1. TAD Registro

TAD Registro

igualdad observacional

$$\left(\begin{array}{l}(\forall\ r_1,r_2{:}\text{reg})\ (r_1 =_{\text{obs}} r_2) \Leftrightarrow \left(\begin{array}{l}\text{campos}(r_1) =_{\text{obs}} \text{campos}(r_2) \wedge_{\text{L}} \Big((\forall\ \text{c:Campos})\\ \Big((\text{c} \in \text{campos}(r_1))) \Rightarrow_{\text{L}} \big((\text{Nat} = \text{tipo(c)} \Rightarrow_{\text{L}} (\text{ValorEn}(r_1,\text{c}) =_{\text{obs}} \text{ValorEn}(r_2,\text{c}))) \wedge\\ (\neg\text{Nat} = \text{tipo(c)} \Rightarrow_{\text{L}} (\text{PalabraEn}(r_1,\text{c}) =_{\text{obs}} \text{PalabraEn}(r_2,\text{c})))\big)\Big)\end{array}\right)\end{array}\right)$$

géneros reg

exporta reg, generadores, observadores, otras operaciones

usa Nat, String, Campo, Tipo, Conjunto(α)

observadores básicos

Campos : reg $\longrightarrow$ conj(Campo)
valorEn : reg r $\times$ campo c $\longrightarrow$ Nat $\quad\{\text{Nat} = \text{tipo(c)} \wedge \text{c} \in \text{Campos(r)}\}$
palabraEn : reg r $\times$ campo c $\longrightarrow$ String $\quad\{\text{String} = \text{tipo(c)} \wedge \text{c} \in \text{Campos(r)}\}$

generadores

NuevoRegistro : $\bullet$ $\longrightarrow$ reg
agValor : reg r $\times$ campo c $\times$ nat n $\longrightarrow$ reg $\quad\{\text{c} \notin \text{campos(r)} \wedge \text{Nat} = \text{tipo(c)}\}$
agPalabra : reg r $\times$ campo c $\times$ string s $\longrightarrow$ reg $\quad\{\text{c} \notin \text{campos(r)} \wedge \text{String} = \text{tipo(c)}\}$

otras operaciones

$\bullet = \bullet$: reg $\times$ reg $\longrightarrow$ bool
Coincide? : conj(reg) rs $\times$ reg r $\times$ campo c $\longrightarrow$ bool $\quad\{(\forall\ r_1 : \text{reg})\ r_1 \in rs \Rightarrow(\text{c} \in \text{campos}(r_1))\}$
Combinar : reg r_1 $\times$ reg r_2 $\longrightarrow$ reg
CombinarDeAcuerdoA : reg r_1 $\times$ reg r_2 $\times$ conj(campo) cs $\longrightarrow$ reg
ConMismoContenido : reg r_1 $\times$ reg r_2 $\times$ conj(campo) cs $\longrightarrow$ bool
$\quad\{cs \subseteq \big(\text{campos}(r_1) \cap \text{campos}(r_2)\big)\}$
DameCoincidente : conj(reg) rs $\times$ reg r $\times$ campo c $\longrightarrow$ reg
$\quad\{(\forall\ r_1 : \text{reg})\ (r_1 \in rs \Rightarrow \text{c} \in \text{campos}(r_1)) \wedge_{\text{L}} (\text{Coincide?}(rs, r, c))\}$
DameCoincidentesVal : campo c $\times$ nat n $\times$ conj(reg) rs $\longrightarrow$ conj(reg)
$\quad\{(\forall\ \text{r:reg})(\text{r} \in \text{rs} \Rightarrow_{\text{L}} c \in \text{campos(r)}) \wedge_{\text{L}} \text{Nat} = \text{tipo}(c)\}$
DameCoincidentesPal : campo c $\times$ string s $\times$ conj(reg) rs $\longrightarrow$ conj(reg)
$\quad\{(\forall\ \text{r:reg})(\text{r} \in \text{rs} \Rightarrow_{\text{L}} \text{c} \in \text{campos(r)}) \wedge_{\text{L}} \text{String} = \text{tipo}(c)\}$
noRepiten : reg r_1 $\times$ reg r_2 $\times$ campo c $\longrightarrow$ bool $\quad\{c \in \text{campos}(r_1) \wedge \text{c} \in \text{campos}(r_2)\}$
AgYCompDefault : reg r_1 $\times$ reg def $\times$ conj(campo) cs $\longrightarrow$ reg
DameCamposCoinc : conj(reg) rs $\times$ conj(campo) ci $\longrightarrow$ conj(reg)
$\quad\{(\forall\ \text{r} : \text{reg})(\ \text{r} \in \text{rs})\ \text{ci} \subseteq \text{campos(r)}\}$
RearmarCoinci : reg r $\times$ conj(campo) ci $\longrightarrow$ reg $\quad\{\text{ci} \subseteq \text{campos(r)}\}$
Busca : conj(campo) cs $\times$ reg r $\times$ conj(reg) rs $\longrightarrow$ conj(reg) $\quad\{\ \text{cs} \subseteq \text{campos(r)} \wedge \neg\ (\emptyset?(\text{cs}))\}$

