

Tema 3: Representación de Hechos en CLIPS

Información sobre
Conocimiento del Dominio

Representación de la información

- **Hechos.** Ordenados y no Ordenados. Índice y dirección.
- **Objetos.** POO. Instancias de objetos.
- **Variables globales.** Constructor defglobal.

Hechos: Órdenes de uso

Órdenes de utilización de Hechos:

- assert.
- facts
- retract
- modify
- duplicate
- deftemplate
- deffacts
- reset
- clear

Hechos: Tipos y ejemplos

(nombre-relación campo1 campo2 ...)

- Ordenados

(casa calle-nueva 32)

(perro gato vaca)

(IA 2 Pedro)

- No ordenados (hechos plantillas)

(coche (marca Ford) (modelo focus) (color gris))

(cliente (nombre "Juan Pérez") (tlf 957123456))

Hechos: Ejemplo Ordenados

CLIPS> (assert (libro))

<Fact-0>

CLIPS> (assert (fuente))

<Fact-1>

CLIPS> (facts)

f-0 (libro)

f-1 (fuente)

For a total of 2 facts.

CLIPS>

Hechos: Ejemplo plantillas

```
eje2.clp - Bloc de notas
Archivo Edición Buscar Ayuda

(deftemplate comprador
  (slot nombre)
  (slot edad)
)
(deftemplate vendedor
  (slot nombre)
  (slot edad)
)
(deftemplate casa
  (slot localizacion-casa)
  (slot superficie)
)
(deftemplate vende-casa
  (slot nombre-comprador)
  (slot nombre-vendedor)
  (slot localizacion-casa)
  (slot precio-venta)
  (slot fecha-venta)
)
(deffacts hechos
  (comprador (nombre Juan) (edad 34))
  (vendedor (nombre Pedro) (edad 24))
  (casa (localizacion-casa Avenida-del-Brillante)
    (superficie 300))

  (vende-casa
    (nombre-comprador Juan)
    (nombre-vendedor Pedro)
    (localizacion-casa Avenida-del-Brillante)
    (precio-venta 30.000.000)
    (fecha-venta 12-02-2000))
)
```

Hechos No Ordenados

- El orden en los campos no es importante.

(clase (estudiantes 30) (profesor "Marta Ramírez"))

(clase (profesor "Marta Ramírez") (estudiantes 30))

- Se pueden modificar utilizando las órdenes (modify) y (duplicate).

Hechos Iniciales

- El constructor deffacts.
 - La orden (reset) añade cada hecho especificado con deffacts en la lista de hechos o factlist.
 - También añade el hecho initial-fact.

CLIPS> (reset)

CLIPS> (facts)

f-0 (initial-fact)

For a total of 1 fact.

CLIPS>

Hechos: comandos

(assert <hecho>+)

(facts [<inicio> [<final> [máximo]]])

(retract <índice>+ | *)

(modify <índice> <nueva-casilla>+)

(duplicate <índice> <nueva-casilla>+)

<nueva-casilla>::= (<nombre> <valor>)

Afirmando Hechos

- Orden (assert)
 - Introduce datos en la base de hechos.
(assert (yo))
 - La orden (facts) sirve para ver la base de hechos con formato:
f-índice (hecho)
 - La orden (clear) limpia la base de hechos.
 - La orden (reset) borra hechos e inserta hecho especial (initial-fact).

Afirmando Hechos

```
CLIPS> (assert (a) (b) (c))
```

```
<Fact-2>
```

```
CLIPS> (facts)
```

```
f-0    (a)
```

```
f-1    (b)
```

```
f-2    (c)
```

For a total of 3 facts.

```
CLIPS> (facts 0)
```

```
f-0    (a)
```

```
f-1    (b)
```

```
f-2    (c)
```

For a total of 3 facts.

Afirmando Hechos

CLIPS> (facts 1)

f-1 (b)

f-2 (c)

For a total of 2 facts.

CLIPS> (facts 2)

f-2 (c)

For a total of 1 fact.

CLIPS> (facts 0 1)

f-0 (a)

f-1 (b)

For a total of 2 facts.

CLIPS> (facts 0 2 2)

f-0 (a)

f-1 (b)

For a total of 2 facts.

CLIPS>

Afirmando Hechos

- Orden (assert)
 - No se pueden insertar un hecho que ya existe, excepto utilizando la orden (override-set-fact-duplication).
- ```
CLIPS> (assert (libro))
<Fact-0>
CLIPS> (assert (libro))
FALSE
CLIPS>
```
- Los índices de los hechos no cambian aunque se borren hechos.

# Iniciando la Base de Hechos

- La orden (clear).
  - Elimina todos los hechos de la base de hechos.
  - Reinicializa el índice de hechos a cero.
  - Reestablece CLIPS a su estado original.
  - También elimina la base de conocimiento.

# Iniciando la base de Hechos

```
CLIPS> (assert (a) (b) (c))
```

```
<Fact-2>
```

```
CLIPS> (facts)
```

```
f-0 (a)
```

```
f-1 (b)
```

```
f-2 (c)
```

For a total of 3 facts.

```
CLIPS> (clear)
```

```
CLIPS> (facts)
```

```
CLIPS>
```

# Hechos: Legibilidad

- Utilizar el retorno de carro (intro) después de cada campo.
- Clips reemplaza los retorno de carro y los tabuladores con espacios simples.



