



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

DSiC
DEPARTAMENTO DE SISTEMAS
INFORMÁTICOS Y COMPUTACIÓN

Escuela Técnica Superior de Ingeniería Informática



**Departamento de Sistemas Informáticos y Computación
Escuela Técnica Superior de Ingeniería Informática
Universitat Politècnica de València**

BOLETÍN DE EJERCICIOS **SISTEMAS INTELIGENTES**

Bloque 1: Representación del Conocimiento

Septiembre 2024

CUESTIONES

1) Dado un SBR, con una única regla:

```
(defrule regla-1
  ?f <- (lista ?x $?y ?x $?y ?x)
=>
  (retract ?f)
  (assert (lista $?y)))
```

, y la BH inicial {(lista a b c b c b c b c b a b c b c b c b a)}, ¿cuál será el estado final de la Base de Hechos?

- A. {(lista a)}
 - B. {(lista a) (lista)}
 - C. {(lista b a b)}
 - D. {(lista c b c)}
-

2) En un determinado SBR para resolver un problema, la declaración de una BH inicial representa:

- A. Los datos estáticos del problema.
 - B. Los datos dinámicos del problema, en su estado inicial.
 - C. Los datos estáticos y dinámicos, en su estado inicial.
 - D. Ninguna de las anteriores es cierta. La BH representa los hechos y el conjunto de reglas del dominio.
-

3) Supongamos que tenemos en una mesa varias cajas de distintos tamaños que queremos apilar en una torre, de mayor a menor tamaño, mediante acciones de 'apilado'. La regla meta aclara el objetivo deseado. Definimos un SBR donde la BH inicial se describe de la siguiente forma:

```
(defacts prueba (mesa 2 5 1 6 8 7 4 torre ))
```

, y se definen las siguientes reglas:

```
(defrule mesa-a-torre
  (mesa $?rest1 ?x $?rest3 torre $?rest1 ?y )
  (test (> ?y ?x))
=>
  (assert (mesa $?rest1 $?rest3 torre $?rest1 ?y ?x )))

(defrule mesa-a-torre-vacia
  (mesa $?rest1 ?x $?rest3 torre )
=>
  (assert (mesa $?rest1 $?rest3 torre ?x )))
```

```
(defrule meta
  (mesa torre 1 2 4 5 6 7 8 )
=>
  (halt))
```

¿Cuál de las siguientes afirmaciones es correcta?

- A. El SBR definido funciona correctamente y obtendrá en la torre las cajas ordenadas por tamaño.
 - B. El SBR definido funciona correctamente, obtendrá en la torre las cajas ordenadas por tamaño, y la regla meta no es necesaria ya que el proceso acabará al quedar la mesa vacía.
 - C. El SBR solo funcionaría si las cajas en la mesa están ordenadas de menor a mayor, por ejemplo: (defacts prueba (mesa 1 2 4 5 6 7 8 torre)).
 - D. Ninguna de las afirmaciones anteriores es cierta.
-

4) Sea el siguiente SBR para calcular el factorial de un número, ¿cuál de las siguientes afirmaciones es correcta?

```
(defacts factorial (numero 5) (fact 1))
```

```
(defrule fact
  ?f1 <- (numero ?n1)
  ?f2 <- (fact ?n2)
=>
  (retract ?f1 ?f2)
  (assert (numero (- ?n1 1)))
  (assert (fact (* ?n2 ?n1 ))))
```

- A. El SBR funciona correctamente.
 - B. Lo anterior no es cierto, pues se requiere una condición (test (> ?n1 1)) en la premisa de la regla para que funcione correctamente.
 - C. La modificación anterior no bastaría, siendo además necesario una regla de parada: (defrule parada (declare (salience 100)) (numero 1) => (halt)).
 - D. Ninguna de las anteriores es cierta.
-

5) Sea el hecho:

```
(escuela clase 1 niños 15 niñas 18 clase 2 niños 21 niñas 14 clase 3 niños 16 niñas 17)
```

, donde el número que aparece después del símbolo 'clase' indica el identificador de dicha clase, y los valores numéricos que aparecen después de los símbolos 'niños' o 'niñas' indican el número de

niños o niñas de la clase correspondiente. Indica el patrón adecuado para obtener únicamente el identificador de una clase cualquiera y el número de niñas de dicha clase:

- A. (escuela \$? clase ?c \$? niñas ?na \$?)
 - B. (escuela \$? clase ?c niños ? niñas ?na \$?)
 - C. (escuela \$? clase ? niños \$? niñas ?na \$?)
 - D. (escuela clase ?c niños ? niñas ?na)
-

6) Sea un SBR formado por $BH_{inicial} = \{(lista\ 23\ 14\ 56\ 33)\}$, y las siguientes reglas:

```
(defrule R1
  (declare (salience 100))
  ?f <- (lista $?x ?z ?y $?w)
  (test (< ?z ?y))
=>
  (assert (lista $?x ?z ?y $?w)))

(defrule R2
  (declare (salience 150))
  ?f <- (lista $?x ?z ?y $?w)
  (test (>= ?z ?y))
=>
  (assert (lista $?x ?z ?y $?w)))

(defrule final
  (declare (salience 200))
  (lista $?list)
=>
  (halt))
```

¿Cuál sería el contenido del Conjunto Conflicto (Agenda) tras el primer *pattern-matching*?

- A. Una instancia de la regla R1, una instancia de la regla R2 y una instancia de la regla final
 - B. Dos instancias de la regla R2 y una instancia de la regla R1
 - C. Dos instancias de la regla R2, una instancia de la regla R1 y una instancia de la regla final
 - D. Una instancia de la regla final
-

7) Sea de nuevo el SBR de la pregunta 6. Asumiendo que la estrategia del Conjunto Conflicto (Agenda) es anchura, ¿cuál es la primera instancia de regla que selecciona el motor de inferencia de CLIPS para ser ejecutada? Indica cuál de las siguientes afirmaciones es CIERTA:

- A. Selecciona una instancia de la regla R1.
 - B. Selecciona una instancia de la regla R2.
 - C. Selecciona una instancia de la regla final.
 - D. Ninguna de las anteriores.
-

8) Sea un SBR, con una única regla:

```
(defrule R1
  ?f <- (lista ?x $?y ?x $?z)
=>
```

```
(retract ?f)
(assert (lista $?y ?x $?z))
(printout t "La lista se ha modificado " crlf))
```

, y la BH inicial {(lista a b a b a)}. Tras ejecutar el SBR, ¿cuántas veces se habrá mostrado en pantalla el mensaje "La lista se ha modificado "?

- A. 1
- B. 2
- C. 3
- D. 4

9) Tenemos un montacargas que recoge paquetes de la planta baja y los reparte entre diferentes plantas. Una instancia del problema tiene dos paquetes A y B, con destino 2ª y 3ª planta, peso 2 kg. y 8 kg. respectivamente. El montacargas está en la planta baja y su peso máximo es 40 kg. ¿Cuál de las siguientes representaciones NO serviría para realizar una búsqueda en grafo en una representación basada en estados?

