

Trabajo Práctico

Ingeniería de Software II

Ignacio Cantore

Licenciatura en Ciencias de la Computación
Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Requerimientos

Un consultorio odontológico necesita asignar turnos a sus pacientes. Para esto, primero deben estar registrados en el sistema, con su DNI y nombre completo. Luego, pueden elegir una fecha y hora para ser atendidos. Dado que algunos tratamientos suelen realizarse en varias sesiones, es frecuente que los pacientes tengan reservados varios turnos.

Las operaciones requeridas son:

- Alta y baja de pacientes
- Alta y baja de turnos
- Dado el DNI de un paciente, consultar los turnos que tiene asignados
- Dadas una fecha y hora, consultar a qué paciente corresponde dicho turno

Especificación

Se dan las siguientes designaciones:

d es el DNI de un paciente $\approx d \in DNI$
 n es el nombre de un paciente $\approx n \in NOMBRE$
 f es una fecha $\approx f \in FECHA$
 h es una hora $\approx h \in HORA$
 El nombre del paciente con DNI $d \approx pacientes\ d$
 El DNI del paciente que tiene asignado un turno el día f en el horario $h \approx turnos\ (f, h)$

Luego, se introducen los tipos a utilizar en la especificación:

$[DNI, NOMBRE, FECHA, HORA]$

Se define el espacio de estados de la especificación y su estado inicial:

$Consultorio$
$turnos : FECHA \times HORA \rightarrow DNI$ $pacientes : DNI \rightarrow NOMBRE$
$ConsultorioInit$
$Consultorio$
$turnos = \emptyset$ $pacientes = \emptyset$

Como todos los turnos asignados corresponden a un paciente registrado en el sistema, debe valer el siguiente invariante:

$ConsultorioInv$
$Consultorio$
$ran\ turnos \subseteq dom\ pacientes$

Ahora se modelan las operaciones del sistema, empezando con la operación para agregar un paciente:

<i>AltaPacienteOk</i>
$\Delta \text{Consultorio}$
$d? : DNI$
$n? : NOMBRE$
$d? \notin \text{dom } \text{pacientes}$
$\text{pacientes}' = \text{pacientes} \cup \{d? \mapsto n?\}$
$\text{turnos}' = \text{turnos}$

<i>PacienteYaExiste</i>
$\Xi \text{Consultorio}$
$d? : DNI$
$d? \in \text{dom } \text{pacientes}$

$$\text{AltaPaciente} == \text{AltaPacienteOk} \vee \text{PacienteYaExiste}$$

Luego, la operación para dar de baja un paciente, considerando el caso en que el paciente aún tiene turnos asignados:

<i>BajaPacienteOk</i>
$\Delta \text{Consultorio}$
$d? : DNI$
$d? \in \text{dom } \text{pacientes}$
$d? \notin \text{ran } \text{turnos}$
$\text{pacientes}' = \{d?\} \triangleleft \text{pacientes}$
$\text{turnos}' = \text{turnos}$

<i>PacienteNoExiste</i>
$\Xi \text{Consultorio}$
$d? : DNI$
$d? \notin \text{dom } \text{pacientes}$

$PacienteTieneTurnos$
$\Xi Consultorio$
$d? : DNI$
$d? \in \text{ran } turnos$

$$BajaPaciente == BajaPacienteOk \vee PacienteNoExiste \vee PacienteTieneTurnos$$

Por otro lado, la operación para asignar un turno a un paciente:

$AltaTurnoOk$
$\Delta Consultorio$
$f? : FECHA$
$h? : HORA$
$d? : DNI$
$d? \in \text{dom } pacientes$
$(f?, h?) \notin \text{dom } turnos$
$turnos' = turnos \cup \{(f?, h?) \mapsto d?\}$
$pacientes' = pacientes$

$TurnoYaAsignado$
$\Xi Consultorio$
$f? : FECHA$
$h? : HORA$
$(f?, h?) \in \text{dom } turnos$

$$AltaTurno == AltaTurnoOk \vee PacienteNoExiste \vee TurnoYaAsignado$$

Y la operación para dar de baja un turno, teniendo en cuenta que se debe corroborar si el DNI brindado corresponde al turno en cuestión:

