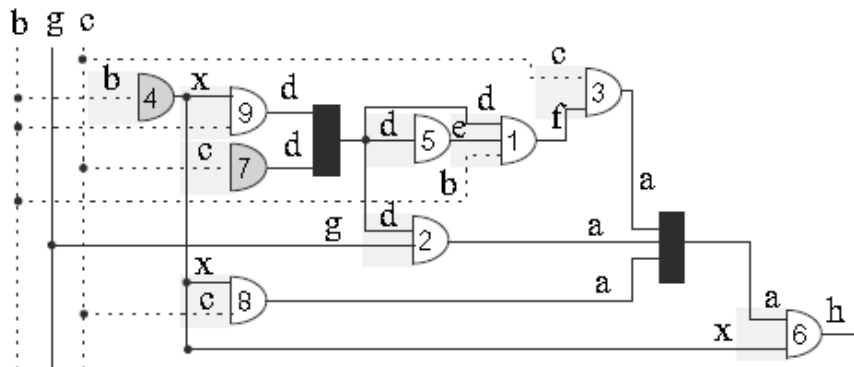
esté continuamente ejecutándose sin producir modificaciones en la BH (esta aproximación práctica se conoce como **principio de refracción**).

Estrategias a seguir: vamos a suponer dos estrategias distintas:

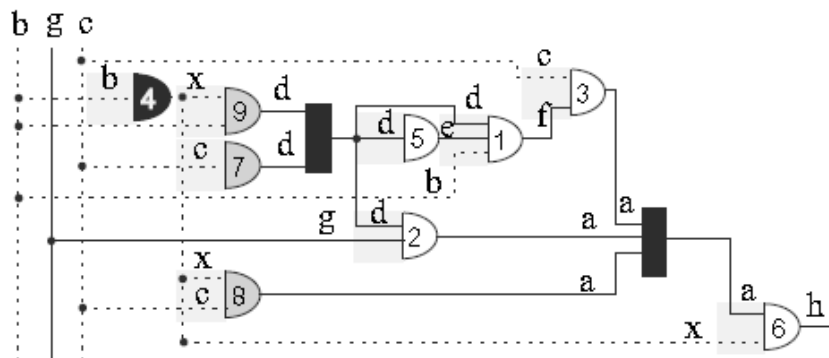
- **Estrategia 1:** Seleccionar la regla de menor número. Inicialmente las únicas reglas que se pueden aplicar en el **ciclo 1** son

ConjuntoConflicto= {4, 7} tal y como se muestra en la siguiente Figura

A continuación, se selecciona y aplica la Regla 4 (la de menor número). La BH se actualiza a **BH: {x, b, c}**

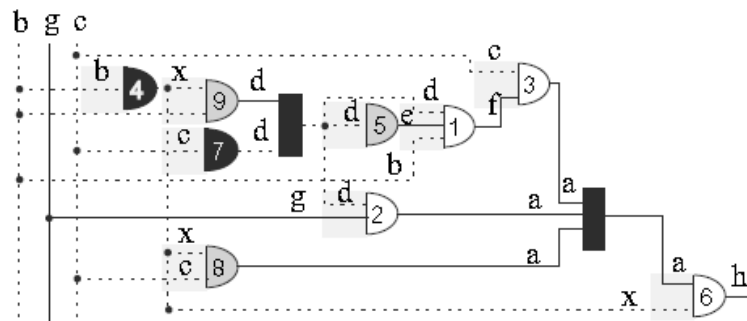


En el **ciclo 2**, las únicas reglas a aplicar son: **ConjuntoConflicto={7,8,9}**. Esto se muestra en la siguiente Figura.



Se selecciona y aplica la Regla 7, y se actualiza la BH a **BH: {d, x, b, c}**.

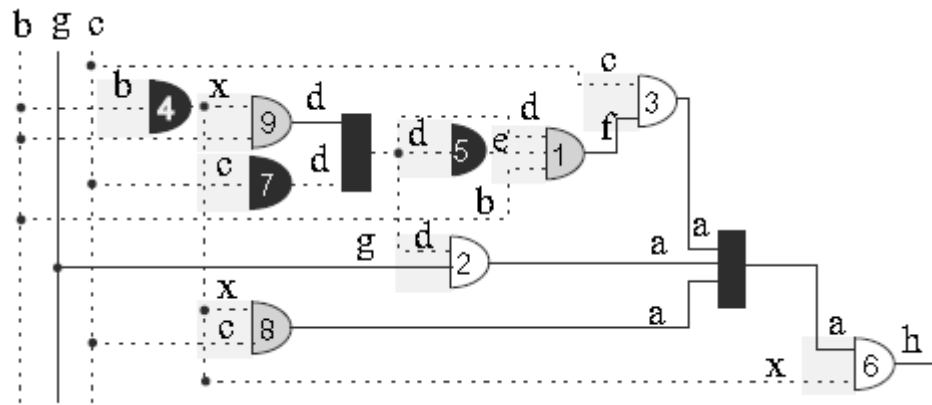
En el **tercer ciclo**, las reglas que se pueden aplicar son las siguientes, tal y como se puede apreciar en la Figura 3.6: **ConjuntoConflicto= {5, 8, 9}**