- A. (montacargas planta 0 carga paquete A 2 2 paquete B 3 8 peso-maximo 40 nivel 0)
- B. (montacargas planta 0 carga paquete A 2 2 paquete B 3 8) (peso-maximo 40)
- C. (montacargas planta 0 carga paquete A 2 2 paquete B 3 8 nivel 0) (peso-maximo 40)
- D. (montacargas planta 0) (carga paquete A 2 2 paquete B 3 8 nivel 0) (peso-maximo 40)

10) Dado el siguiente SBR, ¿cuántas reglas se insertarán en la agenda en el primer ciclo de inferencia?

```
(defrule R1
(lista $?x1 ?y $?x2 ?y $?x3)
=>
(assert (lista $?x1 ?y $?x3)))
```

```
(deffacts inicio
(lista 2 3 1 2 3 2 1))
```

- A. 4
- B. 5
- C. Ninguna
- D. 3

11) Dado el siguiente SBR, indica cuál de las siguientes respuestas es CORRECTA:

```
(defrule R1
(declare (salience 100))
```

```

      ?f <- (lista $?x ?y)
      (test (> ?y 5))
=>
      (retract ?f)
      (assert (lista $?x)))

```

```

(defrule R2
  (declare (salience 200))
  ?f <- (lista ?y $?x)
  (test (> ?y 5))
=>
  (retract ?f)
  (assert (lista $?x)))

```

```

(deffacts inicio
  (lista 3 7 1 5 9))

```

- A. Sólo en el caso de que la estrategia de la agenda sea anchura, se ejecutará en primer lugar una instancia de R1
- B. Sólo en el caso de que la estrategia de la agenda sea profundidad, se ejecutará en primer lugar una instancia de R2
- C. Se ejecutará una instancia de R1 en primer lugar en cualquier caso
- D. Se ejecutará una instancia de R2 en primer lugar en cualquier caso

12) Dado el siguiente hecho (pila A B A A B B A pilaA pilaB), que representa el estado inicial de un SBR, donde se tiene una pila inicial con bloques A y B y el objetivo es separar dichos bloques en dos pilas, una con bloques A y otra con bloques B. Indica cuál de las siguientes reglas NO toma un bloque A de la pila inicial y lo mueve a la pila de bloques A, de manera que se pueda resolver el problema:

```

A. (defrule mover-a-pila-A
    (pila $?x A $?y pilaA $?z)
=>
    (assert (pila $?x $?y pilaA A $?z)))

```

```

B. (defrule mover-a-pila-A
    (pila $?x ?b $?y pilaA $?z)
    (test (eq ?b A))
=>
    (assert (pila $?x $?y pilaA A $?z)))

```

```

C. (defrule mover-a-pila-A
    (pila $?x ?b $?y pilaA $?z)
    (test (eq ?b A))
=>
    (assert (pila $?x ?b $?y pilaA ?b $?z)))

```

```

D. (defrule mover-a-pila-A
    (pila $?x ?b $?y pilaA $?z)
    (test (eq ?b A))
=>
    (assert (pila $?x $?y pilaA ?b $?z)))

```

13) En un almacén se tienen dos zonas: una de carga y otra de descarga. En cada zona, puede haber varias pilas (identificadas con valores de 1 a 5) de pallets de tipo A, B o C para ser cargados o descargados en los camiones. Sea el siguiente estado inicial:

(almacen zona carga pila 1 A B C pila 2 B C B pila 3 A zona descarga pila 4 A B A pila 5 B A B B A)

Indica cuál de los siguientes patrones NO se podría utilizar para almacenar únicamente en la variable ?p el identificador de una pila de la zona de carga cuyo primer pallet sea de tipo A:

- A. (almacen zona carga \$?c pila ?p A \$?r zona descarga \$?d)
 - B. (almacen zona carga \$?c pila ?p A \$?r)**
 - C. (almacen \$?c pila ?p A \$?r zona descarga \$?d)
 - D. (almacen \$?c pila ?p A \$?r 4 \$?d)
-

14) Dada la BH inicial= {(elemento e) (lista e a e b c d e f)}, y el siguiente conjunto de reglas:

```

(defrule R1
  ; (declare (salience 10))
  (elemento ?e)
  (lista $?a ?e $?b)
=>
  (assert (lista ?e $?a $?b)))

(defrule R2
  ; (declare (salience -30))
  (lista ?a $?x ?a)
  (elemento ?a )
=>
  (assert (lista $?x)))

```

¿Cuál de las siguientes afirmaciones es CORRECTA? (NOTA: el ; delante de los comandos (declare (salience ...)) indica que están comentados)

- A. El estado final dependerá de la estrategia de control aplicada (anchura, profundidad, coste uniforme, etc.)
- B. En el SBR definido no llega a lanzarse ninguna regla.
- C. Si se hubieran declarado prioridades en las reglas (salience ...), el estado final dependería de dichas prioridades

D. El estado final será el mismo, cualquiera que sea el tipo de control que se aplique

15) Dada la siguiente parte izquierda de una regla

```
(defrule r1
  (lista $? ?x $? ?y)
  (test (< ?x ?y))
=>
  ...
```

, y el siguiente hecho: (lista 1 3 2 1 3 6), ¿Cuántas instancias de esta regla se incluirán en la agenda?:

- A. 0
 - B. 1
 - C. 5
 - D. Más de 5
-

16) Dada la BH={{(lista1 b a a c c a c b b c)(lista2 a c)} y la siguiente regla

```
(defrule r1
  ?f <- (lista1 $?x ?a ?a $?y)
  (lista2 $? ?a $? )
=>
  (retract ?f)
  (assert (lista1 $?x ?a $?y)))
```

Indicad cuál será la Base de Hechos final.

- A. {(lista1 b b b c) (lista2 a c)}
 - B. {(lista1 b a c a c b b c) (lista2 a c)}
 - C. {(lista1 b b b c)}
 - D. {(lista1 b a c a c b b c) (lista2 a c)}
-

17) Dada la siguiente parte izquierda de una regla:

```
(defrule r2
  ?f <- (lista $? ?b $?x ?b $?x)
=>
  ...
```

y el hecho (lista c c d c c d c c d). ¿Cuántas instancias de esta regla se insertarían en la agenda?

- A. 1

- B. 2
- C. 3
- D. 4

18) Dada la base de hechos inicial: BH={{(lista 5 7 3 1 6 4) (maximo 0)}} y la siguiente regla para calcular el máximo de una lista

```
(defrule r4
  ?f1 <- (lista $?a ?b $?c)
  ?f2 <- (maximo ?x)
  (test (> ?b ?x))
=>
  (assert (lista $?a $?c))
  (assert (maximo ?b)))
```

Si nuestro objetivo es obtener una base de hechos final (tras la ejecución sucesiva de la regla) en la cual el hecho 'maximo' solo puede aparecer una vez (conteniendo el valor máximo de la lista). ¿Cuál de las siguientes afirmaciones es CIERTA para obtener nuestro objetivo?

- A. La regla es correcta
- B. Sería necesario añadir (retract ?f1)
- C. Sería necesario añadir (retract ?f2)
- D. La modificación de C no bastaría ya que sería también necesario añadir (retract ?f1)

19) Dado el siguiente hecho: (problema torre a b c nombre A torre a nombre B torre nombre C) ¿Cuál de los siguientes patrones serviría para obtener el nombre de una torre con un único elemento en ella?

- A. (problema \$?x torre ?a \$?y nombre ? \$?z)
- B. (problema \$?x torre ?a nombre ?z \$?x)
- C. (problema \$?x torre ?a nombre ?z \$?)
- D. (problema \$? torre ?a nombre ?)

