

PRÁCTICA GUIADA DE LABORATORIO 7 – CLIPS Semestre 2022-1

Nombre/Apellidos:

Código:

1. INTRODUCCION

El presente documento pretende servir de guía para la práctica de laboratorio del curso Inteligencia Artificial, a realizarse el 18 de Julio del 2022.

Se presentan algunos tópicos del lenguaje CLIPS con sus respectivos ejemplos, y se proponen algunos ejercicios para practicar lo aprendido.

2. OBJETIVOS DE LA PRÁCTICA

- Que el alumno pueda conocer la programación en Clips a través de los ejms presentados y especialmente mediante la resolución de los ejercicios propuestos.
- Se espera que pueda servir de ayuda para el desarrollo del prototipo de sistema experto a ser solicitado en próxima Práctica Calificada de laboratorio del presente curso.

3. PREPARATIVOS DE LA PRÁCTICA

Con la finalidad de observar el resultado de las acciones a efectuar, abra las ventanas **Agenda** y **Facts**. Para ello, elija **Facts (MAIN)** y **Agenda (MAIN)** de la opción **Window** que aparece en el menú principal.

4. PROCEDIMIENTOS

REGLAS EN CLIPS

Una regla de producción SI-ENTONCES consta de un antecedente -también denominado parte “SI” o lado izquierdo de la regla (LHS, left hand side) y de un consecuente -también denominado parte “Entonces” o lado derecho de la regla (RHS, right hand side).

LHS => RHS

El antecedente está formado por un conjunto de elementos condicionales (EC) que deben satisfacerse para que la regla sea aplicable. Existe un *and* implícito entre todas las condiciones en la parte izquierda de la regla).

Sintaxis o formato:

```
(defrule <nombre_regla>
[<comentario-opcional>]
  (elemento-condicional 1)
  (elemento-condicional 2)
  ...
  (elemento-condicional N)
=>
  (acción 1)
  (acción 2)
  ...
  (acción M))
```

La cabecera está formada por el nombre de la regla que debe ser único y, opcionalmente, por un comentario entre comillas (en el ejm: “Prueba de comentario”). El LHS de la regla está formado por los elementos declarados antes del símbolo “=>”, llamado flecha. Lo que viene después constituye el RHS de la regla, formado por las acciones que se ejecutan cuando la regla es activada. En el ejemplo caso la acción es imprimir el string “Te llamas Juan Perez Lopez”. Cada uno de los elementos condicionales y las **acciones** de la regla y ésta misma, deben estar encerrados entre paréntesis.

EJM.

```
(defrule nombreJuan
  "Prueba de comentario"
  (nombre Juan)
=>
  (printout t "Tu nombre
de pila es Juan" crlf)
)

(defrule nombreyapellidos
  "Prueba de comentario"
  (nombre Juan)
  (apellido-1 Perez)
  (apellido-2 Lopez)
=>
  (printout t "Te llamas
Juan Perez Lopez" crlf)
)
```

Nota: Si una regla no tiene antecedente, el hecho inicial-*fact* actuará como condicional y activará la regla cada vez que se use *reset*.

Cuando todos los elementos condicionales son satisfechos (es decir, existe una correspondencia con la lista de hechos), se dice que la regla se activa o que hay una activación de la regla. El disparo de una regla activa provoca la ejecución secuencial de sus acciones. Si no aparece elemento condicional alguno, se asume que el antecedente de la regla es el hecho (initial-fact). Las acciones de una regla pueden ser cero, una o más llamadas a comandos y/o funciones.

Estado de una regla

Una regla puede tener los siguientes estados

Activada : Existen hechos que hacen que los elementos condicionales sean cumplidos

Desactivada: No existen tales hechos.

Disparada : Se ejecutan las acciones de una regla si esta está activada y el sistema la elige.

Agenda

Contiene la lista de reglas activadas.

Cargar en el interpretador los constructores

- ❖ Seleccionar en la ventana de edición los constructores a cargar (los `deftemplate`, `deffacts` y `defrules` deseados).
- ❖ Hacer clic derecho sobre la región seleccionada, luego ejecutar `Load Selection` (`Ctrl+K`). Los comandos que no sean para adicionar constructs no son tenidos en cuenta por CLIPS.

