Paradigmas Fundamentales de Programación

El modelo de computación con estado

Juan Francisco Díaz Frias

Maestría en Ingeniería, Énfasis en Ingeniería de Sistemas y Computación
Escuela de Ingeniería de Sistemas y Computación,
home page: http://eisc.univalle.edu.co
Universidad del Valle - Cali, Colombia

Modelos y Paradigmas de Programación

Universidad del Valle

Plan

1 El concepto de celda para implementar estado explícito

2 Extensión del modelo declarativo con celdas
- Las nuevas declaraciones
- Semántica de las nuevas declaraciones
- Cuidado con la igualdad

Modelos y Paradigmas de Programación

Universidad del Valle

Plan

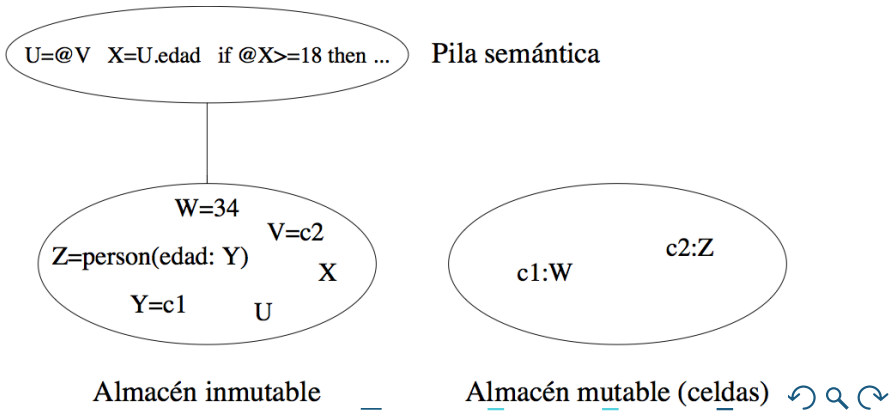
1 El concepto de celda para implementar estado explícito

2 Extensión del modelo declarativo con celdas
- Las nuevas declaraciones
- Semántica de las nuevas declaraciones
- Cuidado con la igualdad

El modelo de computación con estado (1)

Estado explícito: idea

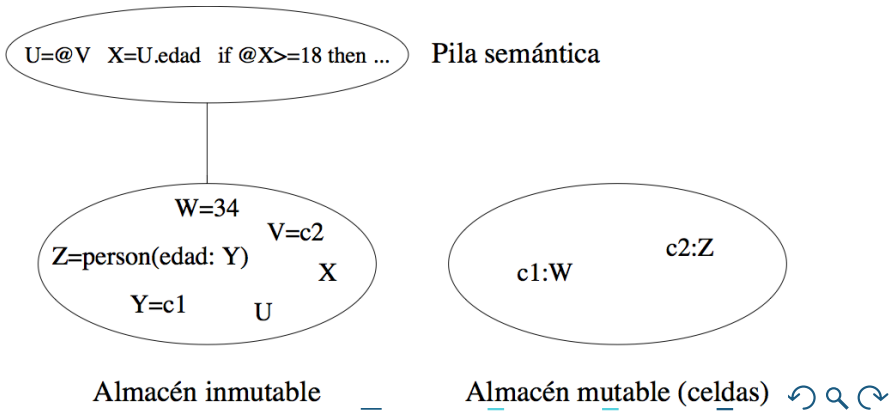
- El estado explícito es una pareja de dos entidades del lenguaje: la identidad del estado y el contenido actual del estado.
- Existe una operación que dada la identidad del estado devuelve el contenido actual.
- Hay una asociación amplia entre las identidades de los estados y todas las entidades del lenguaje.
- ¡Esta asociación se puede modificar!, pero ninguna de las dos entidades del lenguaje se modifica; la asociación es la única que cambia.