20) Sea el siguiente SBR para calcular el número de Fibonacci de un número $n > 0$, por ejemplo $n=5$ (el número de Fibonacci se calcula en fib-1), ¿cuál de las siguientes afirmaciones es correcta?

```
(defrule fact
  ?f1 <- (numero ?n1)
  ?f2 <- (fib-1 ?n2)
  ?f3 <- (fib-2 ?n3)
=>
  (retract ?f1 ?f2 ?f3)
  (assert (numero (- ?n1 1))))
```

```
(assert (fib-1 (+?n2 ?n3)))  
(assert (fib-2 ?n2 )))
```

```
(defacts fibonacci (numero 5) (fib-1 1) (fib-2 0))
```

NOTA: El número de Fibonnaci se calcula como: $f(n)=f(n-1)+f(n-2)$ siendo $f(0)=0$ y $f(1)=1$.

- A. El SBR funciona correctamente.
- B. Lo anterior no es cierto, pues se requiere una condición (test (> ?n1 1)) en la premisa de la regla para que funcione correctamente.
- C. La modificación anterior no bastaría, siendo además necesario una regla de parada: (defrule parada (declare (salience 100)) (numero 1) => (halt)).
- D. Ninguna de las anteriores es cierta.

21) Sea el hecho:

(Mercado Calle 1 Fruta 20 Pescado 0 Calle 2 Fruta 10 Pescado 10 Calle 3 Fruta 16 Pescado 4)

donde el número que aparece después del símbolo 'Calle' indica el identificador de dicha calle en el Mercado, los números que aparecen después de los símbolos 'Fruta' o 'Pescado' indican el número de puestos de fruta o puestos de pescado en la calle correspondiente. ¿Cuál será el patrón adecuado para obtener únicamente el identificador de una calle cualquiera y el número de puestos de pescado de dicha calle?

- A. (Mercado \$? Calle ? Fruta \$? Pescado ?n \$?)
- B. (Mercado \$? Calle ?c \$? Pescado ?n \$?)
- C. (Mercado Calle ?c Fruta ? Pescado ?n)
- D. (Mercado \$? Calle ?c Fruta ? Pescado ?n \$?)

22) Sea un SBR formado por $BH_{inicial}=\{(lista\ 2\ 1\ 6\ 2\ 3)\}$, y las siguientes reglas:

```
(defrule R1  
  ?f <- (lista $?x ?z ?y $?w)  
  (test (< ?z ?y))  
=>  
(assert (lista $?x ?z ?y $?w)))
```

```
(defrule R2  
  ?f <- (lista $?x ?z ?y $?w)  
  (test (> ?z ?y))  
=>  
(assert (lista $?x ?z ?y $?w)))
```

¿Cuál sería el contenido del Conjunto Conflicto (Agenda) tras el primer pattern-matching?

- A. Una instancia de la regla R1 y dos de la R2
 - B. Dos instancias de la regla R1 y una instancia de la regla R2
 - C. Dos instancias de la regla R1 y dos instancias de la regla R2
 - D. Ninguna instancia
-

23) Dado un SBR compuesto de la siguiente regla:

```
(defrule regla-1
  ?f <- (lista ?y $?x ?y $?x ?y)
=>
  (retract ?f)
  (assert (lista $?x)))
```

, y la BH inicial {(lista 1 2 3 2 3 2 3 2 3 2 1 2 3 2 3 2 3 2 3 2 1)}, ¿cuál será el estado final de la Base de Hechos?

- A. {(lista 3 2 3)}
- B. {(lista 1) (lista)}
- C. {(lista 2 1 2)}
- D. {(lista 1)}

24) Sea el formato de patrón (lista [nombre^s edad^s]^m) para representar el nombre y la edad de un conjunto de personas. Dada una lista determinada de personas, se quiere contar el número de ellas cuya edad está comprendida entre 18 y 65 años. Para ello se dispone del hecho que representa la lista de personas, un hecho inicial (contador 0) para contar el número de personas y la regla que se muestra a continuación. Indica la opción CORRECTA:

```
(defrule contar
  ?f1 <-(lista $?x1 ?num $?x2)
  ?f2 <- (contador ?cont)
  (test (numberp ?num)) ;; numberp devuelve TRUE si ?num es un número
  (test (and (>= ?num 18)(<= ?num 65)))
=>
  (retract ?f2)
  (assert (contador (+ ?cont 1))))
```

- A. El SBR funciona correctamente.
- B. Para que el SBR funcione correctamente es necesario añadir solo la instrucción (assert (lista \$?x1 \$?x2)) en la RHS de la regla.
- C. Para que el SBR funcione correctamente es necesario añadir la instrucción (retract ?f1) y la instrucción (assert (lista \$?x1 \$?x2)) en la RHS de la regla.
- D. Ninguna de las anteriores

25) Dada la BH={ (lista b a a a c a c b b c) (lista1 a c d e f g) } y la siguiente regla, indica cuál será la BH final.

```
(defrule R1
  ?f <- (lista $?x ?a ?a $?y)
  (lista1 $? ?a $?)
=>
```

```
(retract ?f)
(assert (lista $?x ?a $?y)))
```

- A. {(lista b b b c) (lista1 a c d e f g)}
- B. {(lista b b b c)}
- C. {(lista b a c a c b b c) (lista1 a c d e f g)}
- D. {(lista b a c b b c) (lista1 a c d e f g)}

26) Dada la base de hechos inicial: BH={{(lista 5 7 3 1 6 4) (minimo 9999)}} y la siguiente regla para calcular el mínimo de una lista

```
(defrule REGLA
  ?f1 <- (lista $?a ?b $?c)
  ?f2 <- (minimo ?x)
  (test (< ?b ?x))
=>
  (assert (lista $?a $?c))
  (assert (minimo ?b)))
```

Si nuestro objetivo es obtener una base de hechos final (tras la ejecución sucesiva de la regla) en la cual el hecho (minimo ...) solo puede aparecer una vez (conteniendo el valor mínimo de la lista).

¿Cuál de las siguientes afirmaciones es CIERTA para obtener nuestro objetivo?

- A. La regla es correcta
- B. Sería necesario añadir (retract ?f2)
- C. Sería necesario añadir (retract ?f1)
- D. Sería necesario añadir (retract ?f1) y (retract ?f2)

27) Indica cuál es el resultado final CORRECTO tras ejecutar el siguiente SBR con la BH inicial={{(lista 34 77 34)}}:

<pre>(defule R1 (declare (salience 25)) ?f <- (lista \$?x1 ?num \$?x2) => (retract ?f) (printout t "Mensaje 1" crlf))</pre>	<pre>(defrule R2 (declare (salience 10)) ?f <- (lista ?num \$?x ?num) => (retract ?f) (printout t "Mensaje 2" crlf))</pre>	<pre>(defrule R3 => (printout t "Mensaje 3" crlf))</pre>
---	--	---

- A. Mostrará tres veces el "Mensaje 1".
- B. Mostrará una vez "Mensaje 1" y una vez "Mensaje 2".
- C. Mostrará una vez "Mensaje 1" y una vez "Mensaje 3".
- D. Mostrará una vez "Mensaje 1", una vez "Mensaje 2" y una vez "Mensaje 3".

28) El siguiente hecho representa un conjunto de pilas y los bloques que contiene cada una de ellas. El número que aparece después del símbolo 'pila' es el identificador de la pila y a

continuación aparecen los bloques de la pila, siendo el primer bloque el tope de la pila. Indica cuál sería la LHS de una regla para que se instancie CORRECTAMENTE el bloque que se encuentra en la base de cualquier pila del hecho que se muestra a continuación:

(problema pila 1 A F G J K pila 2 B D pila 3 C H I L pila 4)

- A. (problema \$?x1 pila ?num \$?x2 ?y pila \$?x3) (test (not (member pila \$?x1)))
- B. (problema \$?x1 pila ?num \$?x2 ?y pila \$?x3) (test (not (member pila \$?x2)))
- C. (problema \$?x1 pila ?num \$?x2 ?y \$?x3)(test (not (member pila \$?x3)))
- D. Ninguna de las anteriores.

