



UNIVERSIDAD DE MURCIA

SISTEMAS INTELIGENTES

Práctica 2: Sistema Basado en Reglas

López Toledo, Francisco
francisco.lopezt@um.es
48852366K
Grupo 1.1
Convocatoria: Enero
Fecha: 5 de Enero de 2020

Garrido Carrera, Maria del Carmen
carmengarrido@um.es
Profesora titular
Grupo 1.1

Índice

1. Definición de Sistema Basado en Reglas (SBR)	2
2. Diseño de elementos del motor de inferencias	4
2.1. Equiparación-Conjunto Conflicto	4
2.2. Condición de parada	4
3. Aplicación del sistema basado en reglas diseñado	5
3.1. Situación 1: Identificación de Frutas	5
3.2. Situación 2: Detección de Inundaciones	15

1. Definición de Sistema Basado en Reglas (SBR)

Los **Sistemas Basados en Reglas (SBR)** se inspiran en los sistemas de deducción en lógica proposicional o de primer orden e utilizan la estructura de inferencia modus ponens (razonamiento deductivo) para obtener conclusiones lógicas. En muchas ocasiones, se trata de usar reglas proporcionadas por un experto dentro del dominio en el que estamos trabajando para hacer deducciones o elecciones, cuando dichas reglas proceden de un experto humano se les conoce también como **sistemas expertos**.

Estas deducciones inferidas deben de calcularse rápidamente ya que serán necesarias para ayudar a un profesional en la toma de decisiones. Sin embargo, no deben de ser el único factor en la toma de decisiones en campos donde estas tengan un peso crucial ya que emulan la habilidad de un experto humano y pueden no ser completamente fiables. La recolección de conocimiento de calidad es la mayor dificultad a la hora de diseñar un sistema basado en reglas fiable.

Un sistema basado en reglas esta formado por tres elementos:

- *Una Base de Conocimiento (BC)*: Contiene las reglas que codifican todo el conocimiento del dominio. Una regla se puede definir como un par condición-acción. El antecedente contiene una lista de cláusulas a verificar y el consecuente una lista de acciones a ejecutar.
- *Una Base de Hechos (BH)*: Contiene hechos establecidos como verdaderos, tanto datos de entrada como conclusiones inferidas, metas a alcanzar y subproblemas. También se llama Memoria de trabajo. Este elemento representa el estado actual del problema.
- *Un Mecanismo de Inferencias (MI)*: Selecciona las reglas que se pueden aplicar y las ejecuta, con el objetivo de obtener alguna conclusión. También se llama Motor de Inferencias. Este mecanismo algorítmico utiliza la base de conocimientos para obtener conclusiones a partir de la memoria de trabajo. Dichas conclusiones serán almacenadas en la misma base de hechos.

Algoritmo 1: Algoritmo genérico de un Motor de Inferencias

Datos: BH, BC

Resultado: BH'

```
1 BH = HechosIniciales;
2 mientras !VerificaCondicionFinalizacion(BH) and
   !EjecutaAccionDeParada hacer
3   ConjuntoConflicto = Equiparar(BC, BH);
4   R = Resolver(ConjuntoConflicto);
5   NuevosHechos = Aplicar(R, BH);
6   Actualizar(BH, NuevosHechos);
7 fin
```

El criterio que sigue este motor de inferencias para la resolución de conflictos es importante ya que de esto depende el tiempo de respuesta del sistema ante cambios del entorno y la facultad del sistema de ejecutar secuencias de acciones relativamente largas (reglas más prometedoras).

Las principales técnicas de resolución de conflictos son las siguientes:

- **Según la base de conocimiento** (criterio estático): selecciona las reglas ordenadas por un criterio, ya sea por pesos de dichas reglas o por el número de antecedente de las reglas.
- **Según la base de hechos** (criterio dinámico): seleccionar las reglas que usan elementos más recientes de la memoria de trabajo.
- **Según la ejecución** (criterio dinámico): selecciona reglas no utilizadas previamente.

Existen dos instancias de un motor de inferencias:

- **Encaminamiento hacia delante** (forward-chaining)[1]: El encadenamiento hacia delante comienza con la base de hechos iniciales y usa reglas de inferencia para extraer más datos (de un usuario final, por ejemplo) hasta que se alcanza un objetivo. Un motor de inferencia que utiliza el encadenamiento hacia delante busca reglas de inferencia hasta que encuentra una donde se sabe que el antecedente es verdadero. Cuando se encuentra dicha regla, el motor puede concluir o inferir el consecuente, lo que resulta en la adición de nueva información a sus datos.

- **Encaminamiento hacia atrás** (backward-chaining)[2]: El encadenamiento hacia atrás comienza con una lista de objetivos (o una hipótesis) y funciona hacia atrás desde el consecuente al antecedente para ver si algún dato respalda alguno de estos consecuentes. Un motor de inferencia que utiliza el encadenamiento hacia atrás buscaría las reglas de inferencia hasta que encuentre una con una consecuente que coincida con un objetivo deseado. Si se sabe que el antecedente de esa regla es verdadero, entonces se agrega a la lista de objetivos (para que el objetivo sea confirmado, uno también debe proporcionar datos que confirmen esta nueva regla).

2. Diseño de elementos del motor de inferencias

En este apartado se va a dar una explicación del diseño seguido a la hora de implementar en el motor de inferencias la fase de equiparación-conjunto conflicto y de la condición de parada del algoritmo. Además, cabe destacar que se ha optado por la forma de razonamiento de encadenamiento hacia delante (data-driven) con un criterio de resolución de conflictos estático basado en prioridades o pesos.

