

## Tema 5: Constructor de hechos deffacts

### Introducción

- La orden (*deffacts*) es una alternativa a (*assert*) para afirmar hechos.

CLIPS> (clear)

CLIPS> (**deffacts** paseo "Algunos hechos sobre el paseo"

(estado paseando) ;Hecho para ser insertado

(tipo-de-paseo largo)) ;Hecho para ser insertado

CLIPS> (reset) ;Hace que los hechos deffacts sean insertados

CLIPS> (facts)

f-0 (initial-fact)

f-1 (estado paseando)

f-2 (tipo-de-paseo largo)

For a total of 3 facts.

## Sintaxis de deffacts

(**deffacts** <nombre-definición>  
[<comentario>] <hecho>\*)

- Si se duplica el nombre, los hechos del deffacts anterior se borran.
- Pueden existir múltiples constructores deffacts y en cada uno puede afirmar cualquier número de hechos, tanto ordenados como no ordenados.

## Expresiones dinámicas en un hecho

- Expresiones que serán evaluadas y que devolverán un valor que será tomado como el valor de un campo dentro de un hecho.
- Primero (load) y luego (reset).

## Orden (list-deffacts)

- **(list-deffacts)** muestra los hechos definidos con deffacts.

```
CLIPS> (list-deffacts)
initial-facts
comienzo
for a total of 2 deffacts.
CLIPS>
```

## Orden (reset) y (clear)

- **(clear)** reinicializa el entorno CLIPS borrando todos los hechos de la **base de afirmaciones** y todas las reglas de la **base de conocimientos**.
- **(reset)** reinicializa el entorno de CLIPS pero no elimina las reglas de la base de conocimiento. Además establece un hecho inicial el **initial-fact** y afirma todos los hechos **deffacts**.

## Orden (undeffacts)

- **(undeffacts)** suprime los hechos insertados por una orden **(deffacts)**.

CLIPS> (undeffacts comienzo)

CLIPS> (list-deffacts)

Initial-facts

For a total of 1 deffacts.

## Orden (ppdeffacts)

- La orden **(ppdeffacts <nombre>)** muestra la definición de un hecho definido mediante deffacts.

## Ejemplos

```
CLIPS> (clear)
CLIPS> (deffacts EstadoCoche "El estado de mi coche" (coche motor "a
punto") (coche bujias limpias))
CLIPS> (assert (coche neumaticos gastados))
<Fact-0>
CLIPS> (facts)
f-0      (coche neumaticos gastados)
For a total of 1 fact.
CLIPS> (reset)
CLIPS> (facts)
f-0      (initial-fact)
f-1      (coche motor "a punto")
f-2      (coche bujias limpias)
For a total of 3 facts.
CLIPS> (assert (coche luces funcionan))
<Fact-3>
CLIPS> (deffacts EstadoCasa "El estado de mi casa" (casa salon
grande))
```

## Ejemplos (continúa)

```
CLIPS> (facts)
f-0      (initial-fact)
f-1      (coche motor "a punto")
f-2      (coche bujias limpias)
f-3      (coche luces funcionan)
For a total of 4 facts.
CLIPS> (reset)
CLIPS> (facts)
f-0      (initial-fact)
f-1      (coche motor "a punto")
f-2      (coche bujias limpias)
f-3      (casa salon grande)
For a total of 4 facts.
```

## Ejercicio

- Haciendo uso de la plantilla *persona*, cree un conjunto de hechos por defecto, llamado "elenco", que contenga 5 personas diferentes.

```
(deffacts elenco "Actores y actrices"  
  (persona  
    (nombre "Carlos") (apellidos "Lopez  
    Salgado")  
    (edad 25)(color-ojos gris)  
    (altura 1.72)  
  )  
; Otras personas ...  
)
```