ELEMENTOS DE CONDICIÓN EN EL LADO IZQUIERDO DE LAS REGLAS

Existen diversas clases de elementos condicionales. Los más básicos son los llamados *patrones*, que describen mediante constantes, variables y restricciones hechos que pueden estar presentes en la memoria de trabajo.

En las reglas definidas sabemos que:

- Las condiciones o premisas de una regla están implícitamente unidas por la conectiva AND
- El lado izquierdo consta de elementos condicionales que pueden ser de varios tipos:
 1. Un patrón a equiparar
 2. Una disyunción
 3. Comprobar condición (test)
 4. Una negación

A continuación se explicará el elemento condicional: patrón a equiparar

Se explicarán los elementos condicionales más básicos, los llamados *patrones*, que describen mediante constantes, variables y restricciones hechos que pueden estar presentes en la memoria de trabajo.

Un patrón establece una o más restricciones que deben ser satisfechas por algún hecho basado en una plantilla.

Comodines y variables

- Dentro de los patrones los campos se pueden sustituir por:

COMODINES: El caracter ? es empleado como comodín dentro de un patrón en la parte izquierda de las reglas, de manera que cualquier valor en la posición referenciada por dicho caracter dentro del patrón será aceptado. Los comodines pueden ser simples y múltiples; estos se denotan \$?.

EJM.

Tomando como referencia el ítem 5 de la Práctica Guiada de Laboratorio 6 sobre la plantilla *persona*, si se quiere buscar una persona llamada Mario sin importar la edad que tenga, se buscaría la correspondencia con un patrón del tipo:

```
(persona (nombre Marco))  
(edad ?)  
)  
=>
```

en el LI de la respectiva regla.

VARIABLES: Cuando se quiere almacenar el valor de un campo en una variable para usarlo posteriormente; las variables pueden ser simples y múltiples. Los nombres de las variables simples van siempre precedidas del caracter `?`, y los de las variables múltiples, además de dicho caracter, van precedidas del caracter `$`.

EJM.

```
(defrule encontrar-dato  
  (dato ?x azul rojo $?))  
=>
```

Los patrones con variables (simples o múltiples) son semejantes a los patrones con comodín, con la diferencia de que la parte del hecho que se equipara se almacena en la variable.

Suponiendo que un elemento de condición se declarase así:

```
(datos ?x $?y ?)
```

Un hecho que lo equipararía sería:

```
(datos uno dos tres)
```

Las variables almacenarían los sgts valores:

$x \leftarrow \text{uno}$

$y \leftarrow (\text{dos})$

en el caso de la variable `y` se almacenaría en una lista porque se trata de una variable múltiple.

El valor `tres` por su posición sería aceptado al comodín `?` pero éste no almacenaría valor alguno.

PROBLEMA DE LA AGENCIA DE VEHÍCULOS

EL dueño de una agencia de vehículos que se dedica a la compra-venta de carros usados, desea construir un sistema basado en el conocimiento para su sistema de inventario; dicho sistema debe permitir registrar los vehículos que se venden y que se compran en su actividad diaria, y luego efectuar algunas operaciones sobre dichos registros. Con esa finalidad pretende usar la herramienta CLIPS.

Los vehículos que dispone actualmente la agencia, y algunas de sus características, son:

ID	Marca	Color	Año de fabricación	Estado
01	Toyota	verde	2007	bueno
02	Mazda	azul	2010	regular
03	Renault	azul claro	2015	excelente
04	Fiat	rojo	1995	muy malo
05	Toyota	negro	2012	bueno
06	Nissan	gris plata	2003	regular
07	Volkswagen	amarillo	1992	malo

EJERCICIO 1

Defina una plantilla llamada *carro* que tenga cuatro atributos: la marca del vehículo, su color, el año de fabricación y su estado. Use valores numéricos para el Año de fabricación y el rango de los mismos, cadenas para el atributo Marca y símbolos para los otros atributos. Para el atributo Marca defina los valores permitidos, y para los atributos Color y Estado defina valores por defecto. Nota: NO se debe concatenar los valores simbólicos, p.e. ~~azul~~-claro.