29) Sea un SBR formado por BHinicial={{(lista 2 1 5 3)}}, y las siguientes reglas:

<pre>(defrule R1 (declare (salience 100)) ?f <- (lista \$?x ?z ?y \$?w) (test (< ?z ?y)) => (assert (lista \$?x ?z ?y \$?w)))</pre>	=>	<pre>(defrule R2 (declare (salience 150)) ?f <- (lista \$?x ?z ?y \$?w) (test (>= ?z ?y)) => (assert (lista \$?x ?z ?y \$?w)))</pre>
<pre>(defrule final (declare (salience 200)) (lista \$?list) => (halt))</pre>		

tras el primer *pattern-matching*, ¿cómo quedarían ordenadas las instancias el Conjunto Conflicto (Agenda)?

- A. Una instancia de la regla R1, una instancia de la regla R2 y una instancia de la regla final
- B. Dos instancias de la regla R2, una instancia de la regla R1, una instancia de la regla final
- C. Una instancia de la regla final, dos instancias de la regla R2, una instancia de la regla R1.
- D. Una instancia de la regla final

30) Sea un SBR formado por BHinicial={{(lista 2 1 6 2 3) (elemento 5)}}, y la regla que se muestra a continuación. ¿Cuál sería el contenido final de la BH?

```
(defrule REGLA
  ?f <- (lista $?x ?z $?w)
  (elemento ?y)
  (test (< ?z ?y))
=>
  (assert (lista $?x $?w))
  (retract ?f))
```

- A. {(lista 6) (elemento 5)}

- B. {(lista 2 1 2 3) (elemento 5)}
- C. {(lista) (elemento 5)}
- D. {(lista 2 2) (elemento 5)}

31) Sea la siguiente regla para calcular el Máximo Común Divisor (mcd) de dos números enteros positivos. Indica la respuesta **CORRECTA**:

```
(defrule mcd
  ?a <- (num ?n1)
  ?b <- (num ?n2)
  (test (> ?n1 ?n2))
=>
  (retract ?a)
  (assert (num (- ?n1 ?n2))))
```

- A. Calcula correctamente el mcd quedando un hecho 'num' con dicho valor
- B. Es necesario añadir una regla parada sin prioridad para que el sistema no entre en una ejecución sin fin
- C. Es necesario añadir una regla parada con prioridad para que el sistema no entre en una ejecución sin fin
- D. Ninguna de las anteriores es correcta

32) Dado un SBR cuya Base de Hechos inicial es (lista b a c c a b b a resto), y una única regla:

```
(defrule pasar
  ?a <- (lista $?x ?y ?y $?x $?z resto $?m)
=>
  (retract ?a)
  (assert (lista $?x $?x $?z resto $?m ?y)))
```

El contenido final de la Base de Hechos será:

- A. (lista b a a b b a resto c)
- B. (lista b a resto c a b)
- C. (lista b a b a resto c b)
- D. Ninguna de las anteriores

33) Dado el siguiente hecho: (Dueños coches a b c dueño P coches d dueño Q coches e f dueño R), donde se relaciona los coches y posteriormente su dueño ¿Cuál de los siguientes patrones serviría para obtener el nombre del dueño de un solo coche?

- A. (Dueños \$?x coches ?a dueño \$?z)
- B. (Dueños \$? coches ? dueño ?z \$?)
- C. (Dueños \$?x coches ?a dueño ?z \$?x)

D. (Dueños \$? coches ? dueño ?z)

34) Sea un SBR, con la BH inicial {(lista A B C A B C C B A C B A)}, y la siguiente regla:

```
(defrule regla1
  ?f1 <- (lista $?x1 ?y $?x2 ?y $?x3)
  (test (> (length $?x2) 0))
  (test (not (member ?y $?x2)))
=>
  (retract ?f1)
  (assert (lista $?x1 ?y ?y $?x3)))
```

Si se ejecuta este SBR, el resultado es:

- A. Una lista que solo contiene letras A
 - B. Una lista que solo contiene letras B
 - C. Una lista que solo contiene letras C
 - D. Dependerá de la estrategia de control que se aplique, sea anchura o profundidad.
-

35) Dado el hecho (prueba 1 2 3 4 5 6 7 8 9 10) y la regla :

```
(defrule regla1
  ?f1 <- (prueba $?a $?c)
=>
  (retract ?f1)
  (assert (lista $?c)))
```

En el primer ciclo inferencial:

- A. No se producirá ninguna instanciación
 - B. Se producirán 9 instanciaciones
 - C. Se producirán 10 instanciaciones
 - D. Se producirán 11 instanciaciones
-

36) Dada la base de hechos inicial: BH={(lista 3 5 2 5 3 4 2 9 8 8 9 6) (numero 5) (repeticiones 0)} y la siguiente regla para calcular el número de repeticiones de un elemento de una lista de números naturales

```
(defrule REGLA
  ?f1 <- (lista $?a ?b $?c)
  ?f2 <- (numero ?x)
  ?f3 <- (repeticiones ?z)
  (test (= ?b ?x))
=>
```



```
(assert (lista $?a $?c))
(assert (repeticiones (+ 1 ?z))))
```

Si nuestro objetivo es obtener una base de hechos final (tras la ejecución sucesiva de la regla) en la cual el hecho (repeticiones ...) solo puede aparecer una vez (conteniendo el número de repeticiones del número indicado en el patrón (numero ..) en la lista). ¿Cuál de las siguientes afirmaciones es CIERTA para obtener nuestro objetivo?

- A. La regla es correcta
- B. Sería necesario añadir (retract ?f1)
- C. Sería necesario añadir (retract ?f1) y (retract ?f3)
- D. Sería necesario añadir (retract ?f3)

37) Sea un SBR formado por BHinicial={{(lista 2 1 5 3)}}, y las siguientes reglas:

```
(defrule R1
  (declare (salience 200))
  ?f <- (lista $?x ?z ?y $?w)
  (test (< ?z ?y))
=>
  (assert (lista $?x ?z ?y $?w)))

(defrule R2
  (declare (salience 50))
  ?f <- (lista $?x ?z ?y $?w)
  (test (>= ?z ?y))
=>
  (assert (lista $?x ?z ?y $?w)))

(defrule final
  (declare (salience 150))
  (lista $?list)
=>
  (halt))
```

tras el primer *pattern-matching*, ¿como quedarían ordenadas las instancias el Conjunto Conflicto (Agenda)?

- A. Una instancia de la regla R1, una instancia de la regla final, y dos instancias de la regla R2
- B. Dos instancias de la regla R2, una instancia de la regla R1, una instancia de la regla final
- C. Una instancia de la regla final, dos instancias de la regla R2, una instancia de la regla R1
- D. Una instancia de la regla final

38) Dada la siguiente parte izquierda de una regla

```
(defrule r1
  (lista $?x $?w ?y ?z $?x)
=>
  ....
```

, y el siguiente hecho: (lista a b a b c c), ¿cuántas instancias de esta regla se incluirán en la agenda?:

- A. 0
- B. 1**
- C. 3
- D. 5

39) Dada Sea el siguiente patrón para el problema de las torres de Hanoi:

$(\text{hanoi } [\text{tower } tw^s \ d1^s \ d2^s \ d3^s \ d4^s]^m) ;; tw^s \in \{T1, T2, T3\} \ d_i^s \in [0-4]$

donde dos ejemplos de hecho serían (hanoi tower T1 2 4 0 0 tower T2 0 0 0 0 tower T3 1 3 0 0)
(hanoi tower T1 2 4 0 0 tower T2 1 3 0 0 tower T3 0 0 0 0).

