DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

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 {$esTableroValido(t) \wedge_L 0 \leq f, c < |t|$}
        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 {$\sum_{i=0}^{|t|-1}(\sum_{j=0}^{|t|-1}$ if $t[i][j] = 0$ then 1 else 0 fi)}
}
```

```
proc sudoku_primeraCeldaVaciaFila (in t: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$, out result: $\mathbb{Z}$) {
        Pre {$esTableroValido(t)$}
        Post {if $(\exists i : \mathbb{Z})(0 \leq i < |t| \wedge_L filaTieneCeldaVacia(t[i]) \wedge_L (\forall j : \mathbb{Z})$
            $(0 \leq j < i \longrightarrow_L \neg filaTieneCeldaVacia(t[j])))$
            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 {$esTableroValido(t)$}
        Post {if $(\exists i : \mathbb{Z})(0 \leq i < |t| \wedge_L filaTieneCeldaVacia(t[i]) \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \longrightarrow_L \neg filaTieneCeldaVacia(t[j])))$
            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: $\mathbb{Z}$) {
        Pre {$esTableroValido(t) \wedge_L 0 \leq f, c \leq 8 \wedge_L t[f][c] \neq 0$}
        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 {$esTableroValido(t) \wedge_L 0 \leq f, c \leq 8 \wedge_L t[f][c] = 0 \wedge_L 1 \leq value \leq 9 \wedge_L t = t_0$}
        Post {$t[f][c] = value \wedge_L (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})((0 \leq i, j < |t| \wedge_L (i \neq f \vee j \neq c)) \longrightarrow_L t[i][j] = t_0[i][j])$}
}
```

```
proc sudoku_vaciarCelda (inout t: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$, in f: $\mathbb{Z}$, in c: $\mathbb{Z}$) {
        Pre {$esTableroValido(t) \wedge_L 0 \leq f, c \leq 8 \wedge_L t[f][c] \neq 0 \wedge_L t = t_0$}
        Post {$t[f][c] = 0 \wedge_L (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})((0 \leq i, j < |t| \wedge_L (i \neq f \vee j \neq c)) \longrightarrow_L t[i][j] = t_0[i][j])$}
}
```

```
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 {$tieneSolucion = (\exists s : seq\langle seq\langle\mathbb{Z}\rangle\rangle)(esTableroTotalmenteResuelto(s) \wedge_L esSubTablero(s, t))$}
}
```

```
proc sudoku_resolver (inout t: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$, out tieneSolucion: Bool) {
        Pre {$esTableroValido(t) \wedge_L t = t_0$}
        Post {if $(\exists s : seq\langle seq\langle\mathbb{Z}\rangle\rangle)(esTableroTotalmenteResuelto(s) \wedge_L esSubTablero(s, t))$
            then $tieneSolucion = \text{True} \wedge t = s$
            else $tieneSolucion = \text{False}$ fi}
}
```

```
proc sudoku_copiarTablero (in src: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$, out target: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
        Pre {$esTableroValido(src)$}
        Post {$esTableroValido(target) \wedge_L (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i, j < |src| \longrightarrow_L target[i][j] = src[i][j])$}
}
```

2. Predicados y Auxiliares generales

```
   pred esMatriz (t: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i, j < |t| \longrightarrow_L |t[i]| = |t[j]|)$
}
```

```
   pred esMatrizCuadrada (t: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$esMatriz(t) \wedge (cantidadFilas(t) = cantidadColumnas(t))$
}
```

```
   pred esTableroValido (t: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {$esMatrizCuadrada(t) \wedge_L |t| = 9 \wedge_L$
$(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i, j < |t| \longrightarrow_L 0 \leq t[i][j] \leq 9)$
}
```

```
   pred filaTieneCeldaVacia (f: $seq\langle\mathbb{Z}\rangle$) {
$(\exists i : \mathbb{Z})(0 \leq i < |f| \wedge_L f[i] = 0)$
}
```

```
   pred noHayRepetidosEnRegion (t: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {$(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(\forall k : \mathbb{Z})(\forall l : \mathbb{Z})$
$(0 \leq i, j, k, l < 9 \wedge (i \text{ div } 3 = k \text{ div } 3) \wedge (j \text{ div } 3 = l \text{ div } 3) \wedge (i \neq k \vee j \neq l) \longrightarrow_L (t[i][j] = 0 \vee_L t[k][l] = 0 \vee_L t[i][j] \neq t[k][l]))$
}
```

```
   pred noHayRepetidosEnFila (s: $seq\langle\mathbb{Z}\rangle$) {$(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i, j < |s| \wedge j \neq i \longrightarrow_L (s[i] = 0 \vee_L s[j] = 0 \vee_L s[i] \neq s[j]))$
}
```

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

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

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

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

`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 ;

`fun indicePrimeraCeldaVaciaEnFila` (s: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ $=$ if $((\exists i : \mathbb{Z})(0 \leq i < |s| \land_L s[i] = 0 \land_L (\forall j : \mathbb{Z})(0 \leq j < i \longrightarrow_L s[j] \neq 0)))$ then i else -1 fi ;

3. Decisiones tomadas