TP de Especificación

Sudoku

---

24 de Abril de 2017   Algoritmos y Estructuras de Datos I

Grupo 17

Integrante	LU	Correo electrónico
Maqueda, Ignacio	279/14	ignaciomaqueda95@gmail.com
Parral, Guillermo	280/16	guillermoeparral@gmail.com
Quintela, Gonzalo	089/16	gquintela@dc.uba.ar
Sirio, Tomás	440/16	tomassirio@gmail.com

1. Problemas

```
proc  sudoku_esTableroValido (in t: seq⟨seq⟨ℤ⟩⟩, out result: Bool)  {
       Pre {True}
       Post {result = esTableroValido(t)}
}
```

```
proc sudoku_esCeldaVacia (in t: seq⟨seq⟨ℤ⟩⟩, in f: ℤ,in c: ℤ, out result: Bool)  {
       Pre {esTableroValido(t) ∧ 0 ≤ f, c ≤ 8}
       Post {result = (t[f][c] = 0)}
}
```

```
proc sudoku_nroDeCeldasVacias (in t: seq⟨seq⟨ℤ⟩⟩, out result: ℤ)  {
       Pre {esTableroValido(t)}
       Post {result = ∑_{i=0}^{|t|-1}(∑_{j=0}^{|t|-1} if t[i][j] = 0 then 1 else 0 fi)}
}
```

```
proc sudoku_primeraCeldaVaciaFila (in t: seq⟨seq⟨ℤ⟩⟩, out result: ℤ)  {
       Pre {esTableroValido(t)}
       Post {(∃i : ℤ)(esPrimeraFilaConCeldaVacia(t, i) ∧ result = i) ∨
          (¬(∃i : ℤ)(0 ≤ i < |t| ∧_L filaTieneCeldaVacia(t[i])) ∧ result = −1)}
}
```

```
proc sudoku_primeraCeldaVaciaColumna (in t: seq⟨seq⟨ℤ⟩⟩, out result: ℤ)  {
       Pre {esTableroValido(t)}
       Post {(∃i : ℤ)(esPrimeraFilaConCeldaVacia(t, i) ∧
          (∃k : ℤ)(esPrimeraCeldaVaciaEnFila(t[i], k) ∧ result = k)) ∨
          (¬(∃i : ℤ)(0 ≤ i < |t| ∧_L filaTieneCeldaVacia(t[i])) ∧ result = −1)}
       pred esPrimeraCeldaVaciaEnFila (s: seq⟨ℤ⟩, k: ℤ) {s[k] = 0 ∧ (∀i : ℤ)(0 ≤ i < k ⟶_L s[i] ≠ 0)}
}
```

```
proc sudoku_valorEnCelda (in t: seq⟨seq⟨ℤ⟩⟩, in f: ℤ, in c: ℤ, out result: ℤ)  {
       Pre {(esTableroValido(t) ∧ 0 ≤ f, c ≤ 8) ∧_L t[f][c] ≠ 0}
       Post {result = t[f][c]}
}
```

```
proc sudoku_llenarCelda (inout t: seq⟨seq⟨ℤ⟩⟩, in f: ℤ, in c: ℤ, in value: ℤ)  {
       Pre {(esTableroValido(t) ∧ 0 ≤ f, c ≤ 8 ∧ 1 ≤ value ≤ 9 ∧ t = t_0) ∧_L t[f][c] = 0}
       Post {esTableroValido(t) ∧ t[f][c] = value ∧ (∀i : ℤ)(∀j : ℤ)((0 ≤ i, j < |t| ∧ (i ≠ f ∨ j ≠ c)) ⟶_L t[i][j] = t_0[i][j])}
}
```

```
proc sudoku_vaciarCelda (inout t: seq⟨seq⟨ℤ⟩⟩, in f: ℤ, in c: ℤ)  {
       Pre {(esTableroValido(t) ∧ 0 ≤ f, c ≤ 8 ∧ t = t_0) ∧_L t[f][c] ≠ 0}
       Post {esTableroValido(t) ∧ t[f][c] = 0 ∧ (∀i : ℤ)(∀j : ℤ)((0 ≤ i, j < |t| ∧ (i ≠ f ∨ j ≠ c)) ⟶_L t[i][j] = t_0[i][j])}
}
```

```
proc sudoku_esTableroParcialmenteResuelto (in t: seq⟨seq⟨ℤ⟩⟩, out result: Bool))  {
       Pre {True}
       Post {result = esTableroParcialmenteResuelto(t)}
}
```

```
proc sudoku_esTableroTotalmenteResuelto (in t: seq⟨seq⟨ℤ⟩⟩, out result: Bool)  {
      Pre {True}
      Post {result = esTableroTotalmenteResuelto(t)}
}


proc sudoku_esSubTablero (in t₀, t₁ : seq⟨seq⟨ℤ⟩⟩, out result : Bool){
      Pre {True}
      Post {result = esSubTablero(t₀, t₁)}
}


proc sudoku_tieneSolucion (in t: seq⟨seq⟨ℤ⟩⟩, out tieneSolucion: Bool)  {
      Pre {esTableroValido(t)}
      Post {tieneSolucion = (∃s : seq⟨seq⟨ℤ⟩⟩)(esTableroTotalmenteResuelto(s) ∧ esSubTablero(t, s))}
}


proc sudoku_resolver (inout t: seq⟨seq⟨ℤ⟩⟩, out tieneSolucion: Bool)  {
      Pre {esTableroValido(t) ∧ t = t₀}
      Post {((∃s : seq⟨seq⟨ℤ⟩⟩)(esTableroTotalmenteResuelto(s) ∧ esSubTablero(t, s)
         ∧ (tieneSolucion = True) ∧ (t = s)) ∨ (¬(∃s : seq⟨seq⟨ℤ⟩⟩)(esTableroTotalmenteResuelto(s) ∧
         esSubTablero(t, s)) ∧ (tieneSolucion = False) ∧ (t = t₀))}
}

proc sudoku_copiarTablero (in src: seq⟨seq⟨ℤ⟩⟩, out target: seq⟨seq⟨ℤ⟩⟩)  {
      Pre {esTableroValido(src)}
      Post {esTableroValido(target) ∧ (∀i : ℤ)(∀j : ℤ)(0 ≤ i, j < |src| ⟶_L target[i][j] = src[i][j])}
}
```