EJERCICIO 2

El dueño de la agencia al hacer el inventario quiere saber si hay al menos un carro de marca Fiat. Con ese objetivo defina una regla llamada *ImprimeFiat*. Adicionalmente defina una regla llamada *Saludar* sin antecedentes (LHS esté vacío). Al ejecutar el programa luego de **haber cargado los hechos al entorno**, se debe obtener:

```
CLIPS> (run)
FIRE 1 ImprimeFiat: f-2
En inventario hay al menos un Fiat.
CLIPS>
```

EJERCICIO 3

Ahora la dueña de la agencia desea obtener los colores y los años de fabricación de todos los carros cuya marca sea Toyota. Para tal efecto se debe mandar imprimir por pantalla mediante las sentencias adecuadas, además de la marca, los respectivos valores simbólicos y numéricos.

EJERCICIO 4

Defina una regla llamada *Carros-excelentes* que permita obtener los carros cuyo estado sea *excelente*. para tal efecto se debe mandar imprimir por pantalla un mensaje indicando que hay un carro con ese estado. Adicionalmente la regla debe mandar imprimir por pantalla un mensaje indicando mediante símbolos la marca del carro y su color, y mediante números el año de fabricación.

EJERCICIO 5

El dueño de la agencia quiere conocer los carros registrados de cualquier marca y cuyo color haya sido definido mediante dos símbolos; además desea obtener dichos símbolos c/u en una línea. Defina una regla de modo que el dueño alcance su objetivo. Al ejecutar el programa luego de haber cargado los hechos al entorno, se debe obtener:

```
CLIPS> (run)
Si hay un carro cuyo color se define con 2 terminos:
xxxx
YYYY
CLIPS>
```

Retractación de hechos

Con un patrón se puede almacenar la dirección de un hecho/instancia que satisfacen el elemento condicional en una variable; luego ya se puede retractar el hecho/instancia.

Sintaxis:

```
<variable-simple> <- <hecho>
=>
(retract <identificador-hecho>*)
donde:
<identificador-hecho> := <variable-simple>
```

EJM.

En este problema de la agencia de vehículos, para eliminar de los registros un carro que se ha vendido se utilizaría la sgte regla. Asuma que se ha vendido el Fiat rojo:

Repetimos los pasos de cargar los hechos y la regla. Al ejecutar el programa, se observa que el hecho:

```
(Carro (Marca "Fiat") (Color rojo) (Año 1995) (Estado muy malo))
```

ha sido eliminado de la lista de hechos.

EJERCICIO 6

En este problema de la agencia de vehículos, asuma que el encargado no recuerda las características del carro que desea eliminar de los registros pues ha sido vendido, sólo se acuerda que era de color negro. Con tal finalidad, defina una regla que le permita identificarlo y luego lo retire del listado de carros disponibles; mandar imprimir en pantalla un mensaje indicando que el carro ha sido retirado de los registros. Adicionalmente, debe mandar imprimir en pantalla en una sola línea otro mensaje indicando todas las características del carro a eliminar.

EJERCICIO 7

Se desea ahora eliminar del listado a todos los carros cuyo estado sea malo, independientemente de la marca y del año de fabricación. Antes de la eliminación, agregar a la **Base de Hechos** el sgte hecho:

ID	Marca	Color	Año de fabricación	Estado
08	Lada	blanco	1995	malo
09	Volkswagen	gris	2008	regular
10	Nissan	verde	2010	bueno

Debe mandar imprimir en pantalla un mensaje indicando todas las características (marca, color, año de fabricación y estado) de los carros que han sido retirados de la lista.

5. Instrucciones para el envío de la solución

La solución de la "Práctica Guiada de Laboratorio 7 - IA 2022-1 G1 EPIS" podrá enviarse como respuesta a la tarea del mismo nombre publicada en la clase de Classroom, hasta las 23:59 h del Viernes 22 de Julio del 2022, en un archivo con extensión .pdf o .docx.

El archivo deberá denominarse: solPGL7_IA_G1_nombre-apellidos.pdf.

Todas las preguntas de la Práctica deben responderse en el espacio creado para tal fin en este mismo documento, mencionando explícitamente como subtítulo: "Solución del ejercicio n", donde "n" corresponde al número del ejercicio. Caso se requiera, puede usar un mayor espacio que el creado para las respuestas.

Alternativamente, se puede enviar un archivo con extensión .clp con las respuestas de los ejercicios. Con esa finalidad, se debe indicar claramente como comentario el número de ejercicio al cual está referida la solución presentada.