

Tema 7: Variables globales

El constructor *defglobal*

Introducción

- *defglobal* define variables globales.
- Las variables globales pueden ser referenciadas desde el entorno de CLIPS.
- Pueden contener valores monocampo o multicampo.
- Su valor se cambia con la función *bind*.
- Se reinicializa su valor con *reset* o con *bind* sin valor.
- Para eliminarlas se utiliza la orden *clear* o *undefglobal*.

Sintaxis

```
(defglobal [<módulo>]<asignación-global>*)
```

```
<asignación-global> ::=
```

```
    <variable-global> = <expresión>
```

```
<variable-global> ::= ?*<símbolo>*
```

En la asignación global, es necesario dejar espacios en blanco entre el nombre de la variable, el símbolo de asignación y la expresión.

Ejemplo

```
(defglobal
  ?*x* = 3
  ?*y* = (create$ 1 2 3))
```

```
(def facts datos
  (dato 34)
  (dato 32)
  (dato 9))
```

```
(defrule comprueba
  (dato ?p & (= ?p (* 3 ?*x*)))
=> )
```

Diferencias con v. locales

- Las variables globales pueden usarse en cualquier lugar en que se pueda usar una variable local, salvo dos excepciones:
 - No pueden usarse como parámetros para una función (constructores *deffunction* y *defmethod*).
 - No pueden usarse de la misma manera que una variable local en la parte izquierda del antecedente para asignarles un valor.

Ejemplo

```
(defrule ejemplo1 ;incorrecto
  (dato ?*x*) => )
```

```
(defrule ejemplo2 ;correcto
  (dato ?y&:(= ?y ?*x*)) => )
```

Atención

- Si cambia el valor de una variable global, eso no hace que se active el ciclo de ejecución de las reglas, es decir, la regla en la que se haga referencia a esa variable no se actualiza.

Comandos relacionados defglobal

```
(ppdefglobal <nombre-var-global>)
```

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

```
(undefglobal <nombre-var-global> | *)
```