

Tema 4: Hechos definidos a partir de plantillas

El constructor *deftemplate*

Plantillas

- Hechos no ordenados.
- Plantillas de conceptos. Similar a las estructuras en C.
- Representan conceptos con sus atributos o relaciones entre conceptos.

Sintaxis del constructor deftemplate

```
(deftemplate      <nombre> [<comentario>]
                  <definición-casilla>*)
```

```
<definición-casilla> ::=      <def-simple> |
                              <def-múltiple>
```

```
<def-simple> ::=  (slot <nombre-casilla>
                  <atributo>*)
```

```
<def-múltiple> ::=      (multislot <nombre-casilla>
                        <atributo>*)
```

```
<atributo> ::=      <atrib-por-defecto> |
                    <atrib-restricción>
```

Ejemplo

```
CLIPS>(deftemplate persona
```

```
  (slot nombre)
```

```
  (slot edad)
```

```
  (multislot direccion))
```

```
CLIPS> (assert      (persona (edad 24))
```

```
          (persona (nombre Juan))
```

```
          (persona (nombre Juan) (edad 24)
```

```
            (direccion Avd Cervantes)))
```

<Fact-2>

```
CLIPS> (facts)
```

```
f-0      (persona (nombre nil) (edad 24) (direccion))
```

```
f-1      (persona (nombre Juan) (edad nil) (direccion))
```

```
f-2      (persona (nombre Juan) (edad 24) (direccion Avd
Cervantes))
```

For a total of 3 facts.

Plantillas

- Si al definir una plantilla se le da el mismo nombre de una ya existente, la plantilla previamente definida desaparecerá, aunque la nueva sea errónea.
- Una plantilla no se puede volver a definir mientras esté siendo usada por un hecho o por un patrón en el antecedente de una regla.

Plantillas

- Una plantilla puede tener cualquier número de campos simples o múltiples.
- Al definir los campos de una plantilla, CLIPS siempre obliga que se indique si va a tratar de un campo monovalor o de un campo multivalor.

Valores por defecto

`<atrib-por-defecto> ::=`

`(default ?DERIVE | ?NONE | <expresión>*) |`

`(default-dynamic <expresión>*)`

`(default ?DERIVE)` equivale a no poner atributo de valor por defecto.

Ejemplo

```
CLIPS> (deftemplate datos
  (slot w (default ?NONE))
  (slot x (default ?DERIVE))
  (slot y (default (gensym*)))
  (slot z (default-dynamic (gensym*))))
CLIPS> (assert (datos))
```

[TMPLTRHS1] Slot w requires a value because of its (default ?NONE) attribute.

```
CLIPS> (assert (datos (w 3))
          (datos (w 4)))
```

<Fact-1>

```
CLIPS> (facts)
f-0      (datos (w 3) (x nil) (y gen1) (z gen2))
f-1      (datos (w 4) (x nil) (y gen1) (z gen3))
```

For a total of 2 facts.

Restricciones

- Se especifican mediante unos atributos de restricción.
- Limitan los valores que pueden tomar los campos.
- Maneras de comprobar el cumplimiento de las restricciones:
 - Estática (por defecto)
 - Dinámica
- Se puede determinar la manera de hacer la comprobación con las funciones *set-static-constraint-checking* y *set-dynamic-constraint-checking*.

Restricciones de tipo

`<atrib-tipo> ::= (type <especificación-tipo>)`

`<especificación-tipo> ::= <tipo>+ | ?VARIABLE`

`<tipo> ::= SYMBOL | STRING | LEXEME |
 INTEGER | FLOAT | NUMBER |
 INSTANCE-NAME | INSTANCE-ADDRESS |
 INSTANCE | EXTERNAL-ADDRESS |
 FACT-ADDRESS`

Ejemplo

```
CLIPS> (deftemplate persona
  (slot nombre (type LEXEME))
  (slot edad (type INTEGER SYMBOL)))
CLIPS> (assert (persona (nombre Juan) (edad 30)))
<Fact-0>
CLIPS> (assert (persona (nombre "Juan") (edad
  treinta)))
<Fact-1>
CLIPS> (assert (persona (nombre Juan) (edad
  20.5)))
```

[CSTRNCHK1] A literal slot value found in the
assert command

does not match the allowed types for slot edad.