El modelo de computación con estado (1)

Estado explícito: idea

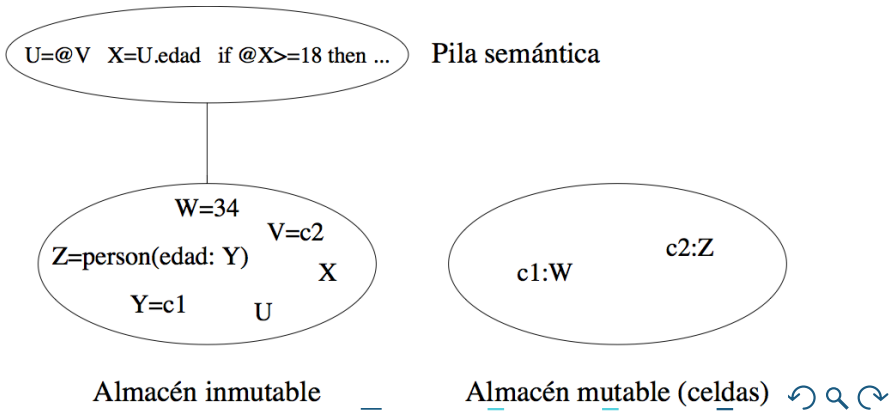
- El estado explícito es una pareja de dos entidades del lenguaje: la identidad del estado y el contenido actual del estado.
- Existe una operación que dada la identidad del estado devuelve el contenido actual.
- Hay una asociación amplia entre las identidades de los estados y todas las entidades del lenguaje.
- ¡Esta asociación se puede modificar!, pero ninguna de las dos entidades del lenguaje se modifica; la asociación es la única que cambia.



El modelo de computación con estado (1)

Estado explícito: idea

- El estado explícito es una pareja de dos entidades del lenguaje: la identidad del estado y el contenido actual del estado.
- Existe una operación que dada la identidad del estado devuelve el contenido actual.
- Hay una asociación amplia entre las identidades de los estados y todas las entidades del lenguaje.
- ¡Esta asociación se puede modificar!, pero ninguna de las dos entidades del lenguaje se modifica; la asociación es la única que cambia.



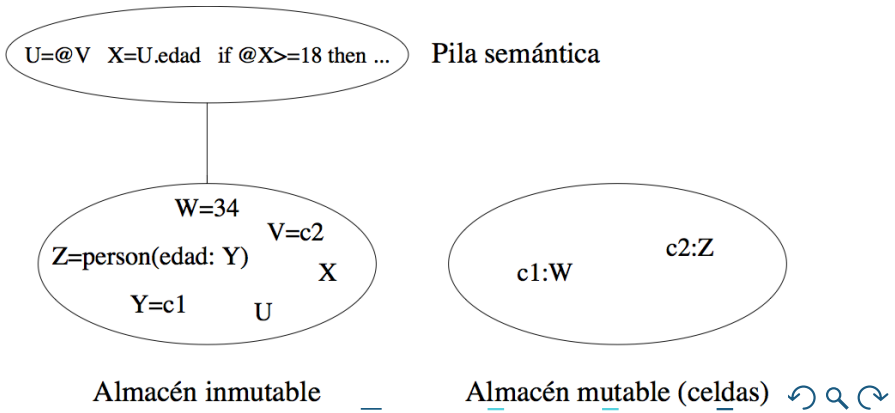
El modelo de computación con estado (1)

Estado explícito: idea

- El estado explícito es una pareja de dos entidades del lenguaje: la identidad del estado y el contenido actual del estado.
- Existe una operación que dada la identidad del estado devuelve el contenido actual.
- Hay una asociación amplia entre las identidades de los estados y todas las entidades del lenguaje.
- ¡Esta asociación se puede modificar!, pero ninguna de las dos entidades del lenguaje se modifica; la asociación es la única que cambia.

Estado explícito: implementación

- Celda: nuevo tipo básico del modelo de computación.
- Una celda es una pareja: (valor de tipo nombre, referencia al almacén de asignación única).
- El conjunto de todas las celdas vive en el almacén mutable.



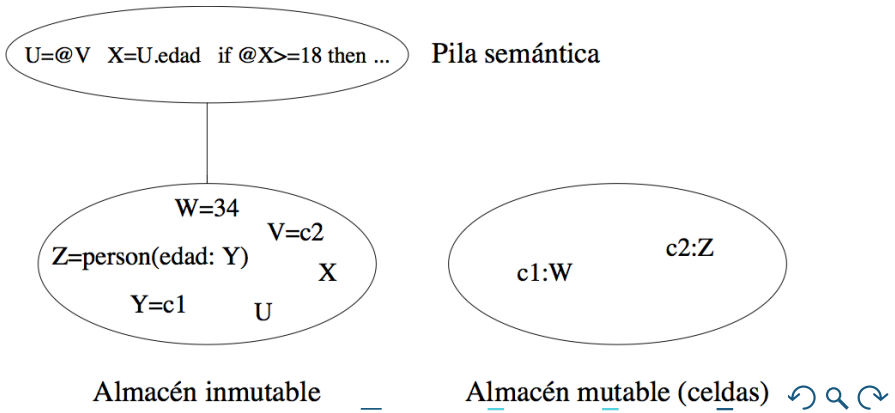
El modelo de computación con estado (1)