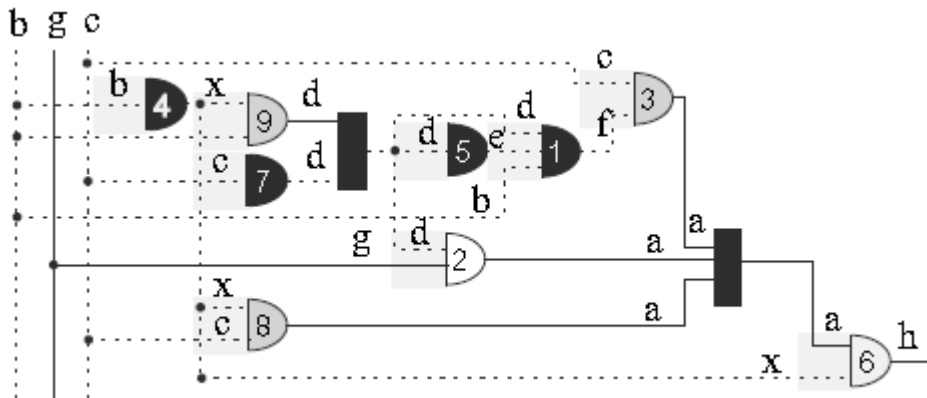
Se selecciona la Regla 5, se aplica y la BH se actualiza a **BH: {e, d, x, b, c}**

En el ciclo 4, tal y como se muestra en la siguiente Figura: **ConjuntoConflicto= {1, 8, 9}**

Se selecciona la Regla 1, se aplica y la BH se actualiza a **BH: {f, e, d, x, b, c}**

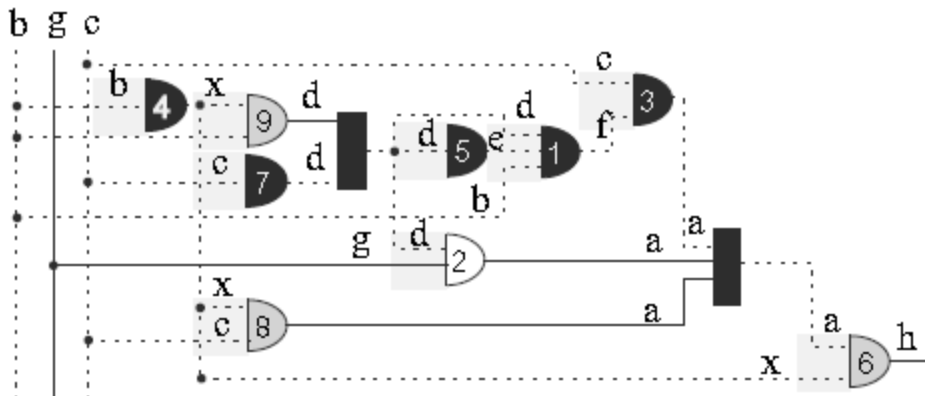


En el ciclo 5, tal y como se muestra en la siguiente Figura: ConjuntoConflicto= {3,8,9}



Se selecciona la Regla 3, se aplica y la BH se actualiza a **BH: {a, f, e, d, x, b, c}**

Por último, tal y como se muestra en la siguiente Figura: ConjuntoConflicto= {6, 8, 9}



Se selecciona la Regla 6, se aplica y la BH se actualiza a **BH: {h, a, f, e, d, x, b, c}**

Como la meta h está contenida en la BH, finaliza el proceso de inferencias alcanzando el éxito. En la siguiente Tabla se muestra todo este proceso de aplicación de la estrategia 1.

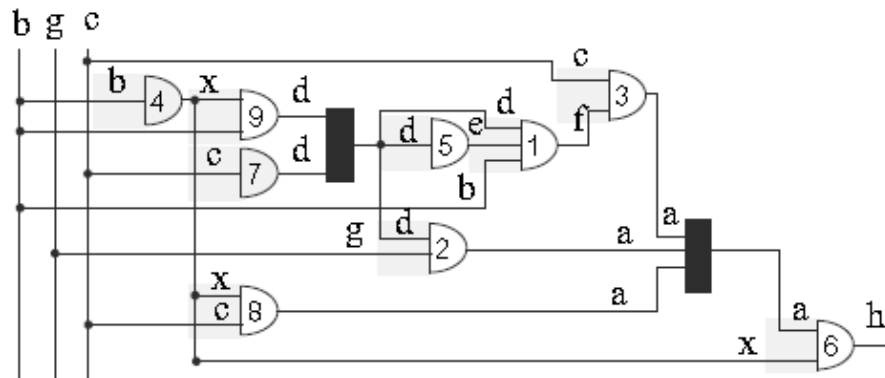
Ciclo	BH	Conjunto Conflicto	Resolución
1	{b, c}	{4, 7}	4
2	{x, b, c}	{7, 8, 9}	7
3	{d, x, b, c}	{5, 8, 9}	5
4	{e, d, x, b, c}	{1, 8, 9}	1
5	{f, e, d, x, b, c}	{3, 8, 9}	3
6	{a, f, e, d, x, b, c}	{6, 8, 9}	6

Estrategia 2: Seleccionar la regla que tiene **mayor número de cláusulas antecedentes** (es decir, la más restrictiva). Entre dos o más reglas con igual número de antecedentes, se elegirá la regla de menor número. Esta segunda estrategia se plantea para clarificar cómo la elección de una estrategia puede influir en el número de ciclos que dura el proceso de encadenamiento.