# Hechos Legibilidad

CLIPS> (clear)

CLIPS> (assert (lista-comidas  
                  helado  
                  fresas  
                  nata  
                  pescado))

<Fact-0>

CLIPS> (facts)

f-0     (lista-comidas helado fresas nata pescado)

For a total of 1 fact.

CLIPS>

# Hechos Legibilidad

- Los espacios en blanco se usan para separar múltiples campos.
- Cuidado con las mayúsculas y minúsculas que pueden producir hechos distintos.
- Dentro de una “cadena de caracteres” los espacios en blanco si afectan.

# Hechos Legibilidad

CLIPS> (clear)

CLIPS> (assert (es-animal caballo))

<Fact-0>

CLIPS> (assert (es-animal caballo ))

FALSE

CLIPS> (assert (es-animal Caballo ))

<Fact-1>

CLIPS> (facts)

f-0 (es-animal caballo)

f-1 (es-animal Caballo)

For a total of 2 fact.

CLIPS>

# Hechos Legibilidad

CLIPS> (clear)

CLIPS> (assert (es-animal "perro"))

<Fact-0>

CLIPS> (assert (animal-es " perro " ))

<Fact-1>

CLIPS> (facts)

f-0 (animal-es "perro")

f-1 (animal-es " perro " )

For a total of 4 facts.

CLIPS>

# Hechos: estilo

PEOR

(leche pan huevos)

MEJOR

(lista-compra leche pan huevos)

PEOR

(perro) (gato) (vaca)

MEJOR

(es-animal perro)

(es-animal gato)

(es-animal vaca)

(animales perro gato vaca)

(nombre-relación campo1 campo2 ...)

# Retractar Hechos

- La orden (retract) elimina hechos de la base de hechos.
  - Se puede especificar un índice o el propio hecho. (retract índice)
  - No se puede eliminar un hecho ya eliminado.
  - Se puede eliminar múltiples hechos con dos índices. (retract i1 i2).
  - Se pueden eliminar todos los hechos con \*. (retract \*).

# Retractor Hechos

CLIPS> (clear)

CLIPS> (assert (animal-es pato))

<Fact-0>

CLIPS> (assert (animal-sonido quack))

<Fact-1>

CLIPS> (assert (El pato dice "Quack."))

<Fact-2>

CLIPS> (facts)

f-0 (animal-es pato)

f-1 (animal-sonido quack)

f-2 (El pato dice "Quack.")

For a total of 3 facts.

CLIPS>

# Retractar Hechos

CLIPS> (retract 2)

CLIPS> (facts)

f-0 (animal-es pato)

f-1 (animal-sonido quack)

For a total of 2 facts.

CLIPS>

CLIPS> (retract 2)

[PRNTUTIL1] Unable to find fact f-2.

CLIPS>

CLIPS> (retract 1)

CLIPS> (facts)

f-0 (animal-es pato)

For a total of 1 fact.

CLIPS> (retract \*)

CLIPS> (facts)

CLIPS>



# Órdenes de Depuración

- La orden (watch facts).
  - Muestra los hechos que están siendo insertados o eliminados.
    - ==> Hecho entrando en lista de hechos
    - <== Hecho saliendo de lista de hechos

CLIPS> (clear)

CLIPS> (watch facts)

CLIPS> (assert (animal-es pato))

==> f-0 (animal-es pato)

<Fact-0>

CLIPS>

# Órdenes de Depuración

```
CLIPS> (reset)
<== f-0 (animal-es pato)
==> f-0 (initial-fact)
CLIPS> (assert (animal-es pato))
==> f-1 (animal-es pato)
<Fact-1>
CLIPS> (retract 1)
<== f-1 (animal-es pato)
CLIPS> (facts)
f-0 (initial-fact)
For a total of 1 fact.
CLIPS>
```

# Órdenes de Depuración

- La orden (unwatch facts).
  - Desactiva la presentación de hechos.  
CLIPS> (unwatch facts)
- Otros elementos a visualizar:
  - (watch slots).
  - (watch rules).
  - (watch methods).
  - (watch deffunctions).
  - (watch compilations).
  - (watch globals).
  - (watch all).
  - ...

# Hechos: ejemplo

```
CLIPS> (clear)
CLIPS> (assert (color rojo))
<Fact-0>
CLIPS> (assert (color azul) (valor (+ 3 4)))
<Fact-2>
CLIPS> (assert (color rojo))
FALSE
CLIPS> (deftemplate estado (slot temperatura) (slot presion))
CLIPS> (assert (estado (temperatura alta) (presion baja)))
<Fact-3>
CLIPS> (facts)
f-0 (color rojo)
f-1 (color azul)
f-2 (valor 7)
f-3 (estado (temperatura alta) (presion baja))
For a total of 4 facts.
```

# Hechos: ejercicios

- Muestra los hechos con índice  $\geq 1$
- Muestra los hechos 1 a 2
- Crea un nuevo hecho que sea como el 3 pero con (*temperatura baja*)
- Elimina el hecho 1
- Añade un hecho (*color verde*)
- Elimina todos los hechos