2.1. Equiparación-Conjunto Conflicto

A la hora de implementar esta característica del motor de inferencia, realizaremos una búsqueda en la base de conocimientos comprobando, si algún hecho de la base de hechos cumple todos los antecedentes de alguna regla de la base de conocimientos y no ha sido ya aplicada anteriormente, entonces se procederá a añadir dicha regla al conjunto conflicto.

Una vez añadidas las reglas al conjunto conflicto, el algoritmo procederá a elegir una regla del conjunto conflicto y dispararla, a este proceso se le denomina **Resolver** y no es parte de la equiparación-conjunto conflicto pero es importante tenerlo en cuenta.

Para resolver los conflictos se aplicará un criterio estático tal y como he mencionado antes, en el cual se elegirá la regla con peso mayor del conjunto conflicto. Sin embargo, en caso de que se de empate de prioridad en todo el conjunto conflicto se elegirá la regla con identificador más pequeño.

Algoritmo 2: Función equiparar

Datos: ReglasAplicadas, BC, BH, ConjuntoConflicto

Resultado: ConjuntoConflicto

```
1 para cada Regla  $r$  : BC hacer
2   si  $\neg \text{Contiene}(r, \text{ReglasAplicadas})$  and
    $\text{cumpleTodosLosAntecedentes}(r, \text{BH})$  entonces
3     | ConjuntoConflicto = ConjuntoConflicto +  $r$ ;
4   fin
5 fin
```

2.2. Condición de parada

En cuanto a la implementación de la condición de parada del algoritmo, tendremos que parar la ejecución del algoritmo si se ha inferido en la base de hechos el objetivo que se está buscando ó en el caso de que no se pueda aplicar ninguna regla más y no se haya encontrado el objetivo.

Algoritmo 3: Condición de parada

```
1 si Vacío(ConjuntoConflicto) or Contiene(objetivo, BH) entonces
2   | parar;
3 fin
```

3. Aplicación del sistema basado en reglas diseñado

En este apartado se muestran los ficheros de salida correspondientes para dos situaciones. Los ficheros que el formato *Salida1-XXX.txt* contendrán toda la información para ver el razonamiento que ha seguido el sistema para inferir o no una solución. Mientras que, los ficheros con el formato *Salida2-XXX.txt* contendrá solo el razonamiento final sin detalles sobre los elementos que utiliza el sistema para inferir conclusiones para quizás la interpretación de un usuario final.

3.1. Situación 1: Identificación de Frutas

En esta situación, el sistema trata de resolver el problema de identificar frutas, para ello podemos observar en el archivo *Salida1-F1.txt* que en primer lugar enumeramos los datos iniciales conocidos (memoria de trabajo). Una vez recogidos estos datos, pasamos a la fase de equiparacion-conjunto conflicto en la cual se observa que se añaden dos reglas al conjunto conflicto ya que por ejemplo la regla **R5** tiene como antecedente que la forma es redonda y que el diametro es menor de 10, hechos los cuales podemos consultar que pertenecen a la base de hechos.

Por tanto, podemos inferir tras resolver el conflicto existente entre **R5** y **R6** que la clase del frutal es un arbol. Se actualizará la memoria de trabajo con este nuevo hecho y seguirá el procedimiento, en este caso, disparando la regla **R6** ya que no hay otra en el conjunto conflicto, infiriendo que el tipo de semilla es hueso. Posteriormente, gracias a inferir los hecho anteriores y a los datos proporcionados por el usuario (color = rojo) podemos concluir con que la fruta que identificamos es una cereza.

Los demás archivos *Salida1-XXX.txt* seguirán el mismo razonamiento que el anteriormente ejemplificado.

Archivo: **Salida1-F1.txt**

```
-DOMINIO DE LA APLICACION-
  ->IDENTIFICACION DE FRUTAS

-OBJETIVO-
  ->fruta

BASE DE HECHOS INICIAL :
  [ color = rojo ]
  [ diametro = 3 ]
  [ nsemillas = 1 ]
  [ forma = redonda ]

-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-

  SE INSERTA EN EL CONJUNTO CONFLICTO:
  ->R5: SI forma = redonda Y diametro < 10
        ENTONCES clasefrutal = arbol; Peso: 0

  SE INSERTA EN EL CONJUNTO CONFLICTO:
  ->R6: SI nsemillas = 1 ENTONCES tiposemilla = hueso; Peso: 0

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
  ->R5: SI forma = redonda Y diametro < 10
        ENTONCES clasefrutal = arbol; Peso: 0
```

~~–FIN FASE RESOLVER–~~

BASE DE HECHOS ACTUALIZADA:

[color = rojo]
[diametro = 3]
[clasefrutal = arbol]
[nsemillas = 1]
[forma = redonda]

~~–FASE DE EQUIPARACION–CONJUNTO CONFLICTO–~~

→Ninguna regla insertada en el conjunto conflicto

~~–FIN FASE EQUIPARACION–CONJUNTO CONFLICTO–~~

~~–FASE RESOLVER–~~

~~–Regla disparada–~~

→R6: SI nsemillas = 1 ENTONCES tiposemilla = hueso; Peso: 0

~~–FIN FASE RESOLVER–~~

BASE DE HECHOS ACTUALIZADA:

[tiposemilla = hueso]
[color = rojo]
[diametro = 3]
[clasefrutal = arbol]
[nsemillas = 1]
[forma = redonda]

~~–FASE DE EQUIPARACION–CONJUNTO CONFLICTO–~~

SE INSERTA EN EL CONJUNTO CONFLICTO:

→R13: SI clasefrutal = arbol Y color = rojo Y tiposemilla = hueso
ENTONCES fruta = cereza; Peso: 10

~~–FIN FASE EQUIPARACION–CONJUNTO CONFLICTO–~~

~~–FASE RESOLVER–~~

~~–Regla disparada–~~

→R13: SI clasefrutal = arbol Y color = rojo Y tiposemilla = hueso
ENTONCES fruta = cereza; Peso: 10