Partiendo de la red de inferencia de la Figura 3.3 y la BH inicial

BH: {b, c}

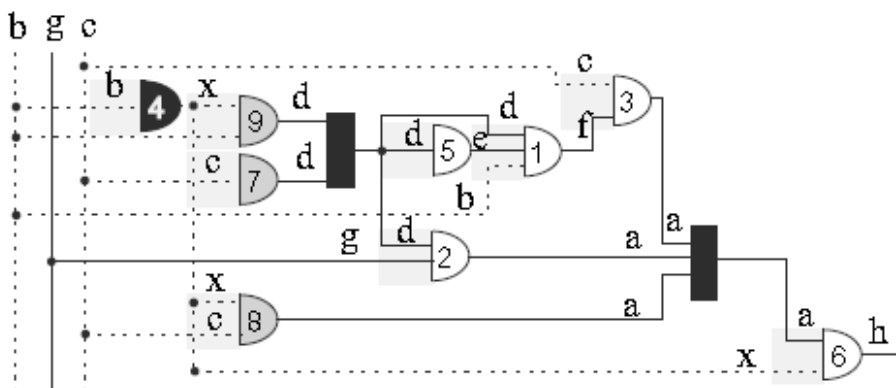
Meta a alcanzar: h



En el primer ciclo, tal y como muestra la siguiente Figura: ConjuntoConflicto= {4, 7}

Ya que tanto la Regla 4 como la 7 tienen una única cláusula antecedente, se selecciona y aplica la Regla 4 (la de menor número). La BH se actualiza a BH: {x, b, c}

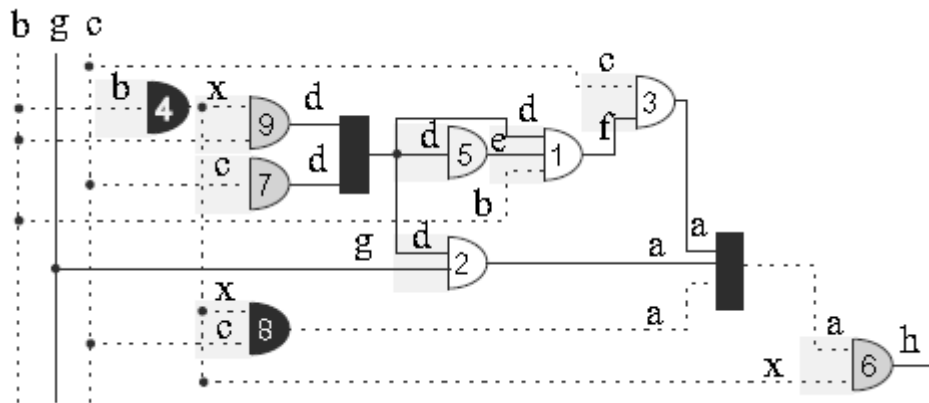
En el ciclo 2, las posibles reglas a aplicar son (véase la siguiente Figura):



ConjuntoConflicto= {7, 8, 9}

Las reglas más restrictivas, ambas con dos cláusulas, son la 8 y la 9. Se selecciona y aplica la 8 (menor número), y se actualiza la BH a BH: {a, x, b, c}

En el tercer ciclo, las reglas que se pueden aplicar son las siguientes, tal y como se puede apreciar en la siguiente Figura:



ConjuntoConflicto= {6, 7, 9}

Las reglas más restrictivas son la 6 y la 9. Se selecciona la Regla 6, se aplica y la BH se actualiza, de tal forma, que pasa a contener la meta. Por tanto, en este caso también se acabaría el proceso de inferencias con éxito. En la siguiente Tabla se muestra todo este proceso de aplicación de la estrategia

2. **BH: {h, a, x, b, c}**

Ciclo	BH	Conjunto Conflicto	Resolución
1	{b, c}	{4, 7}	4
2	{x, b, c}	{7, 8, 9}	8
3	{a, x, b, c}	{6, 7, 9}	6

Comparando la aplicación de las dos estrategias, hay que señalar que se necesitan 6 ciclos para alcanzar la meta con la primera estrategia, mientras que solamente 3 con la segunda. Como vemos, en este segundo caso el coste de aplicación de las reglas se reduce a la mitad. Sin embargo, el coste de selección de las reglas es mayor.

Siguiendo la primera estrategia, no hace falta obtener todo el conjunto conflicto; bastará con buscar la primera regla que verifique todas las cláusulas de sus antecedentes.

Una vez encontrada, se parará el proceso de búsqueda y se aplicará la regla. No obstante, dicha estrategia sólo se debe de aplicar si el mecanismo de inferencias puede retroceder sobre sus pasos cuando llega a una situación en la cual no puede aplicar más reglas, y aún quedan por explorar otras alternativas.

Por otra parte, el proceso de selección de la regla más restrictiva obtiene el conjunto de todas las reglas que se pueden aplicar en cada ciclo, y sobre ese conjunto selecciona la más restrictiva. Este proceso requiere más tiempo en computación que el anterior.

6.3.2 Encadenamiento hacia atrás

En un modo de razonamiento con encadenamiento hacia atrás, se especifica una meta objetivo y se trata de determinar si ese objetivo se verifica o no, teniendo en cuenta el contenido de la BH. Esto puede verse en el siguiente Algoritmo. En dicho procedimiento, se investigan los consecuentes de todas las reglas, con el fin de encontrar aquellas cuyos consecuentes contengan el objetivo a verificar. El subconjunto de reglas resultantes se examina para descubrir alguna que verifique todos sus antecedentes, teniendo en cuenta los contenidos de la BH. Si existe tal regla, entonces se verifica el

objetivo; en caso contrario, los antecedentes que no se pudieron verificar pasan a ser nuevos objetivos a verificar, en un control recursivo.

