



TP de Especificación

Sudoku

1 de Abril de 2017

Algoritmos y Estructuras de Datos I

Grupo 10

Integrante	LU	Correo electrónico
Gomez Salaverri, Francisco	001/01	francisco@gomezsalaverri.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (++54 +11) 4576-3300

<http://www.exactas.uba.ar>

1. Problemas

1. **proc sudoku_esTableroValido (in t: seq<seq<Z>>, out result: Bool) {**
Pre {True}
Post {tableroValido(t) = result}
pred tableroValido (t: seq<seq<Z>>) {
 $esFilaValida(t) \wedge esColumnaValida(t)$
}
pred esFilaValida (t: seq<seq<Z>>) {
 $(\forall i : \mathbb{Z})(\forall j : \mathbb{Z}) enRango(t, i) \wedge_L$
 $enRango(t[i], j) \wedge_L length(t[i]) = 9 \rightarrow_L 0 \leq t[i][j] \leq 9$
}
pred esColumnaValida (t: seq<seq<Z>>) {
 $(\forall i : \mathbb{Z})(\forall j : \mathbb{Z}) length(t) = 9 \wedge enRango(t, i) \wedge_L$
 $enRango(t[i], j) \rightarrow_L 0 \leq t[i][j] \leq 9$
}
}
2. **proc sudoku_esCeldaVacía (in t: seq<seq<Z>>, in f: Z, in c: Z, out result: Bool) {**
Pre {tableroValido(t)}
Post {}
}
3. **proc sudoku_nroDeCeldasVacías (in t: seq<seq<Z>>, out result : Z) {**
Pre {tableroValido(t)}
Post {}
}
4. **proc sudoku_primeraCeldaVacíaFila (in t: seq<seq<Z>>, out result : Z) {**
Pre {tableroValido(t)}
Post {if CeldasVacías(t) = 0 then -1 else $(\exists i : \mathbb{Z})(\exists j : \mathbb{Z}) result = i \wedge enRango(t, i) \wedge_L enRango(t[i], j) \wedge_L$
 $CeldaVacía(t, i, j) \wedge menorFilaVacía (t, i) \wedge menorColumnaDeLaFilaVacía(t, i, j) \text{ fi } }$
}
5. **proc sudoku_primeraCeldaVacíaColumna (in t: seq<seq<Z>>, out result : Z) {**
Pre {tableroValido(t)}
Post {if CeldasVacías(t) = 0 then -1 else $(\exists i : \mathbb{Z})(\exists j : \mathbb{Z}) result = j \wedge enRango(t, i) \wedge_L enRango(t[i], j) \wedge_L$
 $CeldaVacía(t, i, j) \wedge menorFilaVacía (t, i) \wedge menorColumnaDeLaFilaVacía(t, i, j) \text{ fi } }$
}
6. **proc sudoku_valorEnCelda (in t: seq<seq<Z>>, in f: Z, in c: Z, out result: Bool) {**
Pre {tableroValido(t)}
Post {}
}
7. **proc sudoku_llenarCelda (inout t: seq<seq<Z>> in f: Z, in c: Z, out result: Bool) {**
Pre {tableroValido(t)}
Post {}
}
8. **proc sudoku_vaciarCelda (inout t: seq<seq<Z>>, in f: Z, in c: Z, out result: Bool) {**
Pre {tableroValido(t)}
Post {}
}
9. **proc sudoku_esTableroParcialmenteResuelto (in t: seq<seq<Z>>, out result: Bool) {**
Pre {True}
Post {}
}
10. **proc sudoku_esTableroTotalmenteResuelto (in t: seq<seq<Z>>, out result: Bool) {**
Pre {tableroValido(t)}

```

    Post {}
}

11.  proc sudoku_esSubTablero (in t0, t1 : seq⟨seq⟨ℤ⟩⟩, out result : Bool){
    Pre {tableroValido(t0), tableroValido(t1)}
    Post {}
}

12.  proc sudoku_tieneSolucion (in t: seq⟨seq⟨ℤ⟩⟩, out tienesolucion: Bool) {
    Pre {tableroValido(t)}
    Post {}
}

13.  proc sudoku_resolver (inout t: seq⟨seq⟨ℤ⟩⟩, out tienesolucion: Bool) {
    Pre {True}
    Post {}
}

14.  proc sudoku_copiarTablero (in t: seq⟨seq⟨ℤ⟩⟩, out target: seq⟨seq⟨ℤ⟩⟩) {
    Pre {True}
    Post {}
}

```

2. Predicados y Auxiliares generales

```

pred Nombre (t: seq⟨seq⟨ℤ⟩⟩) {True}
pred PredLargo (t: seq⟨seq⟨ℤ⟩⟩) {
(∀i : ℤ)(∀j : ℤ)True
}

fun Aux (i: ℤ) : Bool = True;
pred enRango (t: seq⟨t⟩, i:ℤ) {
0 ≤ i < length(t)
}

fun Resolver (t: seq⟨seq⟨ℤ⟩⟩) : seq⟨seq⟨ℤ⟩⟩ = if esSub(t,x)∧L tableroParcialmenteResuleto(x) then x else t fi ;
pred menorFilaVacía (t: seq⟨seq⟨ℤ⟩⟩, i: ℤ) {
(∀f : ℤ)(∀g : ℤ)enRango(t, f)∧L enRango(t[f],g)
→L celdaVacía(t,f,g) ∧ f ≥ i}
pred menorColumnaDeLaFilaVacía (t: seq⟨seq⟨ℤ⟩⟩, i: ℤ, j: ℤ) {
(∀f : ℤ)enRango(t[i], g)
→L celdaVacía(t,i,g) ∧ g ≥ j)
}

```

3. Decisiones tomadas