~~–FIN FASE RESOLVER–~~

BASE DE HECHOS ACTUALIZADA:

[forma = redonda]
[fruta = cereza]
[nsemillas = 1]
[clasefrutal = arbol]
[diametro = 3]
[color = rojo]

```
[tiposemilla = hueso]
```

Se ha encontrado solución tras aplicar las siguientes reglas:

→R5: SI forma = redonda Y diametro < 10

ENTONCES clasefrutal = arbol; Peso: 0

→R6: SI nsemillas = 1 ENTONCES tiposemilla = hueso; Peso: 0

→R13: SI clasefrutal = arbol Y color = rojo Y tiposemilla = hueso

ENTONCES fruta = cereza; Peso: 10

Solución inferida: fruta = cereza

Archivo : **Salida2-F1.txt**

–DOMINIO DE LA APLICACION–

→IDENTIFICACION DE FRUTAS

–OBJETIVO–

→fruta

Se ha encontrado solución tras aplicar las siguientes reglas:

→R5: SI forma = redonda Y diametro < 10

ENTONCES clasefrutal = arbol; Peso: 0

→R6: SI nsemillas = 1 ENTONCES tiposemilla = hueso; Peso: 0

→R13: SI clasefrutal = arbol Y color = rojo Y tiposemilla = hueso

ENTONCES fruta = cereza; Peso: 10

Solución inferida: fruta = cereza

Para la base de hechos *BH-F2.txt*, se inicializa la memoria de trabajo tras leer dicha base de hechos y se realiza la fase de equiparación-conjunto conflicto en la cual se añaden al conjunto conflicto la regla R1 y R5 al tener en la base de hechos que el número de semillas es 10 (R1), tener la forma redonda y tener un diámetro menor que 10. Al existir empate de prioridades, se resolverá el conflicto disparando la regla R1 al ser la de menos identificador y se actualizará la base de hechos con la conclusión inferida de que el tipo de semilla es múltiple. En este caso, no se puede insertar ninguna regla más en el conjunto conflicto, por lo tanto, la fase de resolver disparará la única regla restante en el conjunto conflicto (R5), actualizando la base de hecho con el hecho de que la clase del frutal es un árbol.

Posteriormente, se insertará en el conjunto conflicto la regla R16 tras comprobar que los antecedentes de esta regla están en la base de hechos, disparándose a continuación al ser la única regla en el conjunto conflicto y llegando a inferir la conclusión de que la fruta es una manzana. El algoritmo comprobará que en la base de hechos está el objetivo y parará el sistema.

Archivo: **Salida1-F2.txt**

–DOMINIO DE LA APLICACION–

→IDENTIFICACION DE FRUTAS

–OBJETIVO–

→fruta

BASE DE HECHOS INICIAL:

[color = verde]

[diametro = 8]

[nsemillas = 10]

[forma = redonda]

–FASE DE EQUIPARACION–CONJUNTO CONFLICTO–

SE INSERTA EN EL CONJUNTO CONFLICTO:

–>R1: SI nsemillas > 1 ENTONCES tiposemilla = multiple; Peso: 0

SE INSERTA EN EL CONJUNTO CONFLICTO:

–>R5: SI forma = redonda Y diametro < 10
ENTONCES clasefrutal = arbol; Peso: 0

–FIN FASE EQUIPARACION–CONJUNTO CONFLICTO–

–FASE RESOLVER–

–Regla disparada–

–>R1: SI nsemillas > 1 ENTONCES tiposemilla = multiple; Peso: 0

–FIN FASE RESOLVER–

BASE DE HECHOS ACTUALIZADA:

[tiposemilla = multiple]
[color = verde]
[diametro = 8]
[nsemillas = 10]
[forma = redonda]

–FASE DE EQUIPARACION–CONJUNTO CONFLICTO–

–>Ninguna regla insertada en el conjunto conflicto

–FIN FASE EQUIPARACION–CONJUNTO CONFLICTO–

–FASE RESOLVER–

–Regla disparada–

–>R5: SI forma = redonda Y diametro < 10
ENTONCES clasefrutal = arbol; Peso: 0

–FIN FASE RESOLVER–

BASE DE HECHOS ACTUALIZADA:

[tiposemilla = multiple]
[color = verde]
[diametro = 8]
[clasefrutal = arbol]
[nsemillas = 10]
[forma = redonda]

–FASE DE EQUIPARACION–CONJUNTO CONFLICTO–

SE INSERTA EN EL CONJUNTO CONFLICTO:

–>R16: SI clasefrutal = arbol Y color = verde Y tiposemilla = multiple
ENTONCES fruta = manzana; Peso: 10

–FIN FASE EQUIPARACION–CONJUNTO CONFLICTO–

–FASE RESOLVER–

–Regla disparada–

→R16: SI clasefrutal = arbol Y color = verde Y tiposemilla = multiple
ENTONCES fruta = manzana; Peso: 10

–FIN FASE RESOLVER–

BASE DE HECHOS ACTUALIZADA:

[forma = redonda]
[fruta = manzana]
[nsemillas = 10]
[clasefrutal = arbol]
[diametro = 8]
[color = verde]
[tiposemilla = multiple]

Se ha encontrado solucion tras aplicar las siguientes reglas:

→R1: SI nsemillas > 1 ENTONCES tiposemilla = multiple; Peso: 0

→R5: SI forma = redonda Y diametro < 10
ENTONCES clasefrutal = arbol; Peso: 0

→R16: SI clasefrutal = arbol Y color = verde Y tiposemilla = multiple
ENTONCES fruta = manzana; Peso: 10