Algoritmo 3.3 Algoritmo de encadenamiento hacia atrás.

```
1: BH = HechosIniciales;
2: si Verificar (Meta,BH) entonces
3:   devolver "éxito";
4: si no
5:   devolver "fracaso";
6: fin si
```

Algoritmo 3.4 Procedimiento Verificar: comprueba si existe un conjunto de reglas verificando una meta.

```
1: Verificado=Falso;
2: si Contendida (Meta,BH) entonces
3:   devolver Verdadero;
4: si no
5:   ConjuntoConflicto = Equiparar(Consecuentes(BC),Meta);
6:   mientras NoVacio(ConjuntoConflicto) y No(Verificado) hacer
7:     R=Resolver(ConjuntoConflicto);
8:     Eliminar(R,ConjuntoConflicto);
9:     NuevasMetas=ExtraerAntecedentes(R), Verificado=Verdadero;
10:    mientras NoVacio(NuevasMetas) y Verificado hacer
11:      Meta=SeleccionarMeta(NuevasMetas);
12:      Eliminar(Meta, NuevasMetas);
13:      Verificado=Verificar(Meta,BH);
14:      si Verificado entonces
15:        Añadir(Meta,BH);
16:      fin si
17:    fin mientras
18:  fin mientras
19:  devolver(Verificado);
20: fin si
```

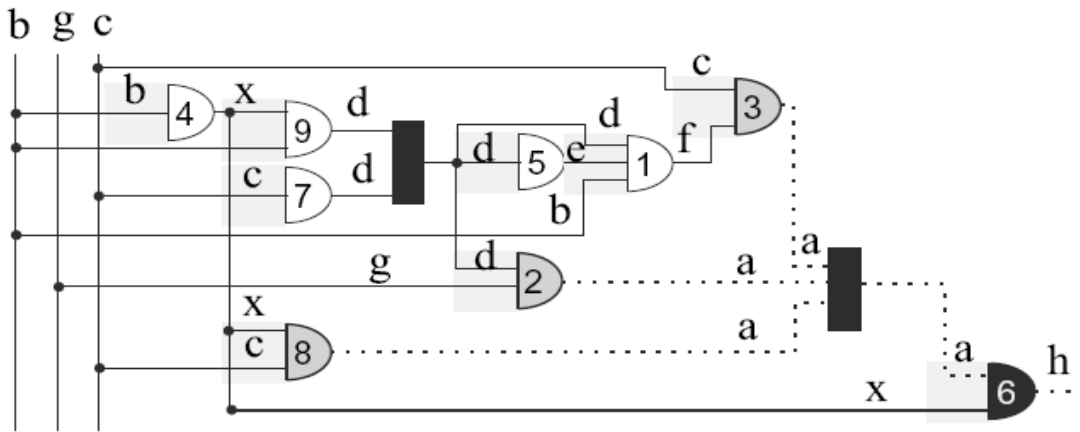
6.3.2.1 Ejemplo de aplicación del encadenamiento hacia atrás

Retomemos la BC representada por la red de inferencia de la Figura 3.3, la misma BH inicial que en el ejemplo del encadenamiento hacia delante y el mismo criterio de resolución del conjunto conflicto (es decir, la regla de menor número).

BH: {b, c}; Meta a alcanzar: h

A diferencia del encadenamiento hacia delante, ahora se recorre la red de inferencia de derecha a izquierda. Veámoslo. En el primer ciclo, la única regla que contiene en su parte consecuente la meta h es la Regla 6. Para verificar que se puede aplicar dicha regla, en el segundo ciclo, las nuevas metas a considerar serán las cláusulas del antecedente de dicha regla, es decir, Nuevas Metas: {a, x}

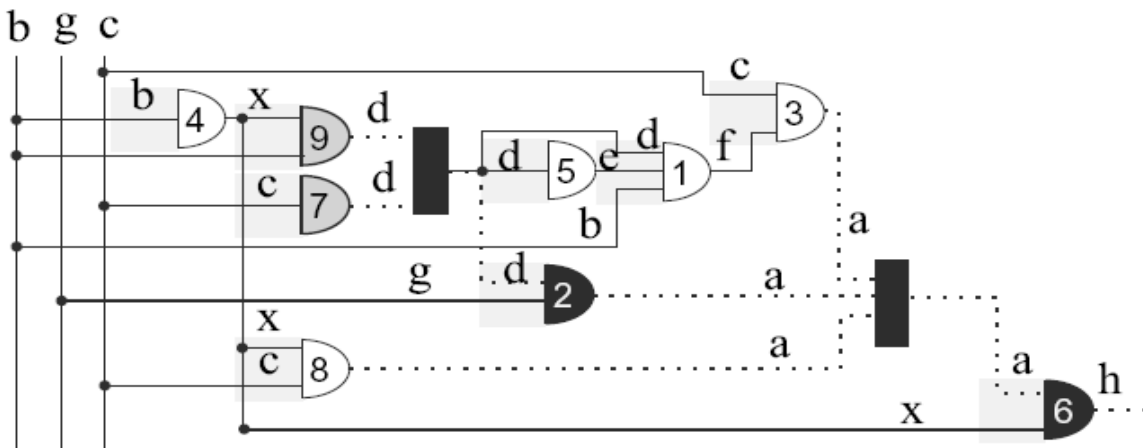
Comenzaremos revisando la **meta a**. Esta puede alcanzarse vía tres reglas distintas como muestra la siguiente Figura: ConjuntoConflicto= {2, 3, 8}