axiomas $\forall r_1, r_2$: reg, $\forall c_1, c_2$: campo, $\forall cs, ci$: conj(campo), $\forall n$: nat, $\forall s$: string, $\forall rs$: conj(reg)

campos(NuevoRegistro) $\equiv$ $\emptyset$
campos(agValor(r_1,c_1,n) $\equiv$ Ag(c_1,campos(r_1))
campos(agPalabra(r_1,c_1,s)) $\equiv$ Ag(c_1,campos(r_1))

valorEn(agValor(r_1,c_1,n), c_2) $\equiv$ **if** $c_1 = c_2$ **then** n **else** valorEn(r_1, c_2) **fi**
valorEn(agPalabra(r_1,c_1,s),c_2) $\equiv$ valorEn(r_1,c_2)

palabraEn(agValor(r_1, c_1,n), c_2) $\equiv$ palabraEn(r_1, c_2)
palabraEn(agPalabra(r_1,c_1,s), c_2) $\equiv$ **if** $c_1 = c_2$ **then** s **else** palabraEn(r_1, c_2) **fi**

$r_1 = r_2$ $\equiv$ campos(r_1) = campos(r_2) $\wedge_{\text{L}}$ ConMismoContenido(r_1, r_2,campos(r_1))

ConMismoContenido(r_1, r_2, cs) $\equiv$ **if** $\emptyset?(cs)$ **then**
true
else
$\neg$noRepiten(r_1, r_2, DameUno(cs)) $\wedge$ ConMismoContenido(r_1, r_2,SinUno(cs))
fi

Coincide?(rs, r, c) $\equiv$ **if** $\emptyset?(rs)$ **then**
false
else
$\neg$ NoRepiten(DameUno(rs), r, c) $\vee$ Coincide?(SinUno(rs), r, c)
fi

DameCoincidente(rs, r, c) $\equiv$ **if** noRepiten(DameUno(rs), r, c) **then**
DameCoincidente(SinUno(rs), r, c)
else
DameUno(rs)
fi

Combinar(r_1, r_2) $\equiv$ CombinarDeAcuerdoA(r_1, r_2,campos(r_1) $\cup$ campos(r_2))

