DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

TP de Especificación

Sudoku

24 de Abril de 2017 — Algoritmos y Estructuras de Datos I

Grupo 5

Integrante	LU	Correo electrónico
Caballero, Tomás Leonel	628/15	`tomycaballero95@gmail.com`
Farias, Dante Ezequiel	365/15	`dantecuervo94@hotmail.com`
Latronico, Joaquin Ignacio	484/16	`ignacio.latronico96@gmail.com`
Sittner, Daiana Natasha	630/15	`daiana.sittner@hotmail.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: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, out result: Bool) {
       Pre {True}
       Post {$result = esTableroValido(t)$}
   }
```

```
2. 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 {$esTableroValido(t) \land (0 \leq f < 9) \land (0 \leq c < 9)$}
       Post {$result = (t[f][c] = 0)$}
   }
```

```
3. proc sudoku_nroDeCeldasVacias (in t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, out result: $\mathbb{Z}$) {
       Pre {$esTableroValido(t)$}
       Post {$result = cantCeldasVacias(t)$}
   }
```

```
4. proc sudoku_primeraCeldaVaciaFila (in t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, out result: $\mathbb{Z}$) {
       Pre {$esTableroValido(t)$}
       Post {
           $(cantCeldasVacias(t) = 0 \land result = -1) \lor$
           $((0 \leq result < 9) \land_L (0 \in t[result]) \land (\forall\ f : \mathbb{Z})(0 \leq f < result \longrightarrow_L \neg(0 \in t[f]))$
       }
   }
```

```
5. proc sudoku_primeraCeldaVaciaColumna (in t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, out result: $\mathbb{Z}$) {
       Pre {$esTableroValido(t)$}
       Post {
           $(cantCeldasVacias(t) = 0 \land result = -1) \lor$
           $((0 \leq result < 9) \land_L (\exists\ f_0 : \mathbb{Z})((0 \leq f_0 < 9) \land_L (t[f_0][result] = 0) \land$
           $(\forall\ f_1 : \mathbb{Z})(0 \leq f_1 \leq f_0 \rightarrow (\forall\ c : \mathbb{Z})((f_1 = f_0 \land 0 \leq c < result) \lor (f_0 \neq f_1 \land 0 \leq c < 9) \longrightarrow_L t[f_1][c] \neq 0))))$
       }
   }
```

```
6. 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 {$esTableroValido(t) \land (0 \leq f < 9) \land (0 \leq c < 9) \land_L (t[f][c] \neq 0)$}
       Post {$result = t[f][c]$}
   }
```