Aplicando el criterio de selección por orden de numeración, se selecciona primero la Regla 2. En el tercer ciclo, para alcanzar la meta a aplicando la Regla 2, se necesita que se verifiquen conjuntamente las dos cláusulas de su antecedente:

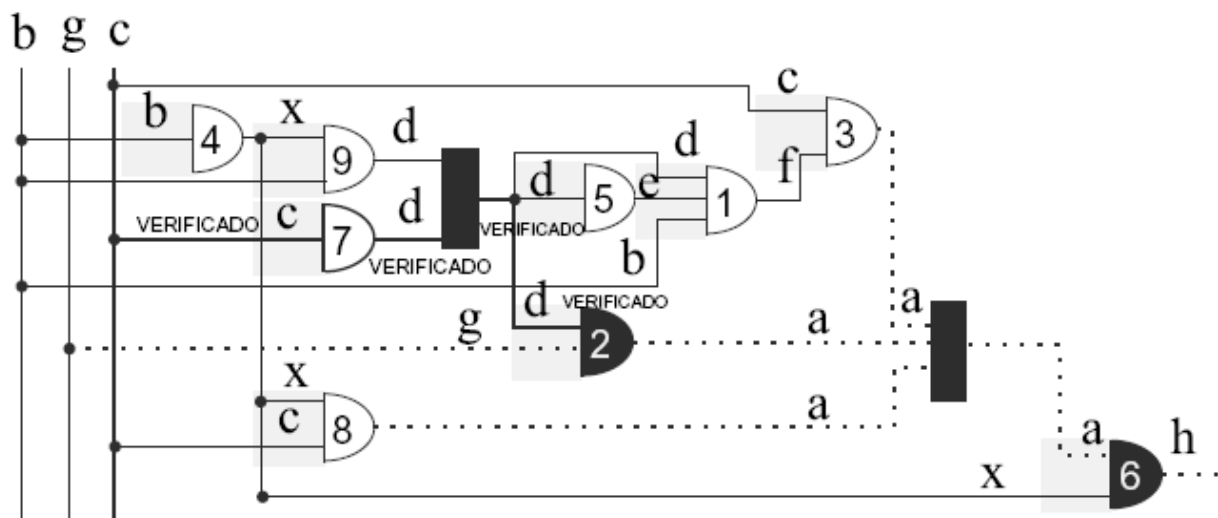
Nuevas Metas: {d, g}

Con el fin de verificar la meta d, se obtiene como muestra la siguiente Figura: ConjuntoConflicto= {7, 9}



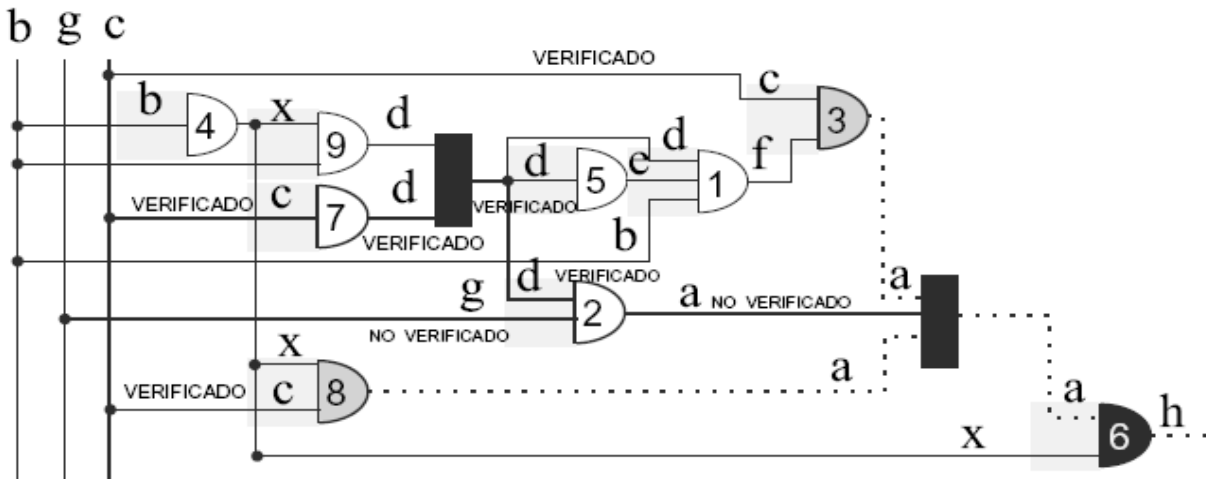
Si intentamos verificar d aplicando la Regla 7, debemos comprobar si su antecedente c está en la BH. Como es así, d pasa a estar verificado y se añade a la BH: **BH: {d, b, c}**

En el ciclo 4, se comprueba si se verifica g, como muestra la siguiente Figura.



Como g no está en la BH, la Regla 2 no se puede verificar.

En el ciclo 5, será necesario probar con otras reglas para comprobar a. Como muestra la siguiente Figura: ConjuntoConflicto= {3, 8}



Se selecciona la Regla 3 y exploran las cláusulas de su antecedente. El primero es c, que está contenido en la BH. El segundo es f, cuya verificación implica comprobar las cláusulas del antecedente de la Regla 1 en el ciclo 6, puesto que es la única que lo tiene como consecuente. Los antecedentes de f son d, que ya está verificado, e, que implica verificar la regla 5 en el ciclo 7, y b, que también está en la BH. En el ciclo 7, se comprueba que el antecedente d de la regla 5 se verifica, por lo que su consecuente e también lo hará. Se actualiza, así, la BH a: BH: {e, d, b, c}

Como consecuencia de ello, la Regla 1 se verifica y su consecuente f se añade a la

BH: BH: {f, e, d, b, c}

Por tanto, se verifican todos los antecedentes de la Regla 3 y se añade su consecuente a a la BH: BH: {a, f, e, d, b, c}

Por último, para verificar la meta h, se necesita comprobar la validez del hecho x.