2. Predicados y Auxiliares generales

```
   pred esMatriz (t: seq⟨seq⟨ℤ⟩⟩) {
(∀i : ℤ)(∀j : ℤ)(0 ≤ i, j < |t| ⟶_L |t[i]| = |t[j]|)
}


   pred esMatrizCuadrada (t: seq⟨seq⟨ℤ⟩⟩) {
esMatriz(t) ∧ (cantidadFilas(t) = cantidadColumnas(t))
}


   pred esTableroValido (t: seq⟨seq⟨ℤ⟩⟩) {esMatrizCuadrada(t) ∧_L |t| = 9 ∧
(∀i : ℤ)(∀j : ℤ)(0 ≤ i, j < |t| ⟶_L 0 ≤ t[i][j] ≤ 9)
}


   pred filaTieneCeldaVacia (f: seq⟨ℤ⟩) {
(∃i : ℤ)(0 ≤ i < |f| ∧_L f[i] = 0)
}


   pred esPrimeraFilaConCeldaVacia (t: seq⟨seq⟨ℤ⟩⟩, i: ℤ) {0 ≤ i < |t| ∧_L filaTieneCeldaVacia(t[i]) ∧
(∀j : ℤ)(0 ≤ j < i ⟶_L ¬filaTieneCeldaVacia(t[j]))}


   pred noHayRepetidosEnRegion (t: seq⟨seq⟨ℤ⟩⟩) {(∀i : ℤ)(∀j : ℤ)(∀k : ℤ)(∀l : ℤ)
((0 ≤ i, j, k, l < |t| ∧ (i div 3 = k div 3) ∧ (j div 3 = l div 3) ∧ (i ≠ k ∨ j ≠ l)) ⟶_L (t[i][j] = 0 ∨ t[k][l] = 0 ∨ t[i][j] ≠ t[k][l]))
}


   pred noHayRepetidosEnFila (s: seq⟨ℤ⟩) {(∀i : ℤ)(∀j : ℤ)((0 ≤ i, j < |s| ∧ i ≠ j) ⟶_L (s[i] = 0 ∨ s[j] = 0 ∨ s[i] ≠ s[j]))
}
```

pred noHayRepetidosEnColumna (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) $\{(\forall j : \mathbb{Z})(0 \leq j < |t| \longrightarrow_L (\forall l : \mathbb{Z})(\forall k : \mathbb{Z})$
$((0 \leq l, k < |t| \wedge l \neq k) \longrightarrow_L (t[l][j] = 0 \vee t[k][j] = 0 \vee t[l][j] \neq t[k][j])))$
}

pred esTableroParcialmenteResuelto (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) $\{esTableroValido(t) \wedge (\forall i : \mathbb{Z})(0 \leq i < |t| \longrightarrow_L$
$noHayRepetidosEnFila(t[i])) \wedge noHayRepetidosEnColumna(t) \wedge noHayRepetidosEnRegion(t)$
}

pred esTableroTotalmenteResuelto (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) $\{esTableroParcialmenteResuelto(t) \wedge (\forall i : \mathbb{Z})(0 \leq i < |t| \longrightarrow_L$
$\neg filaTieneCeldaVacia(t[i]))$
}

pred esSubTablero ($t_0, t_1 : seq\langle seq\langle \mathbb{Z}\rangle\rangle$)$\{(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})$
$((0 \leq i, j < |t| \wedge_L t_0[i][j] \neq 0) \longrightarrow_L t_0[i][j] = t_1[i][j])$
}

fun cantidadFilas (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) $: \mathbb{Z} = |t|$;

fun cantidadColumnas (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) $: \mathbb{Z} =$ if $cantidadFilas(t) > 0$ then $|t[0]|$ else 0 fi ;