

# TP de Especificación

## Sudoku

21 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



### 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

```
proc sudoku_esTableroValido (in t: seq\langle seq\langle \mathbb{Z}\rangle\rangle, out result: Bool) {
           Pre {True}
            Post \{result = esTableroValido(t)\}
}
proc sudoku_esCeldaVacia (in t: seq\langle seq\langle \mathbb{Z}\rangle\rangle, in f: \mathbb{Z},in c: \mathbb{Z}, out result: Bool) {
            Pre \{esTableroValido(t) \land_L 0 \leq f, c < |t|\}
            Post \{result = (t[f][c] = 0)\}
}
proc sudoku_nroDeCeldasVacias (in t: seq\langle seq\langle \mathbb{Z}\rangle\rangle, out result: \mathbb{Z}) {
            Pre \{esTableroValido(t)\}\
           Post \{\sum_{i=0}^{|t|-1}(\sum_{j=0}^{|t|-1} \text{if } t[i][j]=0 \text{ then } 1 \text{ else } 0 \text{ fi})\}
}
proc sudoku_primeraCeldaVaciaFila (in t: seq\langle seq\langle \mathbb{Z}\rangle\rangle, out result: \mathbb{Z}) {
            Pre \{esTableroValido(t)\}
            Post {if (\exists i : \mathbb{Z})(0 \leq i < |t| \land_L filaTieneCeldaVacia(t[i]) \land_L (\forall j : \mathbb{Z})
                 (0 \le j < i \longrightarrow_L \neg filaTieneCeldaVacia(t[j])))
                 then result = i
                 else result = -1 fi
}
proc sudoku_primeraCeldaVaciaColumna (in t: seq\langle seq\langle \mathbb{Z}\rangle\rangle, out result: \mathbb{Z}) {
            Pre \{esTableroValido(t)\}
            \texttt{Post} \ \{ \text{if} \ (\exists i : \mathbb{Z}) (0 \leq i < |t| \land_L filaTieneCeldaVacia(t[i]) \land_L (\forall j : \mathbb{Z}) (0 \leq j < i \longrightarrow_L \neg filaTieneCeldaVacia(t[j])) \} \} \} 
                 then result = indicePrimeraCeldaVaciaEnFila(t[i])
                 else result = -1 fi
}
proc sudoku_valorEnCelda (in t: seq\langle seq\langle \mathbb{Z}\rangle\rangle, in f: \mathbb{Z}, in c: \mathbb{Z}, out result: \mathbb{Z}) {
            \texttt{Pre} \ \{esTableroValido(t) \land_L 0 \leq f, c \leq 8 \land_L t[f][c] \neq 0\}
            Post \{result = t[f][c]\}
}
proc sudoku_llenarCelda (inout t: seq\langle seq\langle \mathbb{Z}\rangle\rangle, in f: \mathbb{Z}, in c: \mathbb{Z}, in value: \mathbb{Z}) {
            Pre \{esTableroValido(t) \land_L 0 \leq f, c \leq 8 \land_L t[f][c] = 0 \land_L 1 \leq value \leq 9 \land_L t = t_0\}
            \texttt{Post}\ \{t[f][c] = value \land_L (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})((0 \le i, j < |t| \land_L (i \ne f \lor j \ne c)) \longrightarrow_L t[i][j] = t_0[i][j])\}
}
proc sudoku_vaciarCelda (inout t: seq\langle seq\langle \mathbb{Z}\rangle\rangle, in f: \mathbb{Z}, in c: \mathbb{Z}) {
            Pre \{esTableroValido(t) \land_L 0 \leq f, c \leq 8 \land_L t[f][c] \neq 0 \land_L t = t_0\}
            \texttt{Post}\ \{t[f][c] = 0 \land_L (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})((0 \le i, j < |t| \land_L (i \ne f \lor j \ne c)) \longrightarrow_L t[i][j] = t_0[i][j])\}
}
proc sudoku_esTableroParcialmenteResuelto (in t: seq\langle seq\langle \mathbb{Z}\rangle\rangle, out result: Bool)) {
            Pre {True}
            Post \{result = esTableroParcialmenteResuelto(t)\}
}
```

```
proc sudoku_esTableroTotalmenteResuelto (in t: seq\langle seq\langle \mathbb{Z}\rangle\rangle, out result: Bool) {
          Pre {True}
          Post \{result = esTableroTotalmenteResuelto(t)\}
}
proc sudoku_esSubTablero (in t_0, t_1 : seq\langle seq\langle \mathbb{Z} \rangle \rangle, out result : Bool){
          Pre {True}
          Post \{result = esSubTablero(t_0, t_1)\}
}
proc sudoku_tieneSolucion (in t: seq\langle seq\langle \mathbb{Z}\rangle\rangle, out tieneSolucion: Bool) {
          Pre \{esTableroValido(t)\}
          Post \{tieneSolucion = (\exists s : seq\langle seq\langle \mathbb{Z}\rangle\rangle)(esTableroTotalmenteResuelto(s) \land_L esSubTablero(s,t))\}
}
proc sudoku_resolver (inout t: seq\langle seq\langle \mathbb{Z}\rangle\rangle, out tieneSolucion: Bool) {