En el ciclo 8, ConjuntoConflicto= {4}

Se selecciona la Regla 4 y la nueva meta es su antecedente b en el ciclo 9. Como dicho antecedente está contenido en la BH, se verifica el hecho x y finaliza de forma exitosa la aplicación del encadenamiento hacia atrás: BH= {h, x, a, f, d, b, c}

6.3.3 Reversibilidad

Ahora podríamos preguntarnos si, dado un problema concreto, es lo mismo usar un modo de razonamiento con encadenamiento hacia delante o hacia atrás. La respuesta es que no, ya que puede ser significativamente más eficiente buscar en una dirección que en otra. El uso de un tipo u otro depende del problema concreto. Aquí sólo podemos indicar un conjunto de estrategias generales, basadas en la forma de la red de inferencia y en el grado de conocimiento que el usuario del sistema tiene de los datos y de los objetivos del problema.

6.4 Técnicas de equiparación

La equiparación del antecedente de las reglas con el estado de la BH no siempre es obvia ya que el antecedente puede no describir situaciones particulares sino generales. Así, un tipo sencillo de equiparación surge cuando dicho antecedente contiene variables.

Otro problema, relacionado con la equiparación en el algoritmo general del motor de inferencias es la necesidad de examinar todas las reglas en cada ciclo de inferencias. Este proceso es poco eficiente, si hay que recorrer toda la BC y ésta contiene numerosas reglas. Se puede simplificar mediante:

1. Técnicas de indexación: Consisten en añadir a las reglas nuevas condiciones relacionadas con el punto de inferencia. Este recurso permite dividir el problema en varias etapas y agrupar las reglas en función de la etapa en la que se aplican. Por ejemplo, las reglas de un SBR en medicina podrían contener una cláusula objetivo que indicase la tarea a realizar: recopilación de antecedentes, diagnóstico, recomendación de tratamiento, aplicación del tratamiento, etc. Lamentablemente, esta forma de indexar las reglas sólo se puede aplicar si las condiciones de las reglas se equiparan exactamente con la BH.
2. Técnicas que aceleran el proceso de equiparación, sin necesidad de examinar toda la BC.

6.4.1 Equiparación con variables

Generalmente, el antecedente de las reglas describe situaciones generales que requieren ser expresadas mediante variables. Nos centraremos primero en ver cómo representar reglas con variables. A continuación, estudiaremos las diferentes situaciones que surgen cuando se equiparan las reglas expresadas mediante variables con la BH.

6.4.1.1 Sintaxis de reglas con variables

Para clarificar la sintaxis que proponemos, vamos a considerar nuevamente la siguiente regla, expresadas en lenguaje natural:

Paciente menor de 10 años con manchas rojas y fiebre → Paciente con varicela

La sintaxis propuesta es la siguiente:

1. Cada elemento de condición presente en el antecedente de una regla debe ir encerrado entre paréntesis y empezar por una constante.
2. Todos los elementos de condición del antecedente de una regla deben ir unidos por el operador lógico Y (representado por el símbolo \wedge).
3. Los elementos de condición están formados por átomos, donde un átomo puede ser una constante o una variable. Para distinguir las variables de las constantes, es usual señalar las primeras con un prefijo de interrogación. Así, la representación de las condiciones de la regla ejemplo incluiría:
 - (paciente ?p) para expresar el paciente ?p.
 - (edad ?p ?e) para indicar que la edad del paciente ?p es ?e.
 - (síntoma ?p fiebre) para representar que el paciente ?p presenta el síntoma fiebre.
4. Los elementos de condición pueden expresar algún tipo de prueba sobre las variables. En el ejemplo, para indicar que el paciente debe ser menor de 10 años, vamos a utilizar la palabra clave test: (test (<?e 10))
5. Las acciones del consecuente también pueden contener variables. Para la regla ejemplo, la acción del consecuente que expresa que el paciente padece varicela, se podría representar como (enfermedad ?p varicela)
6. Las acciones del consecuente pueden incluir dos predicados: Añadir (para agregar nuevos hechos a la BH) y Borrar (para eliminar hechos existentes de la BH). En el ejemplo, Añadir(enfermedad ?p varicela)

La representación completa de la regla ejemplo, siguiendo esta sintaxis será la siguiente:

SI (paciente ?p) \wedge (edad ?p ?e) \wedge (síntoma ?p fiebre) \wedge (síntoma ?p manchas-rojas)

ENTONCES Añadir (enfermedad ?p varicela)

3.4.1.2 Equiparación de reglas utilizando variables con los contenidos de la BH

La equiparación de reglas usando variables es un caso particular del problema de la unificación de términos en la resolución de primer orden. Teniendo en cuenta la sintaxis presentada, veamos ahora cómo se equiparan las reglas con variables con la BH. Como ejemplo, imaginemos un SBR para un hipódromo con los siguientes contenidos en la BH:

1. Hay cuatro caballos: Cometa, Bronco, Veloz y Rayo.
2. La relación entre ellos es la siguiente: Cometa es padre de Veloz y Bronco, y Veloz es padre de Rayo.
3. Dos de ellos son muy rápidos: Bronco y Rayo.

