

1 Registro

1.1 Interfaz

se explica con REGISTRO

usa nat, dato, campo, conjTrie, string, itConj

géneros reg, itReg

Operaciones

CAMPOS(**in** $r : \text{reg}$) $\longrightarrow res : \text{itConj}(\text{campo})$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \text{alias}(res, \text{claves}(r))\}$

Descripción: Devuelve un conjunto de campos que son claves del registro ingresado por parámetro

Complejidad: $O(1)$

Aliasing: Se devuelve un iterador al conjunto, hay aliasing

BORRAR?(**in** $\text{crit} : \text{reg}$, **in** $r : \text{reg}$) $\longrightarrow res : \text{bool}$

Pre $\equiv \{\# \text{campos}(\text{crit}) = 1\}$

Post $\equiv \{res =_{\text{obs}} \text{borrar?}(\text{crit}, r)\}$

Descripción: Devuelve true si y solo si todos los campos de crit pertenecen a campos de r.

Complejidad: $O((1))$

AGCAMPOS(**in/out** $r_1 : \text{reg}$, **in** $r_2 : \text{reg}$) $\longrightarrow res : \text{itReg}$

Pre $\equiv \{r_1 =_{\text{obs}} r_{10}\}$

Post $\equiv \{r_1 =_{\text{obs}} \text{agregarCampos}(r_{10}, r_2)\}$

Descripción: agrega los datos de los campos faltantes de r2 a los campos de r1

Complejidad: $O((1))$

COPIARCAMPOS(**in/out** $r_1 : \text{reg}$, **in** $cc : \text{conj}(\text{campo})$, **in** $r_2 : \text{reg}$)

Pre $\equiv \{r_1 =_{\text{obs}} r_{10} \wedge cc \in \text{campos}(r_2)\}$

Post $\equiv \{r_1 =_{\text{obs}} \text{copiarCampos}(cc, r_{10}, r_2)\}$

Descripción: copia los datos que se encuentran en los campos de r2 a r1

Complejidad: $O(1)$

COINCIDEALGUNO(**in** $r_1 : \text{reg}$, **in** $cc : \text{conj}(\text{campo})$, **in** $r_2 : \text{reg}$) $\longrightarrow res$

: bool

Pre $\equiv \{cc \subseteq \text{campos}(r_1) \cap \text{campos}(r_2)\}$

Post $\equiv \{res =_{\text{obs}} \text{coincideAlguno}(r_1, r_2)\}$

Descripción: Devuelve true si y solo si alguno de los campos(dato) de cc pertenece a r1 y r2

Complejidad: $O(1)$

COINCIDENTODOS(**in** $r_1 : \text{reg}$, **in** $cc : \text{conj}(\text{campo})$, **in** $r_2 : \text{reg}$) $\longrightarrow res$

: bool

Pre $\equiv \{cc \subseteq \text{campos}(r_1) \cap \text{campos}(r_2)\}$

Post $\equiv \{res =_{\text{obs}} \text{coincidenTodos}(r_1, cc, r_2)\}$

Descripción: Devuelve true si y solo si todos los campos(dato) de cc pertenecen a r1 y r2

Complejidad: $O(1)$

ENTODOS(**in** $c : \text{campo}$, **in** $cr : \text{conj}(\text{registro})$) $\longrightarrow res : \text{bool}$

1.2 Representación

se representa con `DiccTrie`

donde `DiccTrie` es tupla $\langle \text{claves} : \text{lista}(\text{campo}), \text{significado} : \text{DiccTrie}(\text{campo}, \text{dato}) \rangle$ **1.2.1 Invariante de representación**

$$\text{diccd} \# \text{claves}(\text{secuADicc}(d.\text{claves})) = \text{long}(d.\text{claves}) \wedge \text{long}(d.\text{claves}) = \text{long}(d.\text{significados})$$

1.2.2 Función de abstracción

```
diccreg if vacía?(d.claves) then  
    vacío  
else  
    definir(prim(d).claves, prim(d).significado, Abs(fin(d)))  
fi
```