Solucion inferida: fruta = manzana

Archivo: **Salida2-F2.txt**

–DOMINIO DE LA APLICACION–

→IDENTIFICACION DE FRUTAS

–OBJETIVO–

→fruta

Se ha encontrado solucion tras aplicar las siguientes reglas:

→R1: SI nsemillas > 1 ENTONCES tiposemilla = multiple; Peso: 0

→R5: SI forma = redonda Y diametro < 10
ENTONCES clasefrutal = arbol; Peso: 0

→R16: SI clasefrutal = arbol Y color = verde Y tiposemilla = multiple
ENTONCES fruta = manzana; Peso: 10

Solucion inferida: fruta = manzana

En cuanto al al razonamiento seguido en la base de hechos *BH-F3.txt*, empieza de nuevo inicializando la base de hechos, para posteriormente añadir dos reglas en la etapa de equiparación-conjunto conflicto (R1 y R3) al verificarse que se cumplen los antecedentes de dichas reglas al comprobarlos con la base de hechos. El conflicto generado se resolverá mediante el criterio estático de desempate de elegir la regla con menor identificador; disparándose R1 y actualizando la base de hecho añadiendo la conclusión de la regla R1.

Posteriormente, no encuentra ninguna regla nueva que añadir al conjunto conflicto y dispara la regla R3 al ser la única restante en el conjunto, actualizando la memoria de trabajo con que la clase del frutal es emparado. Con esto, llega el momento en que en la fase de equiparación se añada al conjunto conflicto R8, regla la cual se dispara dando lugar a que se actualize la base de hechos con la conclusión de que la fruta es una sandía. Tras comprobar que en la base de hechos está el objetivo contenido, el algoritmo parará.

```
-DOMINIO DE LA APLICACION-
->IDENTIFICACION DE FRUTAS

-OBJETIVO-
->fruta

BASE DE HECHOS INICIAL:
    [ color = verde ]
    [ diametro = 11 ]
    [ forma = redonda ]
    [ nsemillas = 2 ]

-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-

    SE INSERTA EN EL CONJUNTO CONFLICTO:
    ->R1: SI nsemillas > 1 ENTONCES tiposemilla = multiple; Peso: 0

    SE INSERTA EN EL CONJUNTO CONFLICTO:
    ->R3: SI forma = redonda Y diametro >= 10
          ENTONCES clasefrutal = emparrado; Peso: 0

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
    ->R1: SI nsemillas > 1 ENTONCES tiposemilla = multiple; Peso: 0

-FIN FASE RESOLVER-

BASE DE HECHOS ACTUALIZADA:
    [ tiposemilla = multiple ]
    [ color = verde ]
    [ diametro = 11 ]
    [ forma = redonda ]
    [ nsemillas = 2 ]

-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-

    ->Ninguna regla insertada en el conjunto conflicto

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
    ->R3: SI forma = redonda Y diametro >= 10
          ENTONCES clasefrutal = emparrado; Peso: 0

-FIN FASE RESOLVER-
```

BASE DE HECHOS ACTUALIZADA:

[tiposemilla = multiple]
[color = verde]
[diametro = 11]
[clasefrutal = emparrado]
[forma = redonda]
[nsemillas = 2]

~~–FASE DE EQUIPARACION–CONJUNTO CONFLICTO–~~

SE INSERTA EN EL CONJUNTO CONFLICTO:

→R8: SI clasefrutal = emparrado Y color = verde
ENTONCES fruta = sandia; Peso: 10

~~–FIN FASE EQUIPARACION–CONJUNTO CONFLICTO–~~

~~–FASE RESOLVER–~~

~~–Regla disparada–~~

→R8: SI clasefrutal = emparrado Y color = verde
ENTONCES fruta = sandia; Peso: 10

~~–FIN FASE RESOLVER–~~

BASE DE HECHOS ACTUALIZADA:

[fruta = sandia]
[nsemillas = 2]
[forma = redonda]
[clasefrutal = emparrado]
[diametro = 11]
[color = verde]
[tiposemilla = multiple]

Se ha encontrado solucion tras aplicar las siguientes reglas:

→R1: SI nsemillas > 1 ENTONCES tiposemilla = multiple; Peso: 0
→R3: SI forma = redonda Y diametro >= 10
ENTONCES clasefrutal = emparrado; Peso: 0
→R8: SI clasefrutal = emparrado Y color = verde
ENTONCES fruta = sandia; Peso: 10

Solucion inferida: fruta = sandia

Archivo: **Salida2-F3.txt**

~~–DOMINIO DE LA APLICACION–~~

→IDENTIFICACION DE FRUTAS

~~–OBJETIVO–~~

→fruta

Se ha encontrado solucion tras aplicar las siguientes reglas:

→R1: SI nsemillas > 1 ENTONCES tiposemilla = multiple; Peso: 0
→R3: SI forma = redonda Y diametro >= 10
ENTONCES clasefrutal = emparrado; Peso: 0
→R8: SI clasefrutal = emparrado Y color = verde
ENTONCES fruta = sandia; Peso: 10

Solucion inferida: fruta = sandia

En cuanto al al razonamiento seguido en la base de hechos *BH-F4.txt*, empieza de nuevo inicializando la base de hechos, para posteriormente añadir dos reglas en la etapa de equiparación-conjunto conflicto (R5 y R6) al verificarse que se cumplen los antecedentes de dichas reglas al comprobarlos con la base de hechos. A continuación, el conflicto generado se resolverá mediante el criterio estático de desempate de elegir la regla con menor identificador; disparándose R5 y actualizando la base de hecho añadiendo la conclusión de la regla R5 de que la clase del frutal es un árbol.

