Sistemas Inteligentes
Grado de Ingeniería Informática
Universidad de Córdoba

Relación de ejercicios nº 2: Reglas

NOTAS:

- Para resolver estos ejercicios no pueden emplearse prioridades, módulos, variables globales, ni funciones procedimentales (condicionales, bucles...), ni eliminación, creación, o alteración de reglas durante la ejecución.
- Los programas deben cargar sin errores con la orden `load`
- La solución de los ejercicios no debe depender del posible orden en el que CLIPS pudiese disparar las reglas activas. Por ello, se recomienda hacer pruebas con la estrategia de ejecución de reglas `Random` (uso: `(set-strategy random)`).

1. En una granja hay animales de los siguientes tipos: perros, gatos, patos, vacas, ovejas y gallos. Escribe una serie de reglas de manera que en base a un hecho que indique el sonido que hace un animal identifique qué clase de animal es. Por ejemplo, si se afirma un hecho como `(sonido kikiriki)`, el programa debe imprimir por pantalla el mensaje "`Se trata de un gallo`".
2. Repite el ejercicio anterior pero utilizando ahora la siguiente plantilla y una sola regla que resuelva el ejercicio:

```
(deftemplate animal
     (slot nombre)
     (slot sonido))
```

3. Haz un programa que dado un conjunto de hechos `(datos <val1> <val2>)` (*hechos de tipo* `datos` *con dos valores numéricos*), detecte qué hechos cumplen que el segundo valor sea mayor que el primero y los imprima.
4. Haz un programa que dado un conjunto de hechos de tipo `datos` con un número indefinido de valores, detecte e imprima aquellos tal que el primer valor sea par y menor o igual al último.
5. Haz un programa que dado un único hecho `datos` con un número indefinido de valores (ejemplo: `(datos hola 1 3 nuevo 1 adios)`), elimine todas las apariciones del valor `1`.
6. Haz un programa que detecte e imprima los hechos `(vector <nombreVector> ...)` que contengan los valores 3 y 4 en alguna posición, y que entre éstos haya un número impar de valores. Se debe utilizar la función `length$` (ver documentación).
7. Haz un programa que dado un único hecho `datos` con un número indefinido de valores, elimine los que no sean numéricos.
8. Haz un programa que dado un conjunto de hechos `(vector <nombreVector> <val1> ... <valN>)`, detecte aquellos hechos cuyos valores no están ordenados de menor a mayor e imprima el mensaje "`El vector <nombreVector> no está ordenado`". Se entenderá que en la base de hechos no habrá dos hechos `vector` con mismo nombre de vector.
9. Haz un programa que dado un conjunto de hechos de la forma `(dato 1)`, `(dato 5)`, `(dato verde)`..., cree un único hecho `(todos-los-datos ...)` con todos los valores de los hechos anteriores.

10. Haz un programa que dado un conjunto hechos (`vector <nombreVector> <val1> ... <valN>`) con valores numéricos, ordene sus valores de menor a mayor.
11. Haz un programa que resuelva el ejercicio 9 pero sin eliminar los hechos (`dato 1`), (`dato 5`), (`dato verde`)….
12. Haz un programa que dado un conjunto de hechos de tipo `dato` y un único valor numérico, imprima los valores numéricos por pantalla de menor a mayor. Vigila que el programa funciona correctamente incluso con la estrategia de ejecución de reglas `Random`.
13. Haz un programa que resuelva el ejercicio 9, pero que sólo utilice los hechos con valores numéricos y los inserte de forma ordenada (de menor a mayor) en el hecho `todos-los-datos`.
14. Una planta industrial tiene diez sensores identificados por un código numérico entre 1 y 10. Cada sensor puede encontrarse en un estado `correcto` o `incorrecto`. Escribe una plantilla que permita representar la información relativa a los sensores y un conjunto de reglas que imprima un mensaje de advertencia si tres o más sensores se encuentran en un estado incorrecto. Sólo debe mostrarse un mensaje de error aunque haya más de tres sensores en estado incorrecto.
15. Escribe un programa para ayudar a una persona a decidir qué plantas podría plantar. La siguiente tabla indica las características de una serie de plantas (tolerancia al frío, tolerancia a la sombra, tolerancia al clima seco...). La entrada al programa debe consistir en un conjunto de hechos (`caracteristica-deseada <característica>`) que indiquen características que se desee que tenga una planta. El programa debe mostrar por pantalla el nombre de las plantas que cuenten con las características indicadas, aunque aparte de esas cuenten con otras características.

Planta	Frío	Sombra	Sequedad	Suelo húmedo	Suelo ácido	Ciudad	Maceta	Cuidado fácil	Crece rápido
Hortensia		xx				x	x		x
Adelfa						x	x	x	x
Laurel	x	x	x	x		x		x	x
Madreselva						x	x	x	x
Gardenia		x			x		x		
Enebro	x		x		x	x		x	
Pimentero	x	x		x	x			x	
Escaramujo	x	x		x		x		x	
Aucuba		x	x				x	x	
Azalea		x		x	x		x		

16. Haz un programa en base a la información del ejercicio anterior, pero ahora se imprimirán sólo aquellas plantas que cumplan exactamente con todas las características deseadas (no puede contener otras características)
17. Complementa el ejercicio 8, con una nueva regla, para que se detecten los vectores que sí están ordenados. Es decir, sin modificar los hechos ni crear hechos auxiliares, se debe imprimir un mensaje por cada vector (`vector <nombreVector> <val1> ... <valN>`), indicando si sus elementos están ordenados o no. Se entenderá que en la base de hechos no habrá dos hechos `vector` con mismo nombre de vector.
18. En una residencia de estudiantes se desea mantener información sobre los estudiantes alojados y las habitaciones de la residencia que ocupan. Se tendrán en cuenta las siguientes consideraciones:

- Hay habitaciones de cuatro tipos: simples, dobles, triples y cuádruples.
- Es más económico llenar las habitaciones más grandes
- Todos los ocupantes de una habitación deben ser del mismo sexo.
- Todos los ocupantes de una habitación deben ser fumadores o no fumadores.

Define las plantillas necesarias para poder almacenar la información necesaria. Además, escribe las reglas necesarias para alojar a un estudiante de manera que sea asignado a una habitación que esté ya parcialmente ocupada y sea compatible o, si no, que sea asignado a la mayor habitación libre disponible. Si no hay ninguna habitación disponible, deberá mostrarse un mensaje por pantalla.

19. Haz un programa que dado un único hecho `vector` con un número indefinido de valores numéricos, imprima el valor que se sitúa justo en medio, o la media de los dos valores de en medio.
20. Haz un programa que dado un único hecho `vector`, detecte si sus valores se repiten de forma simétrica. Siempre que la salida sea correcta, puedes modificar el vector o utilizar hechos auxiliares.
21. OPCIONAL (difícil): ¿Sabrías resolver el ejercicio anterior sin modificar el vector ni utilizar hechos auxiliares?