          Pre \{esTableroValido(t) \land_L t = t_0\}
          Post {if (\exists s : seq\langle seq\langle \mathbb{Z}\rangle\rangle)(esTableroTotalmenteResuelto(s) \land_L esSubTablero(s,t))}
               then tieneSolucion = True \land t = s
               else tieneSolucion = False fi
}
proc sudoku_copiarTablero (in src: seq\langle seq\langle \mathbb{Z}\rangle\rangle, out target: seq\langle seq\langle \mathbb{Z}\rangle\rangle\rangle) {
          Pre \{esTableroValido(src)\}
          Post \{esTableroValido(target) \land_L (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \le i, j < |src| \longrightarrow_L target[i][j] = src[i][j])\}
}
2.
         Predicados y Auxiliares generales
```

```
\begin{aligned} &\text{pred esMatriz } (t:seq\langle seq\langle \mathbb{Z}\rangle\rangle) \; \{ \\ &\langle \forall i: \mathbb{Z} \rangle (\forall j: \mathbb{Z}) (0 \leq i,j < |t| \longrightarrow_L |t[i]| = |t[j]|) \; \} \\ &\text{fun cantidadFilas } (t:seq\langle seq\langle \mathbb{Z}\rangle\rangle) : \mathbb{Z} = |t| \; ; \\ &\text{fun cantidadColumnas } (t:seq\langle seq\langle \mathbb{Z}\rangle\rangle) : \mathbb{Z} = \text{if } cantidadFilas(t) > 0 \; \text{then } |t[0]| \; \text{else } 0 \; \text{fi} \; ; \\ &\text{pred esMatrizCuadrada } (t:seq\langle seq\langle \mathbb{Z}\rangle\rangle) : \mathbb{Z} = \text{if } cantidadFilas(t) > 0 \; \text{then } |t[0]| \; \text{else } 0 \; \text{fi} \; ; \\ &\text{pred esMatrizCuadrada } (t:seq\langle seq\langle \mathbb{Z}\rangle\rangle) \; \{ \\ &esMatriz(t) \land (cantidadFilas(t) = cantidadColumnas(t)) \} \\ &\text{pred esTableroValido } (t:seq\langle seq\langle \mathbb{Z}\rangle\rangle) \; \{ esMatrizCuadrada(t) \land_L |t| = 9 \land_L \\ &\langle \forall i: \mathbb{Z} \rangle (\forall j: \mathbb{Z}) (0 \leq i,j < |t| \longrightarrow_L 0 \leq t[i][j] \leq 9) \} \\ &\text{pred filaTieneCeldaVacia } (f:seq\langle \mathbb{Z}\rangle) \; \{ \\ &(\exists i: \mathbb{Z}) (0 \leq i < |f| \land_L f[i] = 0) \} \\ &\text{fun indicePrimeraCeldaVaciaEnFila } (s:seq\langle \mathbb{Z}\rangle) : \mathbb{Z} = \text{if } ((\exists i: \mathbb{Z}) (0 \leq i < |s| \land_L s[i] = 0 \land_L \\ &\langle \forall j: \mathbb{Z}) (0 \leq j < i \longrightarrow_L s[j] \neq 0))) \; \text{then } i \; \text{else } -1 \; \text{fi} \; ; \\ &\text{pred noHayRepetidosEnRegion } (t:seq\langle seq\langle \mathbb{Z}\rangle\rangle) \; \{ (\forall i: \mathbb{Z}) (\forall j: \mathbb{Z}) (\forall k: \mathbb{Z}) (\forall l: \mathbb{Z}) \\ &(0 \leq i,j,k,l < 9 \land_L (idiv3 = kdiv3) \land_L (jdiv3 = ldiv3) \land_L (i \neq k \lor j \neq l) \longrightarrow_L (t[i][j] = 0 \lor t[i][j] \neq t[k][l])) \} \\ \end{aligned}
```

```
 \begin{array}{l} \text{pred noHayRepetidosEnFila } (s: seq\langle \mathbb{Z} \rangle) \; \{ (\forall i: \mathbb{Z}) (\forall j: \mathbb{Z}) (0 \leq i, j < |s| \wedge_L \; j \neq i \longrightarrow_L \; (s[i] = 0 \vee s[j] = 0 \vee s[i] \neq s[j])) \} \\ \\ \text{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 \; l \neq k \longrightarrow_L \; (t[l][j] = 0 \vee t[k][j] = 0 \vee t[l][j] \neq t[k][j]))) \} \\ \\ \text{pred esTableroParcialmenteResuelto } (t: seq\langle seq\langle \mathbb{Z} \rangle \rangle) \; \{ esTableroValido(t) \wedge_L \; (\forall i: \mathbb{Z}) (0 \leq i < |t| \longrightarrow_L \; noHayRepetidosEnFila(t[i])) \wedge_L \; noHayRepetidosEnColumna(t) \wedge_L \; noHayRepetidosEnRegion(t) \} \\ \\ \text{pred esTableroTotalmenteResuelto } (t: seq\langle seq\langle \mathbb{Z} \rangle \rangle) \; \{ esTableroParcialmenteResuelto(t) \wedge_L \; (\forall i: \mathbb{Z}) (0 \leq i < |t| \longrightarrow_L \; \neg filaTieneCeldaVacia(t[i])) \} \\ \\ \text{pred esSubTablero } (t_0, t_1: seq\langle seq\langle \mathbb{Z} \rangle \rangle) \{ esTableroValido(t_0) \wedge_L \; esTableroValido(t_1) \wedge_L \; (\forall i: \mathbb{Z}) (\forall j: \mathbb{Z}) \; (0 \leq i |t| \wedge_L \; t_0[i][j] \neq 0) \longrightarrow_L \; t_0[i][j] = t_1[i][j]) \} \\ \end{aligned}
```

### 3. Decisiones tomadas