Posteriormente, no encontrará ninguna regla más para añadir al conjunto conflicto, dando lugar a que se dispare la única restante del conjunto (R6) actualizando la base de hecho con el hecho inferido de que el tipo de semilla es hueso. Después de esto, se vuelve a ejecutar la fase de equiparación-conjunto conflicto, añadiendo dos reglas cuyas conclusiones llegan al objetivo. Este conflicto se resolverá aplicando el criterio de menor identificador ya que ambas tienen el mismo peso (10), disparándose R11 llegando a inferir que la fruta es un albaricoque. Archivo: **Salida1-F4.txt**

```
—DOMINIO DE LA APLICACION—
    —>IDENTIFICACION DE FRUTAS

—OBJETIVO—
    —>fruta

BASE DE HECHOS INICIAL:
    [ diametro = 6 ]
    [ nsemillas = 1 ]
    [ forma = redonda ]
    [ clasefrutal = arbol ]
    [ color = naranja ]

—FASE DE EQUIPARACION—CONJUNTO CONFLICTO—

    SE INSERTA EN EL CONJUNTO CONFLICTO:
    —>R5: SI forma = redonda Y diametro < 10
           ENTONCES clasefrutal = arbol; Peso: 0

    SE INSERTA EN EL CONJUNTO CONFLICTO:
    —>R6: SI nsemillas = 1
           ENTONCES tiposemilla = hueso; Peso: 0

—FIN FASE EQUIPARACION—CONJUNTO CONFLICTO—

—FASE RESOLVER—

—Regla disparada—
    —>R5: SI forma = redonda Y diametro < 10
           ENTONCES clasefrutal = arbol; Peso: 0

—FIN FASE RESOLVER—

BASE DE HECHOS ACTUALIZADA:
    [ diametro = 6 ]
    [ nsemillas = 1 ]
    [ forma = redonda ]
```

```

[clasefrutal = arbol]
[color = naranja]

-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-

->Ninguna regla insertada en el conjunto conflicto

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
->R6: SI nsemillas = 1 ENTONCES tiposemilla = hueso; Peso: 0

-FIN FASE RESOLVER-

BASE DE HECHOS ACTUALIZADA:
[tiposemilla = hueso]
[diametro = 6]
[nsemillas = 1]
[forma = redonda]
[clasefrutal = arbol]
[color = naranja]

-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-

SE INSERTA EN EL CONJUNTO CONFLICTO:
->R11: SI clasefrutal = arbol Y color = naranja Y tiposemilla = hueso
      ENTONCES fruta = albaricoque; Peso: 10

SE INSERTA EN EL CONJUNTO CONFLICTO:
->R17: SI clasefrutal = arbol Y color = naranja Y tiposemilla = hueso
      ENTONCES fruta = melocoton; Peso: 10

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
->R11: SI clasefrutal = arbol Y color = naranja Y tiposemilla = hueso
      ENTONCES fruta = albaricoque; Peso: 10

-FIN FASE RESOLVER-

BASE DE HECHOS ACTUALIZADA:
[ color = naranja]
[clasefrutal = arbol]
[forma = redonda]
[fruta = albaricoque]
[nsemillas = 1]
[diametro = 6]
[tiposemilla = hueso]

Se ha encontrado solucion tras aplicar las siguientes reglas:

```

```
->R5: SI forma = redonda Y diametro < 10
      ENTONCES clasefrutal = arbol; Peso: 0
->R6: SI nsemillas = 1 ENTONCES tiposemilla = hueso; Peso: 0
->R11: SI clasefrutal = arbol Y color = naranja Y tiposemilla = hueso
      ENTONCES fruta = albaricoque; Peso: 10
Solucion inferida: fruta = albaricoque
```

Archivo: **Salida2-F4.txt**

```
-DOMINIO DE LA APLICACION-
->IDENTIFICACION DE FRUTAS

-OBJETIVO-
->fruta

Se ha encontrado solucion tras aplicar las siguientes reglas:
->R5: SI forma = redonda Y diametro < 10
      ENTONCES clasefrutal = arbol; Peso: 0
->R6: SI nsemillas = 1 ENTONCES tiposemilla = hueso; Peso: 0
->R11: SI clasefrutal = arbol Y color = naranja Y tiposemilla = hueso
      ENTONCES fruta = albaricoque; Peso: 10

Solucion inferida: fruta = albaricoque
```

3.2. Situación 2: Detección de Inundaciones

Para realizar la detección de inundaciones con el sistema basado en reglas desarrollado, se ha tenido que crear un fichero de configuración y cuatro ficheros de base de hechos. Para comenzar a desarrollar el fichero, he definido los atributos y sus dominios fijandome en la base de conocimientos proporcionada, además, se puede observar que el objetivo en el que intentamos concluir es *inundación*

En segundo lugar, a la hora de diseñar las prioridades de las reglas me he fijado en las reglas de la base de conocimientos proporcionada separando estas en dos conjuntos; las que llegan al objetivo y las que infieren antecedentes necesarios para llegar al objetivo. En cuanto al conjunto de reglas que llegan al objetivo, he asignado la prioridad teniendo en cuenta que una regla con más restricciones tiende a ser más fiable que una con menos, aunque no siempre tenga que ser así. Por tanto, una regla con más antecedentes que otra tendrá mas prioridad en caso de que ambas lleguen a inferir el objetivo.

Sin embargo, el conjunto de reglas que infieren conclusiones necesarias para llegar al objetivo no tendrán el mismo criterio de asignación de prioridades ya que nos interesan más las reglas cuya conclusión forma parte de un menor número de antecedentes en reglas que concluyen en el objetivo. Como pueden ser en primer lugar: **Lluvia** al ser la menos común en los antecedentes del otro conjunto. Después, **Cambio** y por último **Nivel**. Asignando a las reglas que concluyen en esto, de mayor a menor prioridad. Hacemos esto para mejorar la fiabilidad de la solución inferida pese a empeorar la convergencia en una solución.