Estado explícito: idea

- El estado explícito es una pareja de dos entidades del lenguaje: la identidad del estado y el contenido actual del estado.
- Existe una operación que dada la identidad del estado devuelve el contenido actual.
- Hay una asociación amplia entre las identidades de los estados y todas las entidades del lenguaje.
- ¡Esta asociación se puede modificar!, pero ninguna de las dos entidades del lenguaje se modifica; la asociación es la única que cambia.

Estado explícito: implementación

- Celda: nuevo tipo básico del modelo de computación.
- Una celda es una pareja: (valor de tipo nombre, referencia al almacén de asignación única).
- El conjunto de todas las celdas vive en el almacén mutable.



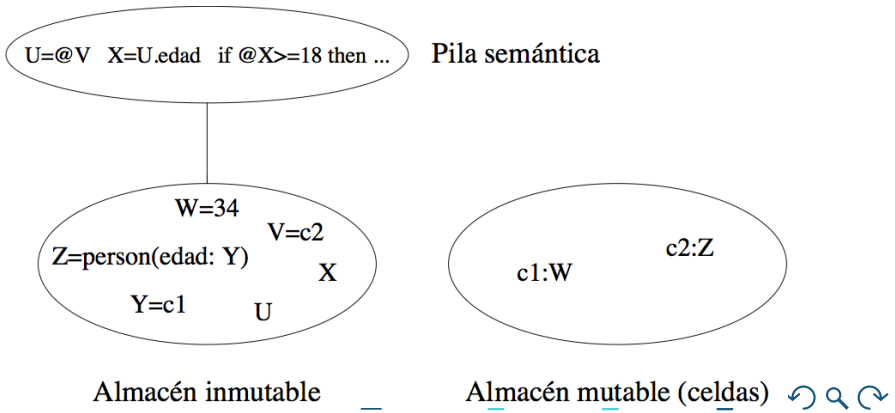
El modelo de computación con estado (1)

Estado explícito: idea

- El estado explícito es una pareja de dos entidades del lenguaje: la identidad del estado y el contenido actual del estado.
- Existe una operación que dada la identidad del estado devuelve el contenido actual.
- Hay una asociación amplia entre las identidades de los estados y todas las entidades del lenguaje.
- ¡Esta asociación se puede modificar!, pero ninguna de las dos entidades del lenguaje se modifica; la asociación es la única que cambia.

Estado explícito: implementación

- Celda: nuevo tipo básico del modelo de computación.
- Una celda es una pareja: (valor de tipo nombre, referencia al almacén de asignación única).
- El conjunto de todas las celdas vive en el almacén mutable.



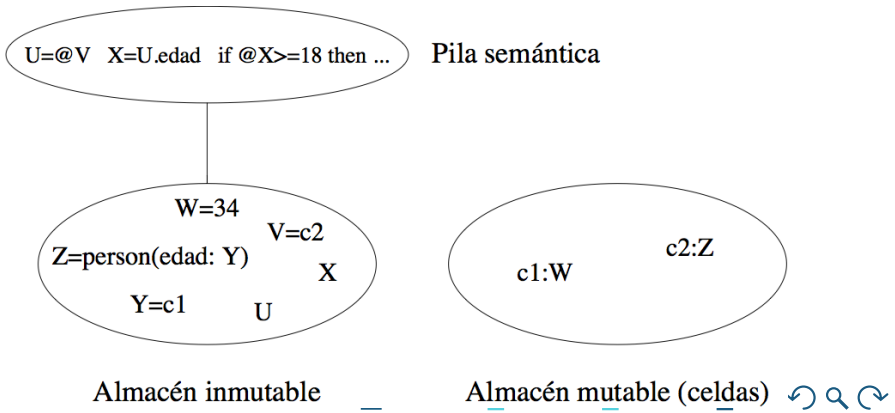
El modelo de computación con estado (1)

Estado explícito: idea

- El estado explícito es una pareja de dos entidades del lenguaje: la identidad del estado y el contenido actual del estado.
- Existe una operación que dada la identidad del estado devuelve el contenido actual.
- Hay una asociación amplia entre las identidades de los estados y todas las entidades del lenguaje.
- ¡Esta asociación se puede modificar!, pero ninguna de las dos entidades del lenguaje se modifica; la asociación es la única que cambia.