Restricciones de constantes permitidas

```
<atrib-permitidos> ::=
    (allowed-symbols <lista-símbolos>) |
    (allowed-strings <lista-cadenas>) |
    (allowed-lexemes <lista-lexemas>) |
    (allowed-integers <lista-enteros>) |
    (allowed-floats <lista-reales>) |
    (allowed-numbers <lista-números>) |
    (allowed-values <lista-valores>)
```

Restricciones de constantes permitidas

$\langle \text{lista-símbolos} \rangle ::= \langle \text{símbolo} \rangle^+ \mid ?\text{VARIABLE}$
 $\langle \text{lista-cadenas} \rangle ::= \langle \text{cadena} \rangle^+ \mid ?\text{VARIABLE}$
 $\langle \text{lista-lexemas} \rangle ::= \langle \text{lexema} \rangle^+ \mid ?\text{VARIABLE}$
 $\langle \text{lista-enteros} \rangle ::= \langle \text{entero} \rangle^+ \mid ?\text{VARIABLE}$
 $\langle \text{lista-reales} \rangle ::= \langle \text{real} \rangle^+ \mid ?\text{VARIABLE}$
 $\langle \text{lista-números} \rangle ::= \langle \text{número} \rangle^+ \mid ?\text{VARIABLE}$
 $\langle \text{lista-valores} \rangle ::= \langle \text{constante} \rangle^+ \mid ?\text{VARIABLE}$

Ejemplo

```
CLIPS> (deftemplate datos
  (slot x (allowed-symbols uno dos))
  (slot y (allowed-values uno dos)))
CLIPS> (assert (datos (x 1) (y uno)))
<Fact-0>
CLIPS> (assert (datos (x 1) (y 1)))
```

[CSTRNCHK1] A literal slot value found in the
assert command

does not match the allowed values for slot y.

Restricciones de rango

`<atrib-rango> ::=`

`(range <especificación-rango>`
 `<especificación-rango>)`

`<especificación-rango> ::= <número> | ?VARIABLE`

Ejemplo

```
CLIPS> (deftemplate calificacion
  (slot alumno (type STRING))
  (slot asignatura (type LEXEME))
  (slot nota (type NUMBER) (range 0 10)))
CLIPS> (assert (calificacion      (alumno "Juan Pérez")
                                (asignatura Lengua)
                                (nota 7.5)))
```

<Fact-0>

```
CLIPS>
```

```
(assert (calificacion      (alumno "José López")
                            (asignatura Lengua)
                            (nota 11)))
```

[CSTRNCHK1] A literal slot value found in the assert command

does not fall in the allowed range 0 to 10 for slot nota.

Restricciones de cardinalidad

`<atrib-cardinalidad> ::=`

`(cardinality <espec-cardinalidad>
<espec-cardinalidad>)`

`<espec-cardinalidad> ::= <entero> | ?VARIABLE`

Ejemplo

```
CLIPS> (deftemplate datos
  (multislot valores (cardinality 2 4)))
CLIPS> (assert (datos (valores 1 2)))
<Fact-0>
CLIPS> (assert (datos (valores 1)))
```

[CSTRNCHK1] Literal slot values found in the assert command

does not satisfy the cardinality restrictions for slot valores.

```
CLIPS> (assert (datos (valores 1 2 3 4 5)))
```

[CSTRNCHK1] Literal slot values found in the assert command

does not satisfy the cardinality restrictions for slot valores.

Comandos relacionados

```
(ppdeftemplate <nombre-plantilla>)
```

```
(list-deftemplates [<nombre-módulo> | *])
```

```
(undeftemplate <nombre-plantilla> | *)
```