Archivo: **Config-I.txt**

ATRIBUTOS

9

Mes Nom {Enero , Febrero , Marzo , Abril , Mayo , Junio ,
Julio , Agosto , Septiembre , Octubre , Noviembre , Diciembre }

Estacion Nom {Seca , Humeda }

Precipitaciones Nom {Ninguna , Ligera , Fuertes }

Cambio Nom {Ninguna , Bajando , Subiendo }

Profundidad NU

Nivel Nom {Bajo , Normal , Alto }

Prediccion Nom {Soleado , Nuboso , Tormenta }

Lluvia Nom {Ninguna , Ligera , Fuerte }

Inundacion Nom {Si , No }

OBJETIVO

Inundacion

PRIORIDADES-REGLAS

32

0

0

0

0

0

0

0

0

0

0

0

0

2

2

2

2

1

1

1

3

3
3
8
9
9
9
9
10
10
10
10
10

Archivo: **Salida1-I1.txt**

```
-DOMINIO DE LA APLICACION-  
  ->DETECCION DE INUNDACIONES  
  
-OBJETIVO-  
  ->inundacion  
  
BASE DE HECHOS INICIAL:  
  [precipitaciones = ninguna]  
  [profundidad = 1]  
  [mes = julio]  
  
-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-  
  
  SE INSERTA EN EL CONJUNTO CONFLICTO:  
  ->R2: SI mes = julio ENTONCES estacion = seca; Peso: 0  
  
  SE INSERTA EN EL CONJUNTO CONFLICTO:  
  ->R17: SI profundidad < 3 ENTONCES nivel = bajo; Peso: 1  
  
-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-  
  
-FASE RESOLVER-  
  
-Regla disparada-  
  ->R17: SI profundidad < 3 ENTONCES nivel = bajo; Peso: 1  
  
-FIN FASE RESOLVER-  
  
BASE DE HECHOS ACTUALIZADA:  
  [precipitaciones = ninguna]  
  [profundidad = 1]  
  [nivel = bajo]  
  [mes = julio]  
  
-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-  
  
  SE INSERTA EN EL CONJUNTO CONFLICTO:  
  ->R23: SI nivel = bajo ENTONCES inundacion = no; Peso: 8
```

```

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
->R23: SI nivel = bajo ENTONCES inundacion = no; Peso: 8

-FIN FASE RESOLVER-

BASE DE HECHOS ACTUALIZADA:
    [precipitaciones = ninguna]
    [profundidad = 1]
    [inundacion = no]
    [nivel = bajo]
    [mes = julio]

Se ha encontrado solucion tras aplicar las siguientes reglas:
->R17: SI profundidad < 3 ENTONCES nivel = bajo; Peso: 1
->R23: SI nivel = bajo ENTONCES inundacion = no; Peso: 8
Solucion inferida: inundacion = no

```

Archivo: **Salida2-I1.txt**

```

-DOMINIO DE LA APLICACION-
->DETECCION DE INUNDACIONES

-OBJETIVO-
->inundacion

Se ha encontrado solucion tras aplicar las siguientes reglas:
->R17: SI profundidad < 3 ENTONCES nivel = bajo; Peso: 1
->R23: SI nivel = bajo ENTONCES inundacion = no; Peso: 8
Solucion inferida: inundacion = no

```

Archivo: **Salida1-I2.txt**

```

-DOMINIO DE LA APLICACION-
->DETECCION DE INUNDACIONES

-OBJETIVO-
->inundacion

BASE DE HECHOS INICIAL:
    [prediccion = tormenta]
    [precipitaciones = fuertes]
    [mes = mayo]
    [profundidad = 5]

-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-

    SE INSERTA EN EL CONJUNTO CONFLICTO:
->R12: SI mes = mayo ENTONCES estacion = humeda; Peso: 0

    SE INSERTA EN EL CONJUNTO CONFLICTO:

```

```

->R16: SI precipitaciones = fuertes ENTONCES cambio = subiendo; Peso: 2

SE INSERTA EN EL CONJUNTO CONFLICTO:
->R18: SI profundidad >= 3 Y profundidad <= 5 ENTONCES nivel = normal; Peso: 1

SE INSERTA EN EL CONJUNTO CONFLICTO:
->R22: SI prediccion = tormenta ENTONCES lluvia = fuerte; Peso: 3

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
->R22: SI prediccion = tormenta ENTONCES lluvia = fuerte; Peso: 3

-FIN FASE RESOLVER-

BASE DE HECHOS ACTUALIZADA:
    [lluvia = fuerte]
    [prediccion = tormenta]
    [precipitaciones = fuertes]
    [mes = mayo]
    [profundidad = 5]

-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-

->Ninguna regla insertada en el conjunto conflicto

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
->R16: SI precipitaciones = fuertes ENTONCES cambio = subiendo; Peso: 2

-FIN FASE RESOLVER-

BASE DE HECHOS ACTUALIZADA:
    [cambio = subiendo]
    [lluvia = fuerte]
    [prediccion = tormenta]
    [precipitaciones = fuertes]
    [mes = mayo]
    [profundidad = 5]

-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-

->Ninguna regla insertada en el conjunto conflicto

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
->R18: SI profundidad >= 3 Y profundidad <= 5 ENTONCES nivel = normal; Peso: 1

```

–FIN FASE RESOLVER–

BASE DE HECHOS ACTUALIZADA:

[nivel = normal]
[profundidad = 5]
[mes = mayo]
[precipitaciones = fuertes]
[prediccion = tormenta]
[lluvia = fuerte]
[cambio = subiendo]