Estado explícito: implementación

- Celda: nuevo tipo básico del modelo de computación.
- Una celda es una pareja: (valor de tipo nombre, referencia al almacén de asignación única).
- El conjunto de todas las celdas vive en el almacén mutable.



Plan

1 El concepto de celda para implementar estado explícito

2 Extensión del modelo declarativo con celdas
- Las nuevas declaraciones
- Semántica de las nuevas declaraciones
- Cuidado con la igualdad

El modelo de computación con estado (2)

El modelo declarativo con estado explícito

$\langle d \rangle$ ::=	
skip	Declaración vacía
$\langle d \rangle_1\ \langle d \rangle_2$	Declaración de secuencia
local $\langle x \rangle$ **in** $\langle d \rangle$ **end**	Creación de variable
$\langle x \rangle_1 = \langle x \rangle_2$	Ligadura variable-variable
$\langle x \rangle = \langle v \rangle$	Creación de valor
if $\langle x \rangle$ **then** $\langle d \rangle_1$ **else** $\langle d \rangle_2$ **end**	Condicional
case $\langle x \rangle$ **of** $\langle$patrón$\rangle$ **then** $\langle d \rangle_1$ **else** $\langle d \rangle_2$ **end**	Reconocimiento de patrones
$\{\langle x \rangle\ \langle y \rangle_1 \cdots \langle y \rangle_n\}$	Invocación de procedimiento
{NewName $\langle x \rangle$}	Creación de nombre
{NewCell $\langle x \rangle$ $\langle y \rangle$}	**Creación de celda**
{Exchange $\langle x \rangle$ $\langle y \rangle$ $\langle z \rangle$}	**Intercambio de celda**

El modelo de computación con estado (3)

Operaciones sobre celdas

Operación	Descripción
`{NewCell X C}`	Crea una celda nueva `C` con contenido inicial `X`.
`{Exchange C X Y}`	Liga atómicamente `X` con el contenido antiguo de la celda `C` y hace que `Y` sea el contenido nuevo.
`X=@C`	Liga `X` con el contenido actual de la celda `C`.
`C:=X`	Coloca a `X` como el contenido nuevo de la celda `C`.

Sólo dos nuevas declaraciones

- `{NewCell X C}`
- `{Exchange C X Y}`

Azúcar sintáctico

- `X=@C`
- `C:=X`

El modelo de computación con estado (3)

Operaciones sobre celdas

Operación	Descripción
`{NewCell X C}`	Crea una celda nueva `C` con contenido inicial `X`.
`{Exchange C X Y}`	Liga atómicamente `X` con el contenido antiguo de la celda `C` y hace que `Y` sea el contenido nuevo.
`X=@C`	Liga `X` con el contenido actual de la celda `C`.
`C:=X`	Coloca a `X` como el contenido nuevo de la celda `C`.

Sólo dos nuevas declaraciones

- `{NewCell X C}`
- `{Exchange C X Y}`

Azúcar sintáctico

- `X=@C`
- `C:=X`

El modelo de computación con estado (3)

Operaciones sobre celdas

Operación	Descripción
`{NewCell X C}`	Crea una celda nueva `C` con contenido inicial `X`.
`{Exchange C X Y}`	Liga atómicamente `X` con el contenido antiguo de la celda `C` y hace que `Y` sea el contenido nuevo.
`X=@C`	Liga `X` con el contenido actual de la celda `C`.
`C:=X`	Coloca a `X` como el contenido nuevo de la celda `C`.

Sólo dos nuevas declaraciones

- `{NewCell X C}`
- `{Exchange C X Y}`

Azúcar sintáctico

- `X=@C`
- `C:=X`

Plan

1 El concepto de celda para implementar estado explícito

2 Extensión del modelo declarativo con celdas
- Las nuevas declaraciones
- Semántica de las nuevas declaraciones
- Cuidado con la igualdad

El modelo de computación con estado (4)

Semántica de las celdas

- Añadimos un almacén nuevo μ: el almacén mutable.
- μ contiene celdas: parejas de la forma $x : y$, donde x y y son variables del almacén de asignación única.
- Inicialmente, μ está vacío.
- x siempre estará ligado a un valor de tipo nombre que representa una celda.
- y puede estar ligado a cualquier valor parcial.
- Estado de ejecución: (MST, σ, μ)

Semántica $(\{\texttt{NewCell}\ \langle x \rangle\ \langle y \rangle\}, E)$

- Crear un nombre de celda fresco n.
- Ligar $E(\langle y \rangle)$ y n en el almacén.
- Si la ligadura tiene éxito, agregar $E(\langle y \rangle) : E(\langle x \rangle)$ a μ.
- Si la ligadura falla, lanzar una condición de error.