¿Movería la siguiente regla un disco de la torre T3 a cualquiera de las otras dos torres T1, T2 (a su izquierda)?

```
(defrule move-disk-from-T3-to-Tx
  (hanoi $?rest1 tower ?Tx ?d2 $?rest2 0 $?rest3 tower T3 ?d1 ?a ?b ?c)
  (test (or (= ?d2 0) (< ?d1 ?d2)))
=>
  (assert (hanoi $?rest1 tower ?Tx ?d1 ?d2 $?rest2 $?rest3 tower T3 ?a ?b ?c 0)))
```

- A. Sí, la regla es correcta
- B. Sería correcta si se añade la comprobación (not (member tower \$?rest2)) en la LHS de la regla.
- C. Además de la comprobación del apartado (B) debería también comprobarse (<> ?d1 0) en la LHS de la regla.**
- D. Además de la comprobación de los apartados (A) y (B), debería comprobarse (> (length \$?rest2) 0) en la LHS de la regla.

40) Sea un SBR formado por $BH_{inicial} = \{(lista \ 3 \ 4 \ 5 \ 6 \ 6 \ 6 \ 8 \ 9)\}$, y las siguientes reglas:

<pre>(defrule R1 ?f <- (lista \$?x ?z ?y \$?w) (test (= ?z ?y)) => (assert (lista \$?x ?z ?y \$?w)))</pre>	<pre>(defrule R2 ?f <- (lista \$?x ?z ?y \$?w) (test (> ?z ?y)) => (assert (lista \$?x ?z ?y \$?w)))</pre>
--	---

¿Cuál sería el contenido del Conjunto Conflicto (Agenda) tras el primer pattern-matching?

- A. Dos instancias de la regla R1 y dos de la R2
- B. Dos instancias de la regla R1**

- C. Dos instancias de la regla R1 y una instancia de la regla R2
D. Ninguna instancia

41) Dada la BH={{(lista e b a c d e a) (lista1 a c a d e e f g)}} y la regla R1, indica cuál será la BH final.

```
(defrule R1
  ?f <- (lista $?x ?a $?y)
  (lista1 $?p ?a $?q ?a $?r)
=>
  (retract ?f)
  (assert (lista $?x $?y)))
```

- A. {(lista e b a c d f g) (lista1 a c a d e e f g)}
B. {(lista b) (lista1 a c a d e e f g)}
C. {(lista b c d) (lista1 a c a d e e f g)}
D. {(lista e b a c d) (lista1)}

42) Indica cuál es el resultado final CORRECTO tras ejecutar el siguiente SBR con la BH inicial={{(lista 34 77 34)}}:

(defrule R1 (declare (salience 15)) ?f <- (lista \$?x1 ?num \$?x2) => (retract ?f) (printout t "Mensaje 1" crlf))	(defrule R2 (declare (salience 20)) ?f <- (lista ?num \$?x ?num) => (retract ?f) (printout t "Mensaje 2" crlf))	(defrule R3 (declare (salience 30)) => (printout t "Mensaje 3"))
--	--	---

- A. Mostrará tres veces el "Mensaje 3".
B. Mostrará una vez "Mensaje 2" y una vez "Mensaje 3".
C. Mostrará una vez "Mensaje 3" y una vez "Mensaje 2".
D. Mostrará una vez "Mensaje 3", una vez "Mensaje 2" y una vez "Mensaje 1".

43) Sea un SBR formado por BHinicial={{(lista 4 5 6 6 6 8 4 8)}}, y las siguientes reglas:

```
(defrule R1
  ?f <- (lista $?x ?z ?y $?w)
  (test (< ?z ?y))
=>
  (assert (lista $?x ?z ?y $?w)))

(defrule R2
  ?f <- (lista $?x ?z ? ?y $?w)
  (test (> ?z ?y))
=>
```

(assert (lista \$?x ?z \$?w)))

¿Cuál sería el contenido del Conjunto Conflicto (Agenda) tras el primer pattern-matching?

- A. Cinco instancias de la regla R1 y una de la R2
 - B. Cuatro instancias de la regla R1 y ninguna de la R2
 - C. Cinco instancias de la regla R1 y ninguna de la R2
 - D. Cuatro instancias de la regla R1 y una de la R2
-

44) Si se quisiera que una regla en CLIPS hiciera matching con el siguiente hecho: (lista nota 3 nota 5 nota 10), que patrón se tendría que incluir en su parte izquierda

- A. (lista \$?nota \$?num \$?nota \$?num)
 - B. (lista nota \$?num \$?nota)
 - C. (lista \$?num \$?nota num)
 - D. (lista \$? ?x \$? ?x \$? ?x)
-

45) Dado el siguiente hecho: (problema torre a b c nombre A torre a nombre B torre nombre C)
¿Cuál de los siguientes patrones NO serviría para obtener únicamente el nombre de una torre con un único elemento en ella?

- A. (problema \$?x torre ?a \$?y nombre ?z \$?)
 - B. (problema \$? torre ?a nombre ?z \$?x)
 - C. (problema \$?x torre ?a nombre ?z \$?)
 - D. (problema \$? torre ? nombre ?z \$?)
-

46) Dada la base de hechos inicial: BH={{(lista 6 3 5 1 4 7 2 6 3) (pares 0)}} y la siguiente regla para calcular el número de pares en una lista de números naturales

```
(defrule contar-pares
  ?f1 <- (lista $?a ?b $?c)
  ?f2 <- (pares ?p)
  (test (= 0 (mod ?b 2)))
=>
  (assert (lista $?a $?c))
  (assert (pares (+ 1 ?p))))
```

Si nuestro objetivo es obtener una base de hechos final (tras la ejecución sucesiva de la regla) en la cual el hecho (pares ...) solo puede aparecer una vez (conteniendo el número de pares en la lista). ¿Cuál de las siguientes afirmaciones es CIERTA para obtener nuestro objetivo?

- A. La regla es correcta.
- B. Sería necesario añadir (retract ?f1)
- C. Sería necesario añadir (retract ?f1) y (retract ?f2)

D. Sería necesario añadir (retract ?f2)

47) Se quiere una regla en CLIPS que haga matching con el siguiente hecho: (lista a b a a b b c a b c). ¿Cuál de los siguientes patrones habría que incluir en la parte izquierda de dicha regla?

- A. (lista \$? \$?x \$? \$?x ?)
- B. (lista \$?y ?x ?x \$?y \$?)
- C. (?a \$?c)
- D. (lista \$? \$?y ?x ?x ?x \$? \$?y \$? ?x)

48) Sea un SBR formado por BHinicial={{(lista 4 5 6 6 6 8 4 8)}}, y las siguientes reglas:

(defrule R1	(defrule R2
?f <- (lista \$?x ?z ?y \$?w)	?f <- (lista \$?x ?z ? ?y \$?w)
(test (< ?z ?y))	(test (> ?z ?y))
=>	=>
(assert (lista \$?x ?z ?y \$?w)))	(assert (lista \$?x ?z \$?w)))

¿Cuál sería el contenido del Conjunto Conflicto (Agenda) tras el primer pattern-matching?