–FASE DE EQUIPARACION–CONJUNTO CONFLICTO–

SE INSERTA EN EL CONJUNTO CONFLICTO:

–>R26: SI cambio = subiendo Y nivel = normal
ENTONCES inundacion = no; Peso: 9

SE INSERTA EN EL CONJUNTO CONFLICTO:

–>R28: SI cambio = subiendo Y nivel = normal Y lluvia = fuerte
ENTONCES inundacion = si; Peso: 10

–FIN FASE EQUIPARACION–CONJUNTO CONFLICTO–

–FASE RESOLVER–

–Regla disparada–

–>R28: SI cambio = subiendo Y nivel = normal Y lluvia = fuerte
ENTONCES inundacion = si; Peso: 10

–FIN FASE RESOLVER–

BASE DE HECHOS ACTUALIZADA:

[inundacion = si]
[nivel = normal]
[profundidad = 5]
[mes = mayo]
[precipitaciones = fuertes]
[prediccion = tormenta]
[lluvia = fuerte]
[cambio = subiendo]

Se ha encontrado solucion tras aplicar las siguientes reglas:

–>R22: SI prediccion = tormenta ENTONCES lluvia = fuerte; Peso: 3

–>R16: SI precipitaciones = fuertes ENTONCES cambio = subiendo; Peso: 2

–>R18: SI profundidad >= 3 Y profundidad <= 5

ENTONCES nivel = normal; Peso: 1

–>R28: SI cambio = subiendo Y nivel = normal Y lluvia = fuerte
inundacion = si; Peso: 10

Solucion inferida: inundacion = si

Archivo: **Salida2-I2.txt**

–DOMINIO DE LA APLICACION–

–>DETECCION DE INUNDACIONES

–OBJETIVO–

->inundacion

Se ha encontrado solucion tras aplicar las siguientes reglas:

->R22: SI prediccion = tormenta ENTONCES lluvia = fuerte; Peso: 3

->R16: SI precipitaciones = fuertes ENTONCES cambio = subiendo; Peso: 2

->R18: SI profundidad >= 3 Y profundidad <= 5

ENTONCES nivel = normal; Peso: 1

->R28: SI cambio = subiendo Y nivel = normal Y lluvia = fuerte

ENTONCES inundacion = si; Peso: 10

Solucion inferida: inundacion = si

Archivo: **Salida1-I3.txt**

~~—DOMINIO DE LA APLICACION—~~

~~—>DETECCION DE INUNDACIONES~~

~~—OBJETIVO—~~

~~—>inundacion~~

BASE DE HECHOS INICIAL:

[nivel = alto]

[mes = enero]

[precipitaciones = fuertes]

[prediccion = tormenta]

~~—FASE DE EQUIPARACION—CONJUNTO CONFLICTO—~~

SE INSERTA EN EL CONJUNTO CONFLICTO:

->R8: SI mes = enero ENTONCES estacion = humeda; Peso: 0

SE INSERTA EN EL CONJUNTO CONFLICTO:

->R16: SI precipitaciones = fuertes ENTONCES cambio = subiendo; Peso: 2

SE INSERTA EN EL CONJUNTO CONFLICTO:

->R22: SI prediccion = tormenta ENTONCES lluvia = fuerte; Peso: 3

~~—FIN FASE EQUIPARACION—CONJUNTO CONFLICTO—~~

~~—FASE RESOLVER—~~

~~—Regla disparada—~~

->R22: SI prediccion = tormenta ENTONCES lluvia = fuerte; Peso: 3

~~—FIN FASE RESOLVER—~~

BASE DE HECHOS ACTUALIZADA:

[lluvia = fuerte]

[nivel = alto]

[mes = enero]

[precipitaciones = fuertes]

[prediccion = tormenta]

~~—FASE DE EQUIPARACION—CONJUNTO CONFLICTO—~~

```

->Ninguna regla insertada en el conjunto conflicto

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
->R16: SI precipitaciones = fuertes ENTONCES cambio = subiendo; Peso: 2

-FIN FASE RESOLVER-

BASE DE HECHOS ACTUALIZADA:
    [cambio = subiendo]
    [lluvia = fuerte]
    [nivel = alto]
    [mes = enero]
    [precipitaciones = fuertes]
    [prediccion = tormenta]

-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-

    SE INSERTA EN EL CONJUNTO CONFLICTO:
    ->R32: SI cambio = subiendo Y nivel = alto Y lluvia = fuerte
          ENTONCES inundacion = si; Peso: 10

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
->R32: SI cambio = subiendo Y nivel = alto Y lluvia = fuerte
          ENTONCES inundacion = si; Peso: 10

-FIN FASE RESOLVER-

BASE DE HECHOS ACTUALIZADA:
    [inundacion = si]
    [precipitaciones = fuertes]
    [mes = enero]
    [nivel = alto]
    [prediccion = tormenta]
    [lluvia = fuerte]
    [cambio = subiendo]

Se ha encontrado solucion tras aplicar las siguientes reglas:
->R22: SI prediccion = tormenta ENTONCES lluvia = fuerte; Peso: 3
->R16: SI precipitaciones = fuertes ENTONCES cambio = subiendo; Peso: 2
->R32: SI cambio = subiendo Y nivel = alto Y lluvia = fuerte
          ENTONCES inundacion = si; Peso: 10

Solucion inferida: inundacion = si

```

~~–DOMINIO DE LA APLICACION–~~
~~–>DETECCION DE INUNDACIONES~~

~~–OBJETIVO–~~
~~–>inundacion~~

Se ha encontrado solucion tras aplicar las siguientes reglas:

~~–>R22: SI prediccion = tormenta ENTONCES lluvia = fuerte; Peso: 3~~
~~–>R16: SI precipitaciones = fuertes ENTONCES cambio = subiendo; Peso: 2~~
~~–>R32: SI cambio = subiendo Y nivel = alto Y lluvia = fuerte~~
~~ENTONCES inundacion = si; Peso: 10~~