El modelo de computación con estado (4)

Semántica de las celdas

- Añadimos un almacén nuevo μ: el almacén mutable.
- μ contiene celdas: parejas de la forma $x : y$, donde x y y son variables del almacén de asignación única.
- Inicialmente, μ está vacío.
- x siempre estará ligado a un valor de tipo nombre que representa una celda.
- y puede estar ligado a cualquier valor parcial.
- Estado de ejecución: (MST, σ, μ)

Semántica $\left(\texttt{\{NewCell}\ \langle x \rangle\ \langle y \rangle\texttt{\}}, E\right)$

- Crear un nombre de celda fresco n.
- Ligar $E(\langle y \rangle)$ y n en el almacén.
- Si la ligadura tiene éxito, agregar $E(\langle y \rangle) : E(\langle x \rangle)$ a μ.
- Si la ligadura falla, lanzar una condición de error.

El modelo de computación con estado (4)

Semántica de las celdas

- Añadimos un almacén nuevo μ: el almacén mutable.
- μ contiene celdas: parejas de la forma $x : y$, donde x y y son variables del almacén de asignación única.
- Inicialmente, μ está vacío.
- x siempre estará ligado a un valor de tipo nombre que representa una celda.
- y puede estar ligado a cualquier valor parcial.
- Estado de ejecución: (MST, σ, μ)

Semántica $\left(\{\texttt{NewCell}\ \langle x \rangle\ \langle y \rangle\}, E\right)$

- Crear un nombre de celda fresco n.
- Ligar $E(\langle y \rangle)$ y n en el almacén.
- Si la ligadura tiene éxito, agregar $E(\langle y \rangle) : E(\langle x \rangle)$ a μ.
- Si la ligadura falla, lanzar una condición de error.

El modelo de computación con estado (4)

Semántica de las celdas

- Añadimos un almacén nuevo μ: el almacén mutable.
- μ contiene celdas: parejas de la forma $x : y$, donde x y y son variables del almacén de asignación única.
- Inicialmente, μ está vacío.
- x siempre estará ligado a un valor de tipo nombre que representa una celda.
- y puede estar ligado a cualquier valor parcial.
- Estado de ejecución: (MST, σ, μ)

Semántica $\left(\{\texttt{NewCell}\ \langle x \rangle\ \langle y \rangle\}, E\right)$

- Crear un nombre de celda fresco n.
- Ligar $E(\langle y \rangle)$ y n en el almacén.
- Si la ligadura tiene éxito, agregar $E(\langle y \rangle) : E(\langle x \rangle)$ a μ.
- Si la ligadura falla, lanzar una condición de error.

El modelo de computación con estado (4)

Semántica de las celdas

- Añadimos un almacén nuevo μ: el almacén mutable.
- μ contiene celdas: parejas de la forma $x : y$, donde x y y son variables del almacén de asignación única.
- Inicialmente, μ está vacío.
- x siempre estará ligado a un valor de tipo nombre que representa una celda.
- y puede estar ligado a cualquier valor parcial.
- Estado de ejecución: (MST, σ, μ)

Semántica $(\texttt{\{NewCell}\ \langle x \rangle\ \langle y \rangle\texttt{\}}, E)$

- Crear un nombre de celda fresco n.
- Ligar $E(\langle y \rangle)$ y n en el almacén.
- Si la ligadura tiene éxito, agregar $E(\langle y \rangle) : E(\langle x \rangle)$ a μ.
- Si la ligadura falla, lanzar una condición de error.