Primeramente, nos deberemos plantear cómo representar la BH. Una opción sería la siguiente:

- (caballo ?x) para expresar el caballo ?x.
- (padre ?x ?y) para indicar que el caballo ?x es padre de ?y.
- (rápido ?x) para representar que el caballo ?x es rápido.

Los contenidos concretos de la BH inicial serán los siguientes:

- (caballo Cometa), (caballo Bronco), (caballo Veloz), (caballo Rayo)
- (padre Cometa Veloz), (padre Cometa Bronco), (padre Veloz Rayo)
- (rápido Bronco), (rápido Rayo)

Durante la equiparación, pueden darse las siguientes situaciones:

1. Una variable aparece una sola vez en la parte de condición de una regla. En este caso, la variable se equipara con cualquier valor que ocupe la misma posición en un elemento de la BH.

Por ejemplo, si la regla es **SI** (caballo ?x) **ENTONCES** se equipara con todos los elementos de la BH cuyo valor para la primera posición es caballo. En nuestro ejemplo, se equiparía con (caballo Cometa), (caballo Bronco), (caballo Veloz) y (caballo Rayo) para los valores de Cometa, Bronco, Rayo y Veloz para la variable x.

2. Una variable aparece más de una vez en la parte de condición de una regla. En este caso, la variable debe equiparse siempre con el mismo valor en todas las ocurrencias de la regla. Por ejemplo, si la regla es:

SI (caballo ?x) \wedge (rápido ?x) ENTONCES (candidato-competición ?x)

La primera condición se equipara con todos los elementos de la BH cuyo valor para la primera posición es caballo, y la segunda condición se equipara con todos los elementos de la BH cuyo valor para la primera posición es rápido. La regla se equipara con los elementos de la BH que verifiquen el mismo valor para la variable x en ambas condiciones. Es decir,

- a) Equiparación de la primera condición: (caballo Cometa), (caballo Bronco), (caballo Veloz) y (caballo Rayo) para los valores de Cometa, Bronco, Rayo y Veloz para la variable x.
 - b) Equiparación de la segunda condición: (rápido Bronco) y (rápido Rayo) para los valores de Bronco y Rayo para la variable x.
 - c) Equiparación de la regla: (caballo Bronco) \wedge (rápido Bronco) y (caballo Rayo) \wedge (rápido Rayo) para los valores de Bronco y Rayo para la variable x.
3. Varias variables aparecen en la parte de condición de una regla. Se pueden equiparar variables distintas con un mismo valor. Por ejemplo, si la regla es

SI (caballo ?x) \wedge (padre ?x ?y) \wedge (rápido ?y) ENTONCES ...

Resultan las siguientes equiparaciones:

- a) Equiparación de la primera condición: (caballo Cometa), (caballo Bronco), (caballo Veloz) y (caballo Rayo) para los valores de Cometa, Bronco, Rayo y Veloz para la variable x.
 - b) Equiparación de la segunda condición: (padre Cometa Veloz), (padre Cometa Bronco) y (padre Veloz Rayo) para los pares de valores (Cometa, Veloz), (Cometa, Bronco) y (Veloz, Rayo) para las variables x e y, respectivamente.
 - c) Equiparación de la tercera condición: (rápido Bronco) y (rápido Rayo) para los valores de Bronco y Rayo para la variable y.
 - d) Equiparación de la primera y segunda condición: (caballo Cometa) \wedge (padre Cometa Veloz), (caballo Cometa) \wedge (padre Cometa Bronco), y (caballo Veloz) \wedge (padre Veloz Rayo) para los pares de valores (Cometa, Veloz), (Cometa, Bronco) y (Veloz, Rayo) para las variables x e y, respectivamente.
 - e) Equiparación de la regla: (caballo Cometa) \wedge (padre Cometa Bronco) \wedge (rápido Bronco) y (caballo Veloz) \wedge (padre Veloz Rayo) \wedge (rápido Rayo) para los pares de valores (Cometa, Bronco) y (Veloz, Rayo) para las variables x e y, respectivamente.
4. Una o varias variables aparecen en una o varias condiciones negadas de una regla. La equiparación se da si se cumplen las dos siguientes restricciones:
- a) Se equiparan todas las condiciones no negadas para algún o algunos elementos de la BH.
 - b) No existe ningún elemento en la BH que pueda equiparar las condiciones negadas.

La aplicación de estas dos restricciones se conoce como Hipótesis del Mundo Cerrado: Todo hecho que no esté en la BH, durante la equiparación de una regla, se considera falso. Por ejemplo, si la regla es SI (caballo ?x) \wedge \neg (rápido ?x) ENTONCES (candidato-reserva ?x) resultan las siguientes equiparaciones:

- a) Equiparación de la primera condición: (caballo Cometa), (caballo Bronco), (caballo Veloz) y (caballo Rayo) para los valores de Cometa, Bronco, Rayo y Veloz para la variable x.
- b) Equiparación de la segunda condición sin negar: (rápido Bronco) y (rápido Rayo) para los valores de Bronco y Rayo para la variable x.
- c) Equiparación de la regla: (caballo Cometa) y (caballo Veloz) para los valores de Cometa y Veloz para la variable x.