1.3 Algoritmos

CAMPOS(in $r : \text{reg}$) $\longrightarrow res : \text{Conj}(\text{campo})$	
$cn \leftarrow \text{vacio}();$	$O(1)$
$it \leftarrow \text{crearItDicc}(r);$	
while haySiguiente(it) do ;	$O(1)$
$\text{agregar}(cn, \text{clave.siguiente}(it));$	$O(1)$
$\text{avanzar}(it);$	$O(1)$
end while	
$res \leftarrow cn;$	$O(1)$
	<hr/>
	$O(1)$
BORRAR?(in $crit : \text{reg}$ in $r : \text{reg}$) $\longrightarrow res : \text{bool}$	
$cc \leftarrow \text{crit.campos};$	$O(1)$
$res \leftarrow \text{coincidenTodos}(crit, cc, r);$	$O(1)$
	<hr/>
	$O(1)$
AGREGARCAMPOS(in $r_1 : \text{reg}$ in $r_2 : \text{reg}$) $\longrightarrow res :$	
$cc \leftarrow r_2.campos / r_1.campos;$	$O(1)$
$res \leftarrow \text{copiarCampos}(cc, r_2, r_1);$	$O()$
	<hr/>
	$O(1)$
COPIARCAMPOS(in $cc : \text{conj}(\text{campo})$ in/out $r1 : \text{reg}$, in $r2 : \text{reg}$)	
$itcc \leftarrow \text{crearItConj}(cc);$	$O(1)$
while haySiguiente(itcc) do ;	$O(1)$
$\text{definirRapido}(\text{siguiente}(itcc), \text{obtener}(\text{siguiente}(itcc) r2), r1);$	$O(1)$
$\text{avanzar}(itcc);$	$O(1)$
end while	
	<hr/>
	$O(1)$
COINCIDEALGUNO(in $r1 : \text{reg}$ in $cc : \text{conj}(\text{campo})$ in $r2 : \text{reg}$) $\longrightarrow res : \text{bool}$	
$res \leftarrow \text{false};$	$O(1)$
if $\neg \text{vacio?}(cc)$ then ;	$O(1)$
$itcc \leftarrow \text{crearItConj}(cc);$	$O(1)$
while haySiguiente(itcc) do ; $O(1)$	
if $\text{obtener}(\text{siguiente}(itcc), r1) == \text{obtener}(\text{siguiente}(itcc), r2)$ then ;	
$res \leftarrow \text{true};$	
end if	
$\text{avanzar}(itcc);$	
end while	
	<hr/>
	$O(1)$
COINCIDENTODOS(in $r1 : \text{reg}$ in $cc : \text{conj}(\text{campo})$ in $r2 : \text{reg}$) $\longrightarrow res : \text{bool}$	
$res \leftarrow \text{true};$	$O(1)$
$itcc \leftarrow \text{crearItConj}(cc);$	$O(1)$

```

while haySiguiente(itcc)  $\wedge$  res do; O(1)
  if  $\neg$ obtener(siguiente(itcc), r1) == obtener(siguiente(itcr), r2) then;
    res  $\leftarrow$  false;
  end if
  avanzar(itcc);
end while=0

```

O(1)

```

ENTODOS(in c : campo in cr: conj(reg))  $\longrightarrow$  res : bool
  res  $\leftarrow$  True;
  itcr  $\leftarrow$  CrearItConj(cr);

```

```

while haySiguiente(itcr)  $\wedge$  res do; O(#(cr))
  if  $\neg$ pertenece(c,siguiente(itcr)) then;
    res  $\leftarrow$  false
  end if
end while

```

O(#(cr))

```

COMBINARTODOS(in c : campo in r1: reg in cr: conj(reg))
  itcr  $\leftarrow$  CrearItConjDic(cr);

```

```

while haySiguiente(itcr)  $\wedge$  res do; O(#(cr))
  if obtener(c,siguiente(itcr)) == obtener(c,r) then;
    agregarCampos(r, siguiente(itcr))
  end if
end while

```

O(#(cr))