El modelo de computación con estado (5)

Semántica de ({Exchange ⟨x⟩ ⟨y⟩ ⟨z⟩}, E)

- Si la condición de activación es cierta ($E(\langle x \rangle)$ está determinada), entonces realizar las acciones siguientes:
  - Si $E(\langle x \rangle)$ no está ligada al nombre de una celda, entonces lanzar una condición de error.
  - Si μ contiene $E(\langle x \rangle) : w$, entonces:
    - Actualizar μ con $E(\langle x \rangle) : E(\langle z \rangle)$.
    - Ligar $E(\langle y \rangle)$ y w en σ.
- Si la condición de activación es falsa, entonces suspender la ejecución.

Componentes con estado con comportamiento declarativo

Es una buena costumbre de diseño el escribir componentes con estado de manera que se puedan comportar declarativamente:

```
fun {Reverse Xs}
   Rs={NewCell nil}
in
   for X in Xs do
      Rs := X|@Rs
   end
   @Rs
end
```

El modelo de computación con estado (5)

Semántica de $(\{\texttt{Exchange}\ \langle x\rangle\ \langle y\rangle\ \langle z\rangle\}, E)$

- Si la condición de activación es cierta ($E(\langle x\rangle)$ está determinada), entonces realizar las acciones siguientes:
  - Si $E(\langle x\rangle)$ no está ligada al nombre de una celda, entonces lanzar una condición de error.
  - Si μ contiene $E(\langle x\rangle) : w$, entonces:
    - Actualizar μ con $E(\langle x\rangle) : E(\langle z\rangle)$.
    - Ligar $E(\langle y\rangle)$ y w en σ.
- Si la condición de activación es falsa, entonces suspender la ejecución.

Componentes con estado con comportamiento declarativo

Es una buena costumbre de diseño el escribir componentes con estado de manera que se puedan comportar declarativamente:

```
fun {Reverse Xs}
   Rs={NewCell nil}
in
   for X in Xs do
      Rs := X|@Rs
   end
   @Rs
end
```

Plan

1 El concepto de celda para implementar estado explícito

2 Extensión del modelo declarativo con celdas
- Las nuevas declaraciones
- Semántica de las nuevas declaraciones
- Cuidado con la igualdad

El modelo de computación con estado (6)

Referencias compartidas

- `X` es un alias de `Y` si referencian la misma celda.
- Cambiar el contenido de una celda, cambia el de todos sus alias.

```
declare X Y
X={NewCell 0}
Y=X
Y:=10
{Browse @X}
```

Igualdad

- Hasta ahora: igualdad estructural. Dos valores son iguales si tienen la misma estructura.

```
X=persona(edad:25
nombre:"Jorge")
Y=persona(edad:25
nombre:"Jorge")
{Browse X==Y}
```

- Ahora: igualdad de lexemas. ¿Dos celdas no son iguales porque tengan el mismo contenido sino porque son la misma celda?

```
X={NewCell 10}
Y={NewCell 10}
{Browse X==Y} % false
{Browse @X==@Y} % true
```

El modelo de computación con estado (6)

Referencias compartidas

- `X` es un alias de `Y` si referencian la misma celda.
- Cambiar el contenido de una celda, cambia el de todos sus alias.

```
declare X Y
X={NewCell 0}
Y=X
Y:=10
{Browse @X}
```

Igualdad

- Hasta ahora: igualdad estructural. Dos valores son iguales si tienen la misma estructura.

```
 X=persona(edad:25
nombre:"Jorge")
 Y=persona(edad:25
nombre:"Jorge")
 {Browse X==Y}
```

- Ahora: igualdad de lexemas. ¡Dos celdas no son iguales porque tengan el mismo contenido sino porque son la misma celda!

```
X={NewCell 10}
Y={NewCell 10}
{Browse X==Y} % false
{Browse @X==@Y} %true
```

El modelo de computación con estado (6)

Referencias compartidas

- `X` es un alias de `Y` si referencian la misma celda.
- Cambiar el contenido de una celda, cambia el de todos sus alias.

```
declare X Y
X={NewCell 0}
Y=X
Y:=10
{Browse @X}
```

Igualdad

- Hasta ahora: igualdad estructural. Dos valores son iguales si tienen la misma estructura.

```
 X=persona(edad:25
nombre:"Jorge")
 Y=persona(edad:25
nombre:"Jorge")
 {Browse X==Y}
```

- Ahora: igualdad de lexemas. ¡Dos celdas no son iguales porque tengan el mismo contenido sino porque son la misma celda!

```
 X={NewCell 10}
 Y={NewCell 10}
 {Browse X==Y} % false
 {Browse @X==@Y} %true
```