```
7. 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 {$esTableroValido(t) \land (0 \leq f < 9) \land (0 \leq c < 9) \land (1 \leq value \leq 9) \land (t_0 = t) \land_L (t[f][c] = 0)$}
       Post {
           $esTableroValido(t) \land_L (t[f][c] = value) \land$
           $(\forall\ f_0 : \mathbb{Z})(0 \leq f_0 < 9 \rightarrow (\forall\ c_0 : \mathbb{Z})(0 \leq c_0 < 9 \longrightarrow_L ((f_0 = f \land c_0 = c) \lor t[f_0][c_0] = t_0[f_0][c_0])))$
       }
   }
```

8. `proc sudoku_vaciarCelda` (inout t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, in f: $\mathbb{Z}$, in c : $\mathbb{Z}$) {

    `Pre` $\{esTableroValido(t) \land (0 \leq f < 9) \land (0 \leq c < 9) \land (t_0 = t) \land_L (t[f][c] \neq 0)\}$

    `Post` {

        $esTableroValido(t) \land_L (t[f][c] = 0) \land$
        $(\forall\ f_0 : \mathbb{Z})(0 \leq f_0 < 9 \rightarrow (\forall\ c_0 : \mathbb{Z})(0 \leq c_0 < 9 \longrightarrow_L ((f_0 = f \land c_0 = c) \lor t[f_0][c_0] = t_0[f_0][c_0])))$

    }

}

9. `proc sudoku_esTableroParcialmenteResuelto` (in t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, out result: Bool) {

    `Pre` $\{esTableroValido(t)\}$

    `Post` $\{result = tableroSinRepetidos(t)\}$

}

10. `proc sudoku_esTableroTotalmenteResuelto` (in t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, out result: Bool) {

    `Pre` $\{esTableroValido(t)\}$

    `Post` $\{result = esTotalmenteResuelto(t)\}$

}

11. `proc sudoku_esSubTablero` (in t_0, t_1: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, out result: Bool) {

    `Pre` $\{esTableroValido(t_0) \land esTableroValido(t_1)\}$

    `Post` $\{result = esSubTablero(t_0, t_1)\}$

}

12. `proc sudoku_tieneSolucion` (in t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, out tieneSolucion: Bool) {

    `Pre` $\{esTableroValido(t)\}$

    `Post` $\{tieneSolucion = tableroTieneSolucion(t)\}$

}

13. `proc sudoku_resolver` (inout t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, out tieneSolucion: Bool) {

    `Pre` $\{esTableroValido(t) \land t = t_0\}$

    `Post` {

        $(tieneSolucion \land_L esSubTablero(t_0, t) \land esTotalmeneResuelto(t)) \lor_L$
        $(\neg tieneSolucion \land_L \neg tableroTieneSolucion(t_0) \land (t = t_0))$

    }

}

14. `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` $\{src = target\}$

}

2. Predicados y Auxiliares generales

`pred esTamanoValido` (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

  $(\text{length}(t) = 9) \land (\forall\ f : \mathbb{Z})(0 \leq f < \text{length}(t) \longrightarrow_L \text{length}(t[f]) = 9)$

}

`pred elementosValidos` (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

  $(\forall\ f : \mathbb{Z})(0 \leq f < \text{length}(t) \rightarrow (\forall\ c : \mathbb{Z})(0 \leq c < \text{length}(t[f]) \longrightarrow_L 0 \leq t[f][c] \leq 9))$

}

`pred esTableroValido` (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

$\quad esTamanoValido(t) \land elementosValidos(t)$

}

`fun cantCeldasVacias` (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) : $\mathbb{Z} \ = \sum_{f=0}^{\mathsf{length}(t)-1}(\sum_{c=0}^{\mathsf{length}(t[f])-1} \mathsf{if}\ t[f][c] = 0\ \mathsf{then}\ 1\ \mathsf{else}\ 0\ \mathsf{fi})$;

`pred filasSinRepetidos` (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

$\quad (\forall\, f : \mathbb{Z})(0 \le f < 9 \rightarrow$

$\quad \neg((\exists\, c_0 : \mathbb{Z})(0 \le c_0 < 9 \land (\exists\, c_1 : \mathbb{Z})((0 \le c_1 < 9 \land c_0 \neq c_1) \land_L \neg(t[f][c_0] = 0 \land t[f][c_1] = 0) \land t[f][c_0] = t[f][c_1]))))$

}

`pred columnasSinRepetidos` (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

$\quad (\forall\, c : \mathbb{Z})(0 \le c < 9 \rightarrow$

$\quad \neg((\exists\, f_0 : \mathbb{Z})(0 \le f_0 < 9 \land (\exists\, f_1 : \mathbb{Z})((0 \le f_1 < 9 \land f_0 \neq f_1) \land_L \neg(t[f_0][c] = 0 \land t[f_1][c] = 0) \land t[f_0][c] = t[f_1][c]))))$

}

/ Devuelve la fila superior (primer fila) de una región comprendida entre 1 y 9 */*
`fun desdeFila` (r: $\mathbb{Z}$) : $\mathbb{Z} \ = ((r-1)\ \mathsf{div}\ 3) * 3$;

/ Devuelve la primer columna de una región comprendida entre 1 y 9 */*
`fun desdeColumna` (r: $\mathbb{Z}$) : $\mathbb{Z} \ = ((r-1)\ \mathsf{mod}\ 3) * 3$;

`pred esRegionSinRepetidos` (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, r: $\mathbb{Z}$) {

$\quad \neg((\exists\, f_0 : \mathbb{Z})(desdeFila(r) \le f_0 < desdeFila(r) + 3 \land (\exists\, c_0 : \mathbb{Z})(desdeCol(r) \le c_0 < desdeCol(r) + 3 \land$

$\quad (\exists\, f_1 : \mathbb{Z})(desdeFila(r) \le f_1 < desdeFila(r) + 3 \land (\exists\, c_1 : \mathbb{Z})(desdeCol(r) \le c_1 < desdeCol(r) + 3 \land$

$\quad (f_0 \neq f_1) \land (c_0 \neq c_1) \land_L \neg(t[f_0][c_0] = 0 \land t[f_1][c_1] = 0) \land t[f_0][c_0] = t[f_1][c_1])))))$

}

`pred regionesSinRepetidos` (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

$\quad (\forall\, r : \mathbb{Z})(1 \le r \le 9 \longrightarrow_L esRegionSinRepetidos(t, r))$

}

`pred tableroSinRepetidos` (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

$\quad filasSinRepetidos(t) \land columnasSinRepetidos(t) \land regionesSinRepetidos(t)$

}

`pred esTotalmenteResuelto` (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

$\quad (cantCeldasVacias(t) = 0) \land tableroSinRepetidos(t)$

}

`pred esSubTablero` ($\mathrm{t}_0, \mathrm{t}_1$: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

$\quad (\forall\, f : \mathbb{Z})(0 \le f < 9 \longrightarrow_L (\forall\, c : \mathbb{Z})(0 \le c < 9 \longrightarrow_L t_0[f][c] = 0 \lor \mathrm{t}_0[f][c] = t_1[f][c]))$

}

`pred tableroTieneSolucion` (t: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {

$\quad (\exists\, t_0 : seq\langle seq\langle \mathbb{Z}\rangle\rangle)(esTotalmenteResuelto(t_0) \land esSubTablero(t, t_0))$

}

3. Decisiones tomadas

- En el ejercicio 3, se considera como tablero totalmente resuelto a un tablero sin celdas vacias y no necesariamente resuelto.
- Para el predicado `regionesSinRepetidos` (Ejercicio 9) se tomó la decisión de enumerar las regiones del tablero del 1 al 9, comenzando desde la parte superior izquierda del tablero hasta la parte inferior derecha.
- En los ejercicios 13 y 14 se decidió que la especificación considere el tablero de entrada como válido, considerando que se trabaja en el contexto de tableros de dimensión 9x9 (sudokus).