- A. Cinco instancias de la regla R1 y una de la R2
- B. Cuatro instancias de la regla R1 y ninguna de la R2
- C. Cinco instancias de la regla R1 y ninguna de la R2
- D. Cuatro instancias de la regla R1 y una de la R2

49) Dada la siguiente base de hechos BH={{(ciudades Valencia min 14 max 36 Alicante min 18 max 34 Castellón min 12 max 30 Requena min 8 max 38)(selección Castellón)}}, se desea hacer una regla que calcule la diferencia de temperatura (máxima – mínima) de la ciudad seleccionada. ¿Cuál de los siguientes patrones **NO SERVIRÍA** para la parte izquierda de la regla?

- A. (selección ?c) (ciudades \$?x ?c ? ?t1 ? ?t2 \$?w)
- B. (selección ?c) (ciudades \$? ?c ?n1 ?t1 ?n2 ?t2 \$?)
- C. (selección ?c) (ciudades \$?x ?c ? ?t1 ? ?t2 \$?x)
- D. (selección ?c) (ciudades \$?x ?c min ?t1 max ?t2 \$?w)

50) Dado el SBR, indica la afirmación **CORRECTA** tras realizarse el primer *pattern-matching*:

```
(deffacts prueba
  (prueba 1 2 3 4 5 6 7 8 9 10))
```

```
(defrule regla
  ?f1 <- (prueba $?a ?b $?c ?d)
  =>
```

```
(retract ?f1)
(assert (lista $?c)))
```

- A. Se producirán 9 instanciaciones
- B. Se producirán 10 instanciaciones
- C. Se producirán 11 instanciaciones
- D. No se producirá ninguna instanciación

51) Dada la base de hechos inicial: BH={{(lista 6 5 9 0 4 4 3) (minimo 10)}} y la siguiente regla para calcular el mínimo de una lista

```
(defrule r1
  ?f1 <- (lista $?a ?b $?c)
  ?f2 <- (minimo ?m)
  (test (< ?b ?m))
=>
  (assert (minimo ?b))
  (retract ?f2))
```

Si nuestro objetivo es obtener una base de hechos final (tras la ejecución sucesiva de la regla) en la cual el hecho 'mínimo' solo puede aparecer una vez (conteniendo el valor mínimo de la lista). ¿Cuál de las siguientes afirmaciones es **CIERTA** para obtener nuestro objetivo?

- A. Sería necesario añadir (retract ?f1)
- B. Sería necesario quitar (retract ?f2)
- C. Sería necesario cambiar el test poniendo (test (> ?b ?m))
- D. La regla es correcta

52) Un hecho en CLIPS como (puzzle 2 8 3 1 6 4 7 0 5) es una representación lineal de un tablero 3x3 de la configuración del 8-puzzle que se puede ver en la figura:

2	8	3
1	6	4
7		5

Sea la siguiente regla en CLIPS:

```
(defrule R1
  ?f<-(puzzle $?x ?a ?b ?c 0 $?y)
=>
  (retract ?f)
  (assert (puzzle $?x 0 ?b ?c ?a $?y)))
```

Indica la respuesta **CORRECTA**:

- A. La regla realiza un movimiento correcto de una ficha hacia abajo.
 - B. La regla realiza un movimiento correcto de una ficha hacia arriba.
 - C. Para que fuera un movimiento correcto de una ficha hacia abajo habría que añadir un test para comprobar que el 0 no está en la fila superior.
 - D. Para que fuera un movimiento correcto de una ficha hacia arriba habría que añadir un test para comprobar que el 0 no está en la fila inferior.
-

53) Dado el hecho:

(servidores servidor 1 cpu 5 gpu 3 servidor 2 cpu 7 gpu 5 servidor 3 cpu 10 gpu 4)

donde el número que aparece después de servidor es el identificador de un servidor y los valores numéricos que aparecen después de cpu y gpu indican el número de CPU y GPU que tiene el servidor. ¿Cual de los siguientes patrones permitiría obtener el identificador de un servidor y el número de GPUs que tiene dicho servidor?

- A. (servidores servidor ?s \$? gpu ?g \$?)
 - B. (servidores \$? servidor ? cpu \$? gpu ?g \$?)
 - C. (servidores servidor ?s cpu ? gpu ?g)
 - D. (servidores \$? servidor ?s cpu ? gpu ?g \$?)
-

54) De acuerdo al siguiente SBR, ¿Cuál sería el estado final de la BH tras la ejecución del mismo?

(deffacts prueba
 (A) (B) (C) (D))

(defrule R1
 ?f1<-(A)
 (B)
=>
 (assert (E))
 (retract ?f1))

(defrule R2
 ?f1<-(C)
 (E)
=>
 (assert (A))
 (retract ?f1))

- A. BH= A, B, C, D
- B. BH= A, B, C, D, E
- C. BH= B, D, E
- D. La ejecución no acaba nunca, puesto que R1 y R2 se ejecutan continuamente

55) El hecho CLIPS (transporte bus 1 Plaza-Ayuntamiento Malvarrosa bus 2 Benimaclet San-Marcelino bus 3 Torrefiel Hospital-La-Fe) representa el origen y destino de las rutas de tres autobuses. Determina cuál sería el patrón correcto para averiguar el número de autobús que tengo que coger si quiero llegar a San-Marcelino:

- A. (transporte \$?x bus ?numbus San-Marcelino \$?y)
 - B. (transporte \$?x bus ?numbus San-Marcelino ?y)
 - C. (transporte \$?x bus ?numbus ? San-Marcelino \$?y)
 - D. (transporte \$?x bus ?numbus ? San-Marcelino ?y)
-

56) Sea el siguiente conjunto de hechos y regla en CLIPS

```
(deffacts datos
(cadena B C A)
(cadena A C A B)
(union))
(defrule R1
(cadena $?x ?letra $?y)
(union $?lis)
(test (not (member ?letra $?lis)))
=>
(assert (union ?letra $?lis)))
```

Tras la ejecución del SBR, indica la respuesta **CORRECTA:**

- A. Se genera un único hecho 'union' que contiene las tres letras A B y C sin repeticiones
- B. Se generan 6 hechos 'union' cada uno de los cuales contiene las tres letras A B y C en diferente orden
- C. Se generan 6 hechos 'union' cada uno de los cuales contiene las tres letras A B y C en diferente orden y 6 hechos 'union' cada uno de los cuales contiene una combinación de dos letras en diferente orden
- D. Se genera un total de 15 hechos 'union'

PROBLEMAS

Ejercicio 1

Dada una BH que represente una lista de naturales no ordenados, escribir una única (si es posible) regla de producción que obtenga, como BH-meta, la lista inicial con sus elementos ordenados. Por ejemplo:

BH inicial: (lista 4 5 3 46 12 10)

BH final: (lista 3 4 5 10 12 46)

Determinar además las instancias de dicha regla que resultan aplicables sólo para la BH-inicial del ejemplo.

Solución:

```
(defrule ordenar
  ?a <- (lista $?x ?y ?z $?w)
  (test (< ?z ?y))
=>
  (retract ?a)
  (assert (lista $?x ?z ?y $?w)))
```

Instancias aplicables con la BH-inicial:

- 1) ?y=5, ?z=3
- 2) ?y=46, ?z=12
- 3) ?y=12, ?z=10

Ejercicio 2

Diseña un sencillo SBR (una única regla si es posible) para determinar el número de aciertos de los boletos sellados de la Lotería Primitiva. En la BH se dispone de un hecho que determina la combinación ganadora y otro hecho que especifica la combinación jugada (asumir que existe una variable global inicializada a 0 que sirve para contar los aciertos). Ejemplo de BHinicial:

BHinicial= {(boleto-ganador 2 5 8 13 24 35) (combinacion 3 5 15 24 26 37)}

Asumiendo que los números de la combinación no se introducen en orden creciente, ¿funcionaría el S.P. que has diseñado en el apartado a)?. ¿Por qué?.