CombinarDeAcuerdoA(r_1, r_2, cs) $\equiv$ **if** $\emptyset?$(cs) **then**
r_1
else
if DameUno(cs)$\in$campos(r_1) **then**
CombinarDeAcuerdoA(r_1, r_2, SinUno(cs))
else
if Nat = tipo(DameUno(cs)) **then**
agValor$\big($CombinarDeAcuerdoA(r_1, r_2,SinUno(cs)),DameUno(cs),
valorEn(r_2,DameUno(cs))$\big)$
else
agPalabra$\big($CombinarDeAcuerdoA(r_1, r_2,SinUno(cs)),
DameUno(cs),palabraEn(r_2,DameUno(cs))$\big)$
fi
fi
fi

noRepiten(r_1, r_2, c) $\equiv$ Nat = tipo(c)$\wedge_{\text{L}}$ $\neg$ $\big($valorEn(r_1, c)=valorEn(r_2, c)$\big)\vee$
(String = tipo(c)$\wedge_{\text{L}}\neg$(palabraEn(r_1, c)=palabraEn(r_2, c)))

DameCoincidentesVal(c, n, rs) $\equiv$ **if** $\emptyset?(rs)$ **then**
$\emptyset$
else
if ValorEn(DameUno(rs),c)=n **then**
Ag$\big($DameUno(rs),DameCoincidentesVal(c,n,SinUno(rs))$\big)$
else
DameCoincidentesVal(c,n,SinUno(rs))
fi
fi

DameCoincidentesPal(c, s, rs) $\equiv$ **if** $\emptyset?(rs)$ **then**
$\emptyset$
else
if PalabraEn(DameUno(rs),c)=s **then**
Ag$\big($DameUno(rs),DameCoincidentesPal(c,s,SinUno(rs))$\big)$
else
DameCoincidentesPal(c,s,SinUno(rs))
fi
fi

```
AgYCompDefault(r, def, cs) ≡ if ∅?(cs) then
                                NuevoRegistro
                             else
                                if DameUno(cs) ∈ campos(r) then
                                   if Nat = tipo(DameUno(cs)) then
                                      AgValor(AgYCompDefault(r, def, SinUno(cs)),DameUno(cs),
                                      ValorEn(r,DameUno(cs)))
                                   else
                                      AgPalabra(AgYCompDefault(r, def, SinUno(cs)),DameUno(cs),
                                      PalabraEn(r,DameUno(cs)))
                                   fi
                                else
                                   if Nat = tipo(DameUno(cs)) then
                                      AgValor(AgYCompDefault(r, def, SinUno(cs)), DameUno(cs),
                                      ValorEn(def, DameUno(cs)))
                                   else
                                      AgPalabra(AgYCompDefault(r, def, SinUno(cs)), DameUno(cs),
                                      PalabraEn(def, DameUno(cs)))
                                   fi
                                fi
                             fi

DameCamposCoinci(rs,ci) ≡ if ∅?(rs) then
                             ∅
                          else
                             Ag(RearmarCoinci(DameUno(rs),ci),DameCamposCoinci(SinUno(rs),ci)
                          fi
RearmarCoinci(r,ci) ≡ if ∅?(ci) then
                         NuevoRegistro
                      else
                         if tipo(DameUno(ci))= Nat then
                            AgValor(RearmarCoinci(r,SinUno(ci)),DameUno(ci),valorEn(r,DameUno(ci)))
                         else
                            AgPalabra(RearmarCoinci(r,SinUno(ci)),DameUno(ci),palabraEn(r,DameUno(ci)))
                         fi
                      fi
Busca(cs,r,rs) ≡ if ∅?(SinUno(cs)) then
                    if tipo(DameUno(cs)) = Nat then
                       DameCoincidentesVal(DameUno(cs),valorEn(r,DameUno(cs)),rs)
                    else
                       DameCoincidentesPal(DameUno(cs),valorEn(r,DameUno(cs)),rs)
                    fi
                 else
                    if tipo(DameUno(cs)) = Nat then
                       DameCoincidentesVal(DameUno(cs),valorEn(r,DameUno(cs)),Busca(SinUno(cs),r,rs))
                    else
                       DameCoincidentesPal(DameUno(cs),valorEn(r,DameUno(cs)),Busca(SinUno(cs),r,rs))
                    fi
                 fi
```

Fin TAD