Solucion inferida: inundacion = si

Archivo: **Salida1-I4.txt**

~~–DOMINIO DE LA APLICACION–~~
~~–>DETECCION DE INUNDACIONES~~

~~–OBJETIVO–~~
~~–>inundacion~~

BASE DE HECHOS INICIAL:

[profundidad = 7]
[prediccion = tormenta]
[precipitaciones = fuertes]

~~–FASE DE EQUIPARACION–CONJUNTO CONFLICTO–~~

SE INSERTA EN EL CONJUNTO CONFLICTO:

~~–>R16: SI precipitaciones = fuertes ENTONCES cambio = subiendo; Peso: 2~~

SE INSERTA EN EL CONJUNTO CONFLICTO:

~~–>R19: SI profundidad > 5 ENTONCES nivel = alto; Peso: 1~~

SE INSERTA EN EL CONJUNTO CONFLICTO:

~~–>R22: SI prediccion = tormenta ENTONCES lluvia = fuerte; Peso: 3~~

~~–FIN FASE EQUIPARACION–CONJUNTO CONFLICTO–~~

~~–FASE RESOLVER–~~

~~–Regla disparada–~~

~~–>R22: SI prediccion = tormenta ENTONCES lluvia = fuerte; Peso: 3~~

~~–FIN FASE RESOLVER–~~

BASE DE HECHOS ACTUALIZADA:

[lluvia = fuerte]
[profundidad = 7]
[prediccion = tormenta]
[precipitaciones = fuertes]

```

-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-

    ->Ninguna regla insertada en el conjunto conflicto

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
    ->R16: SI precipitaciones = fuertes ENTONCES cambio = subiendo; Peso: 2

-FIN FASE RESOLVER-

BASE DE HECHOS ACTUALIZADA:
    [cambio = subiendo]
    [lluvia = fuerte]
    [profundidad = 7]
    [prediccion = tormenta]
    [precipitaciones = fuertes]

-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-

    ->Ninguna regla insertada en el conjunto conflicto

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
    ->R19: SI profundidad > 5 ENTONCES nivel = alto; Peso: 1

-FIN FASE RESOLVER-

BASE DE HECHOS ACTUALIZADA:
    [nivel = alto]
    [cambio = subiendo]
    [lluvia = fuerte]
    [profundidad = 7]
    [prediccion = tormenta]
    [precipitaciones = fuertes]

-FASE DE EQUIPARACION-CONJUNTO CONFLICTO-

    SE INSERTA EN EL CONJUNTO CONFLICTO:
    ->R32: SI cambio = subiendo Y nivel = alto Y lluvia = fuerte
           ENTONCES inundacion = si; Peso: 10

-FIN FASE EQUIPARACION-CONJUNTO CONFLICTO-

-FASE RESOLVER-

-Regla disparada-
    ->R32: SI cambio = subiendo Y nivel = alto Y lluvia = fuerte
           ENTONCES inundacion = si; Peso: 10

```


–FIN FASE RESOLVER–

BASE DE HECHOS ACTUALIZADA:

```
[inundacion = si]
[precipitaciones = fuertes]
[profundidad = 7]
[prediccion = tormenta]
[lluvia = fuerte]
[cambio = subiendo]
[nivel = alto]
```

Se ha encontrado solucion tras aplicar las siguientes reglas:

```
–>R22: SI prediccion = tormenta ENTONCES lluvia = fuerte; Peso: 3
–>R16: SI precipitaciones = fuertes ENTONCES cambio = subiendo; Peso: 2
–>R19: SI profundidad > 5 ENTONCES nivel = alto; Peso: 1
–>R32: SI cambio = subiendo Y nivel = alto Y lluvia = fuerte
      ENTONCES inundacion = si; Peso: 10
```

Solucion inferida: inundacion = si

Archivo: **Salida2-I4.txt**

–DOMINIO DE LA APLICACION–

–>DETECCION DE INUNDACIONES

–OBJETIVO–

–>inundacion

Se ha encontrado solucion tras aplicar las siguientes reglas:

```
–>R22: SI prediccion = tormenta ENTONCES lluvia = fuerte; Peso: 3
–>R16: SI precipitaciones = fuertes ENTONCES cambio = subiendo; Peso: 2
–>R19: SI profundidad > 5 ENTONCES nivel = alto; Peso: 1
–>R32: SI cambio = subiendo Y nivel = alto Y lluvia = fuerte
      ENTONCES inundacion = si; Peso: 10
```

Solucion inferida: inundacion = si

Referencias

- [1] WIKIPEDIA, *Forward-Chaining*, Definición de encaminamiento hacia delante.
Available: https://en.wikipedia.org/wiki/Forward_chaining
- [2] WIKIPEDIA, *Backward-Chaining*, Definición de encaminamiento hacia atrás.
Available: https://en.wikipedia.org/wiki/Backward_chaining
- [3] CPPREFERENCE, *std::hash*, Para la utilización de conjuntos y mapas en C++ en clases definidas por usuarios. Available: <https://en.cppreference.com/w/cpp/utility/hash>
- [4] MATA PÉREZ, MIGUEL, *Bibliografía en LATEX*, Utilizado para el desarrollo de referencias en LATEX.
Available: <http://logistica.fime.uanl.mx/miguel/docs/BibTeX.pdf>.
- [5] PROFESORES DE LA ASIGNATURA DE SISTEMAS INTELIGENTES DE LA UNIVERSIDAD DE MURCIA, *Apuntes de la asignatura de sistemas inteligentes*, Utilizado para el desarrollo del sistema basado en reglas.