Solución:

```
(defglobal ?*num-aciertos* = 0)

(defrule cuenta-aciertos
  (boleto-ganador $?x ?y $?z)
  (combinacion $?a ?y $?b)
=>
  (bind ?*num-aciertos* (+ ?*num-aciertos* 1))
  (printout t "El numero de aciertos hasta el momento es " ?*num-aciertos* crlf))

(deffacts datos
  (boleto-ganador 2 4 8 12 22 34))
```

(combinacion 42 13 22 30 2 5))

Sí, el SBR funcionaría de igual modo porque lo que hace la regla es detectar cualquier combinación de dos números iguales de los hechos (boleto-ganador) y (combinacion).

Ejercicio 3

Sea un SBR cuya BH inicial es BH={{(lista 3 6 8 5 10)}} y cuya Base de Reglas se compone de las siguientes reglas:

(defrule R1 ?f <- (lista ?x ?y \$?z) (test (< ?x ?y)) => (retract ?f) (assert (lista ?y ?z))	(defrule R2 ?f <- (lista ?y \$?x) (test (and (<= (length \$?x) 3) (>= (length \$?x) 1))) => (retract ?f) (assert (lista ?x)))
---	--

asumiendo una estrategia de búsqueda en anchura (mayor prioridad para las instancias más antiguas, comenzando por R1), ¿cuál sería el estado final de la BH?. Realiza una traza que muestre el proceso inferencial.

Solución:

Base de Hechos (BH)	Conjunto Conflictivo (CC)
H1: (lista 3 6 8 5 10) (1*)	R1: H1 (1*)
=====	=====
H2: (lista 6 8 5 10) (2*)	R1: H2 (2*)
	R2: H2 (se elimina al eliminar H2)
=====	=====
H3: (lista 8 5 10) (3*)	R2: H3 (3*)
=====	=====
H4: (lista 5 10) (4*)	R1: H4 (4*)
	R2: H4 (se elimina al eliminar H4)
=====	=====
H5: (lista 10)	No hay nuevas instancias
=====	=====

Nota: El número entre paréntesis indica el orden de disparo de la regla y la eliminación del hecho correspondiente como consecuencia del disparo de la regla. (*) indica que el hecho o la instancia de la regla se eliminan de la BH o CC respectivamente.

BHfinal= {(lista 10)}

Ejercicio 4

Sea un SBR cuya BH inicial es BH={{(lista 1 2 3 4)}} y cuya Base de Reglas se compone de las siguientes reglas:

```
(defrule R1
  ?f <- (lista ?x $?z)
=>
  (retract ?f)
  (assert (lista ?z))
  (assert (elemento ?x)))
```

```
(defrule R2
  ?f <- (elemento ?x)
  (elemento ?y)
  (test (< ?x ?y))
=>
  (retract ?f)
  (assert (lista-new ?x ?y)))
```

asumiendo una estrategia de búsqueda en anchura (mayor prioridad para los hechos e instancias de reglas más antiguas, comenzando por R1):

- ¿cuál sería el estado final de la BH?. Realiza una traza que muestre el proceso inferencial.
- Si la BH inicial fuera BH={{(lista 1 2 2 4)}}, ¿cuántos hechos (lista-new) habría en la BH final ?. ¿Cuáles?.
- Si se eliminara el comando (retract ?f) de R1, ¿qué cambios se producirían con la BH inicial= {(lista 1 2 3 4)} respecto a los hechos (lista-new) ?
- Si se eliminara el comando (retract ?f) de R2, ¿qué cambios se producirían con la BH inicial= {(lista 1 2 3 4)} respecto a los hechos (lista-new) ?

Solución:

a)

Base Hechos (BH)	Conjunto Conflictivo (CC)
H1: (lista 1 2 3 4)(1*)	R1,H1 (1*)
=====	=====
H2: (lista 2 3 4) (2*)	R1, H2 (2*)
H3: (elemento 1) (4*)	
=====	=====
H4: (lista 3 4) (3*)	R1, H4 (3*)
H5: (elemento 2) (6*)	R2, H3, H5 (4*)
=====	=====
H6: (lista 4) (5*)	R1, H6 (6*)
H7: (elemento 3) (7*)	R2, H3, H7 (5*, se elimina al eliminar H3)
	R2, H5, H7 (7*)
=====	=====
H8: (lista-new 1 2)	
=====	=====
H9: (lista)	R2, H5, H10 (8*, se elimina al eliminar H5)
H10: (elemento 4)	R2, H7, H10 (9*)
=====	=====
H11: (lista-new 2 3)	
=====	=====
H12: (lista-new 3 4)	

Nota: (*) indica que el hecho o la instancia de la regla se eliminan de la BH o CC respectivamente.

BHfinal= {(lista), (elemento 4), (lista-new 1 2), (lista-new 2 3), (lista-new 3 4)}

b) Asumiendo que no se permite la duplicidad de hechos, los hechos lista-new que habría son:

(lista-new 1 2), (lista-new 2 4)

c) Ninguno. Al no eliminar el hecho (lista ...) que ha provocado la instanciación de R1 no se produce ningún cambio porque dicho hecho no volverá a instanciar de nuevo regla-1 (no se puede producir una instanciación de una misma regla, con un mismo hecho asignando los mismos valores a las variables del patrón).

d) Se producirían todas las combinaciones posibles con los hechos (lista-new ...). Esto es, (lista-new 1 2) **(lista-new 1 3)** **(lista-new 1 4)** (lista-new 2 3) **(lista-new 2 4)** (lista-new 3 4). (los hechos que están en negrita serían los nuevos hechos que aparecerían si se eliminara el comando (retract ?f) de R2.

Ejercicio 5

Escribir **una única regla** en clips que, dada una lista de números, obtenga una nueva lista en la que se reemplace por un 0 todos aquellos números cuyo valor no coincida con su orden de posición (asumiendo que el primer número a la izquierda ocupa la posición uno). Por ejemplo, dada la lista (lista 15 2 4 6 5 3 7 8 15), obtendría la lista (lista 0 2 0 0 5 0 7 8 0).

Solución:

```
(defacts lista (lista 1 2 4 6 5 3 7 8 9)) ;Es un ejemplo de lista

(defrule uno
  ?f <- (lista $?x ?v $?y)
  (and (test (<> ?v 0)) ;Debe controlarse, para que pueda acabar!!
        (test (<> (length $?x) (- ?v 1))))

=>
  (retract ?f)
  (assert (lista $?x 0 $?y)))
```

Ejercicio 6

Dada una BH que represente una lista de naturales no ordenados y posiblemente repetidos un número indefinido de veces cada uno de los números, escribir una **única** (si es posible) regla de producción que obtenga, como BH-meta, la lista inicial de números, pero en los que no haya ningún número repetido.

