

Tema 4: Hechos definidos a partir de plantillas

El constructor deftemplate

Introducción

- Hechos ordenados y no ordenados.
- Plantillas de conceptos. Similaridad con las estructuras.
- Representan conceptos con sus atributos o relaciones entre conceptos.
- Nombre de los campos. slots.

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>*)

Sintaxis del constructor deftemplate

$\langle \text{atributo} \rangle ::= \langle \text{atributo-default} \rangle \mid$

$\langle \text{atributo- constraint} \rangle$

$\langle \text{atributo- default} \rangle ::= (\text{default } ?\text{DERIVE} \mid ?\text{NONE} \mid$

$\langle \text{expression} \rangle^*) \mid (\text{default-dynamic}) \langle \text{expression} \rangle^*)$

Constructor deftemplate

- Si al definir una plantilla se identifica con el mismo nombre de una ya existente, la plantilla previamente definida desaparecerá.
- Una plantilla no se puede volver a definir mientras se este usando por un hecho o por un patrón en el antecedente de una regla.

Número y Tipos de campos en una plantilla

- 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 de tipo simple o univaluado, o un campo de tipo compuesto o multivaluado.

Ejemplo deftemplate

- **Persona.clp**

```
(deftemplate persona "datos de una persona"
  (slot nombre)
  (slot edad)
  (multislot direccion))
```

- Cargar persona.clp y afirmar los hechos:

```
(assert (persona (edad 34)))
(assert (persona (nombre juan)))
(assert (persona (nombre juan) (edad 34)
                (direccion Avda de cervantes) ))
```

Propiedades de los slots

Valor por defecto.

- Propiedad **<default-attribute>** puede ser de dos tipos:
 - Propiedad **default**. ?DERIVE. ?NONE.
 - Propiedad **default-dynamic**.
- Restricciones de los campos por defecto para la comparación de patrones.



Ejemplo propiedades por defecto

- **Dato.clp**

```
(deftemplate dato
```

```
  (slot w (default ?NONE))
```

```
  (slot x (default ?DERIVE))
```

```
  (slot y (default (gensym*) ))
```

```
  (slot z (default-dynamic (gensym*) )) )
```

- Cargar dato.clp y afirmar los hechos:

```
(assert dato)
```

```
(assert (dato (w 3)))
```

```
(assert (dato (w 4)))
```

Propiedades de los slots

- Propiedad type de los slots:
 - SYMBOL.
 - STRING.
 - NUMBER.
 - INTEGER.
 - FLOAT.

Propiedades de los slots

- Propiedad para especificar valores permitidos de un slot.

allowed-symbols	rico pobre
allowed-strings	"Ricardo" "Juan" "Pedro"
allowed-numbers	1 2 3 4.5 -2.01 1.3e-4
allowed-integers	-100 53
allowed-floats	-2.3 1.0 300.00056
allowed-values	"Ricardo" rico 99 1.e9



Ejemplo Propiedades type y allowed

(deftemplate objeto

(slot nombre

(type SYMBOL)

(default ?DERIVE))

(slot peso

(allowed-values ligero pesado)

(default ligero))

(slot contenidos

(type SYMBOL)

(default ?DERIVE)))

Propiedades de los slots

- Propiedad de slot para especificar un rango de valores permitidos.

(deftemplate persona

 (slot nombre

 (type STRING)

 (default ?DERIVE))

 (slot edad

 (type FLOAT)

 (default (* 2.0 3.4))

 (range 0.0 100.0)))

Propiedades de los slots

- Propiedad de slot para restringir los valores de los slots de una plantilla.
- Verificación de las restricciones estáticas (por defecto) o dinámicas.
- Función *set-static-constraint-checking* y *set-dynamic-constraint-checking* para cambiar el chequeo.

Orden (list-deftemplates)

- Lista las plantillas que están siendo usadas por el sistema.

```
CLIPS> (list-deftemplates)
```

```
initial-fact
```

```
persona
```

For a total of 2 deftemplates.

```
CLIPS>
```

Orden (ppdeftemplate)

- La orden (ppdeftemplate nombre) muestra una plantilla ya definida.

The screenshot shows the CLIPS 6.2 application window. The title bar reads 'CLIPS 6.2'. The menu bar includes 'File', 'Edit', 'Buffer', 'Execution', 'Browse', 'Window', and 'Help'. Below the menu is a toolbar with icons for file operations and help. The main area is a 'Dialog Window' containing a text input field and a text area. The text area displays the command '(ppdeftemplate persona)' entered at the 'CLIPS>' prompt, followed by the definition of the 'persona' template. The definition includes slots for 'nombre' (type STRING) and 'edad' (type FLOAT), both with default values, and a random number generator function.

```
CLIPS> (ppdeftemplate persona)
(deftemplate MAIN::persona
  (slot nombre (type STRING) (default ?DERIVE))
  (slot edad (type FLOAT) (default (* 2.0 3.4)) (ran
CLIPS>
```


Orden (undeftemplate)

- La orden (undeftemplate nombre) elimina una plantilla existente.

CLISP> (undeftemplate persona)

CLIPS> (list-deftemplates)

initial-fact

For a total of 1 deftemplates.

CLIPS>