Se llama instanciación al par formado por una regla y los elementos de la BH que equiparan dicha regla. En el ejemplo anterior, la equiparación ha dado lugar a dos instanciaciones (conjunto conflicto):

1. {SI (caballo ?x) \wedge \neg (rápido ?x) ENTONCES (candidato-reserva ?x), (?x Cometa)}
2. {SI (caballo ?x) \wedge \neg (rápido ?x) ENTONCES (candidato-reserva ?x), (?x Veloz)}

En la primera instanciación la variable x queda ligada al valor Cometa, y en la segunda al valor Veloz. Esta ligadura se realiza en la BH y no en la regla. Así, durante la fase de acción, no se ejecuta la regla sino una instanciación de ella. Si en el consecuente de una regla aparece la variable (como en el ejemplo), durante la fase de acción la variable se sustituye por su valor ligado en la instanciación. En el ejemplo, si se selecciona aleatoriamente una instanciación del conjunto conflicto, la ejecución de la primera instanciación añadiría un nuevo hecho a la BH, actualizándose ésta al siguiente contenido:

- (caballo Cometa), (caballo Bronco), (caballo Veloz), (caballo Rayo)
- (padre Cometa Veloz), (padre Cometa Bronco), (padre Veloz Rayo)
- (rápido Bronco), (rápido Rayo), (candidato-reserva Cometa)

EJERCICIOS

REGLAS

- R₁:** SI x es carnívoro \wedge x es color leonado \wedge x tiene manchas oscuras ENTOCES x es una ONZA
- R₂:** SI x da leche ENTOCES x es un mamífero
- R₃:** SI x tiene plumas ENTOCES x es ave
- R₄:** SI x vuela \wedge x es pone huevos ENTOCES x es ave
- R₅:** SI x es ave \wedge x no vuela \wedge x tiene cuello alto ENTOCES x es AVESTRUZ
- R₇:** SI x es mamífero \wedge x tiene pezuñas ENTOCES x es ungulado
- R₈:** SI x es mamífero \wedge x rumia ENTOCES x es ungulado
- R₉:** SI x tiene pelo ENTOCES x es un mamífero
- R₁₀:** SI x es carnívoro \wedge x es color leonado \wedge x tiene franjas negras ENTOCES x es TIGRE
- R₁₁:** SI x es ungulado \wedge x tiene patas largas \wedge x tiene cuello largo \wedge x es color leonado \wedge x tiene manchas oscuras ENTOCES x es JIRAFA
- R₁₂:** SI x es mamífero \wedge x come carne ENTOCES x es carnívoro
- R₁₃:** SI x es ungulado \wedge x es color blanco \wedge x tiene franjas negras ENTOCES x es CEBRA
- R₁₄:** SI x es ave \wedge x no vuela \wedge x nada ENTOCES x es PINGÜINO
- R₁₅:** SI x es ave \wedge x vuela \wedge x vive en rocas ENTOCES x es ALBATROS
- R₁₆:** SI x es ave \wedge x vuela \wedge x tiene garras ENTOCES x es AGUILA
- R₁₇:** SI x es ave \wedge x vuela \wedge x vive en hogar ENTOCES x es GALLINA

HECHOS (ENCADENAMIENTO HACIA ADELANTE)

H₁: z tiene pelo
H₂: z tiene patas largas
H₃: z rumia
H₄: z tiene cuello largo
H₅: z es color leonado
H₆: z tiene manchas oscuras

R9 →

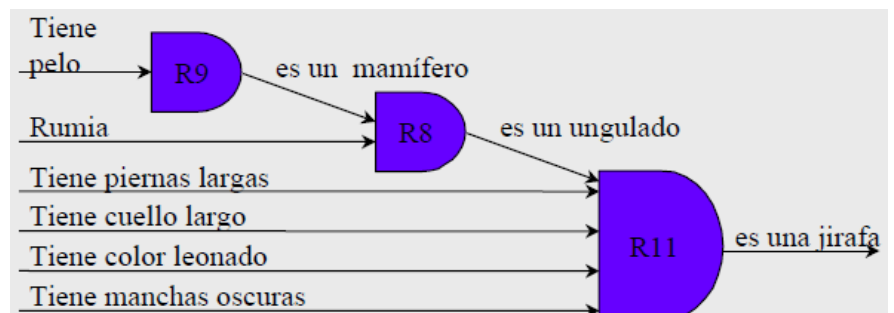
H₁: z tiene pelo
H₂: z tiene patas largas
H₃: z rumia
H₄: z tiene cuello largo
H₅: z es color leonado
H₆: z tiene manchas oscuras
z es un Mamífero

R8 →

H₁: z tiene pelo
H₂: z tiene patas largas
H₃: z rumia
H₄: z tiene cuello largo
H₅: z es color leonado
H₆: z tiene manchas oscuras
z es un Mamífero
z es un ungulado

R11 →

H₁: z tiene pelo
H₂: z tiene patas largas
H₃: z rumia
H₄: z tiene cuello largo
H₅: z es color leonado
H₆: z tiene manchas oscuras
z es un Mamífero
z es un ungulado
z es un JIRAFA



HYPÓTESIS: Z es ONZA

HECHOS (ENCADENAMIENTO HACIA ATRAS)

Y tiene pelo
Y es de color leonado
Y tiene manchas oscuras

R_1 necesita: Z sea carnívoro, de color leonado y tiene manchas oscuras, se busca que Z es carnívoro

Existen dos reglas:

R_6 = debe verificar que Z es mamífero, de nuevo existen dos alternativas R_2 y R_9

R_9 : Z tiene pelo luego Z es mamífero para que sea carnívoro se requerirá que Z come carne o tenga garras o, **aunque se tiene los hechos de: Z es de color Leonardo y Z tiene manchas oscuras.**

No se cumple carnívoro

Se concluye NO Z es ONZA