Ejemplo:

BH-Inicial: (lista 1 3 4 5 6 7 4 6 4 4 3 4 5 6 4 5 6 4 8 5 7 3 2 4 4 4 1 5 6 7 1 2 3 2 3 4 6 5 3 4 5 1 4 3)
BH Final: (lista 2 7 6 5 4 3 1 8)

Solución:

Esta es una posible regla. Hay otras alternativas.

```
(defrule examen
  ?f1 <- (lista $?a ?x $?b ?x $?c)
  =>
  (retract ?f1)
  (assert (lista ?x ?a ?b ?c)))
```

```
(deffacts hechos
  (lista 1 3 4 5 6 7 4 6 4 4 3 4 5 6 6 4 5 6 4 8 5 7 3 2 4 4 4 1 5 6 7 1 2 3 2 3 4 6 5 3 4 5 1 4 3))
```

Ejercicio 7

Un cartón de bingo se compone de 5 líneas de 5 números cada una. La BH de un SBR contiene un hecho que representa una línea ganadora, por ejemplo, (línea-ganadora 12 21 34 56 77). Escribe un hecho que represente un cartón de bingo y una regla de producción que permita determinar si hay alguna línea ganadora en el cartón.

Solución:

Patrón para el cartón: (carton línea x1^m línea x2^m línea x3^m línea x4^m línea x5^m)
 x1, x2, x3, x4, x5 ∈ INTEGER

```
(defrule bingo
  (línea-ganadora $?x)
  (carton $?a línea $?x $?b)
  =>
  (printout t "Línea ganadora " crlf))
```

```
(deffacts datos
  (línea-ganadora 12 21 34 56 77)
  (carton línea 1 2 3 4 5 línea 23 44 55 66 77 línea 12 21 34 56 77 línea 6 7 8 9 10 línea 18 28 38 48 58))
```

Ejercicio 8

Sea un SBR cuya BH inicial es BH={ (elem e) (lista a b c d e f) }, y cuya Base de Reglas se compone de las reglas R1 y R2.

<pre>(defrule R1 ?l1 <- (elem ?e) ?l2 <- (lista \$?a ?e \$?b) => (assert (lista ?e ?a ?b)) (assert (cambio)))</pre>	<pre>(defrule R2 ?l1 <- (cambio) (lista \$?l) => (retract ?l1))</pre>
--	---

asumiendo una estrategia en anchura (mayor prioridad para los hechos e instancias de reglas más antiguas, comenzando por R1), ¿cuál será el estado de la BH final?. Realiza una traza que muestre el proceso inferencial.

Solución:

<u>Base Hechos (BH)</u>	<u>Conjunto Conflicto (CC)</u>
H1: (elem e) H2: (lista a b c d e f)	R1: H1, H2 (?e=e, \$?a=a b c d, \$?b=f)
=====	=====
H1, H2 H3: (lista e a b c d f) H4: (cambio)	R1: H1, H3 (?e=e, \$?a={}) \$?b=a b c d f) R2: H4, H2 (\$?l= a b c d e f) R2: H4, H3 (\$?l= e a b c d f)
=====	=====
H1, H2, H3, H4 no se generan los hechos de los comandos assert porque son los mismos que H3 y H4	R2: H4, H2 (\$?l= a b c d e f) R2: H4, H3 (\$?l= e a b c d f) – se elimina al eliminar la instancia anterior el hecho H4 -
=====	=====
H1, H2, H3	
=====	
BH final= {H1: (elem e), H2: (lista a b c d e f), H3: (lista e a b c d f)}	

Ejercicio 9

Sea un SP cuya BH inicial es BH={{S 0 lista 2 3 1 20}} y cuya Base de Reglas se compone de la siguiente regla:

```
(defrule exam
  ?f <- (S ?s lista $?x ?y ?z $?w)
  (test (< ?z ?y))
=>
  (assert (S (+?s 1) lista ?x ?z ?y ?w)))
```

- 1) Asumiendo un encadenamiento hacia delante y una estrategia de búsqueda en anchura (mayor prioridad para los hechos e instancias de reglas más antiguas), ¿cuál será el estado final de la BH?. Realiza una traza que muestre el proceso inferencial.
- 2) Asume que se añade un comando (retract ?f) en la RHS de la regla; muestra la nueva regla y el estado final de la BH.

Solución:

a)

Base de Hechos (BH)

Conjunto Conflicto (CC)

H1: (S 0 list 2 3 1 20)

exam: H1, ?y=3, ?z=1, \$?x=(2), \$?w=(20) (1*)

=====

H2: (S 1 list 2 1 3 20)

exam: H2, ?y=2, ?z=1, \$?x=(), \$?w=(3 20) (2*)

=====

H3: (S 2 list 1 2 3 20)

No se producen más activaciones. De todas las posibles activaciones el patrón de la regla con el hecho H3, ninguna instancia satisface el test.

=====

La BH final contendrá tres hechos: BHfinal={{(S 0 list 2 3 1 20) (S 1 list 2 1 3 20) (S 2 list 1 2 3 20)}}

b) La traza será la misma excepto que se elimina de la BH el correspondiente hecho. De este modo, en cada ciclo del motor de inferencia, la BH contendrá únicamente un hecho. El contenido final de la BH será el hecho H3. BH final= BHfinal={{(S 2 list 1 2 3 20)}}.

Ejercicio 10

Sea un SP cuya BH inicial es BH={{(lista a a b a) (par a 1) (par b 2)}} y cuya Base de Reglas se compone de la siguiente regla:

```
(defrule R1
  ?f <- (lista $?x ?sym $?y)
  (par ?sym ?num)
=>
  (retract ?f)
  (assert (lista $?x ?num $?y)))
```

Se pide:

Asumiendo un encadenamiento hacia delante, realiza y muestra una traza del SP aplicando sucesivamente el ciclo reconocimiento-acción de RETE hasta que el proceso se detenga. ¿Cuál es el resultado de la BH final?

Solución:

Base de Hechos	Agenda (instancias de reglas)
f-1:(lista a a b a) f-2:(par a 1) f-3:(par b 2)	R1:f-1,f-2 {?sym=a, \$?x=(), \$?y=(a b a), ?num=1, ?f=1} R1:f-1,f-2 {?sym=a, \$?x=(a), \$?y=(b a), ?num=1, ?f=1} R1:f-1,f-2 {?sym=a, \$?x=(a a b), \$?y=(), ?num=1, ?f=1} R1:f-1,f-3 {?sym=b, \$?x=(a a), \$?y=(a), ?num=2, ?f=1}
f-4:(lista 1 a b a)	R1:f-4,f-2 {?sym=a, \$?x=(1), \$?y=(b a), ?num=1, ?f=4} R1:f-4,f-3 {?sym=b, \$?x=(1 a), \$?y=(a), ?num=2, ?f=4} R1:f-4,f-2 {?sym=a, \$?x=(1 a b), \$?y=(), ?num=1, ?f=4}
f-5:(lista 1 1 b a)	R1:f-5,f-3 {?sym=b, \$?x=(1 1), \$?y=(a), ?num=2, ?f=5} R1:f-5,f-2 {?sym=a, \$?x=(1 1 b), \$?y=(), ?num=1, ?f=5}
f-6:(lista 1 1 2 a)	R1:f-6,f-2 {?sym=a, \$?x=(1 1 2), \$?y=(), ?num=1, ?f=6}
f-7: (lista 1 1 2 1)	

En la primera iteración se producen 4 instancias de R1, se dispara la primera de la agenda y se borran el resto por ser dependientes del hecho que se elimina f-1. En las siguientes iteraciones se produce sucesivamente una instancia menos ya que en cada iteración se reemplaza en el hecho (lista ...) un símbolo por un número. Independientemente de la estrategia de la agenda y el orden en el que se disparen las reglas, la BH final contendrá únicamente los hechos (lista 1 1 2 1)(par a 1)(par b2).