$BajaTurnoOk$ $\Delta Consultorio$ $f? : FECHA$ $h? : HORA$ $d? : DNI$
$(f?, h?) \in \text{dom } turnos$ $d? = turnos(f?, h?)$ $turnos' = \{(f?, h?)\} \triangleleft turnos$ $pacientes' = pacientes$

$TurnoNoAsignado$ $\Xi Consultorio$ $f? : FECHA$ $h? : HORA$
$(f?, h?) \notin \text{dom } turnos$

$TurnoNoCoincide$ $\Xi Consultorio$ $f? : FECHA$ $h? : HORA$ $d? : DNI$
$(f?, h?) \in \text{dom } turnos$ $d? \neq turnos(f?, h?)$

$$BajaTurno == BajaTurnoOk \vee TurnoNoAsignado \vee TurnoNoCoincide$$

Además, se define una operación para consultar todos los turnos de un paciente, así como una para consultar el nombre del paciente que tiene asignado un turno dado:

$TurnosPaciente$ $\Xi Consultorio$ $d? : DNI$ $t! : FECHA \leftrightarrow HORA$
$t! = \text{dom}(turnos \triangleright \{d?\})$

<i>NombreTurnoOk</i>
\exists <i>Consultorio</i>
$f? : FECHA$
$h? : HORA$
$n! : NOMBRE$
$(f?, h?) \in \text{dom } turnos$
$n! = \text{pacientes } (turnos (f?, h?))$

$$NombreTurno == NombreTurnoOk \vee TurnoNoAsignado$$

Simulaciones

Simulación no tipada

En la simulación se agrega un paciente al sistema, se le asigna un turno, se intenta borrar al paciente y se elimina el turno antes mencionado. Luego, se consultan los turnos de un paciente cuyo DNI no está en el sistema. Por último, se consulta el nombre del paciente de un turno no asignado:

```
consultorioInit(T1,P1) &
altaPaciente(T1,P1,dni:d40140078,nombre:'IgnacioCantore',T2,P2) &
altaTurno(T2,P2,fecha:f010323,hora:h1030,dni:d40140078,T3,P3) &
bajaPaciente(T3,P3,dni:d40140078,T4,P4) &
bajaTurno(T4,P4,fecha:f010323,hora:h1030,dni:d40140078,T5,P5) &
turnosPaciente(T5,P5,dni:20016929,Turnos,T5,P5) &
nombreTurno(T5,P5,fecha:f050323,hora:h1500,Nombre,T5,P5) .
```

Como puede observarse a continuación, la solución encontrada por $\{log\}$ es la esperada, donde en $P4$ no se borró al paciente del sistema ya que tiene un turno, $Turnos$ es vacío ya que no hay turnos asignados para el DNI dado, y para la variable $Nombre$ no se da un valor ya que para la fecha y hora dadas no hay un turno asignado.

```
T1 = {},
P1 = {},
T2 = {},
```

```

P2 = {[dni:d40140078,nombre:IgnacioCantore]},
T3 = {[fecha:f010323,hora:h1030],dni:d40140078}},
P3 = {[dni:d40140078,nombre:IgnacioCantore]},
T4 = {[fecha:f010323,hora:h1030],dni:d40140078}},
P4 = {[dni:d40140078,nombre:IgnacioCantore]},
T5 = {},
P5 = {[dni:d40140078,nombre:IgnacioCantore]},
Turnos = {}

```

Simulación tipada

En la simulación se agrega un paciente al sistema, se le asignan dos turnos, se da de alta un segundo paciente, se intenta asignarle un turno que está reservado por el primer paciente, y se intenta eliminar el turno. Luego, se consulta el nombre del paciente correspondiente al turno antes mencionado. Por último, se consultan los turnos del primer paciente:

```

consultorioInit(T1,P1) &
altaPaciente(T1,P1,dni:d40140078,nombre:'IgnacioCantore',T2,P2) &
altaTurno(T2,P2,fecha:f010323,hora:h1030,dni:d40140078,T3,P3) &
altaTurno(T3,P3,fecha:f100323,hora:h1215,dni:d40140078,T4,P4) &
altaPaciente(T4,P4,dni:d40769801,nombre:'EloyYachini',T5,P5) &
altaTurno(T5,P5,fecha:f010323,hora:h1030,dni:d40769801,T6,P6) &
bajaTurno(T6,P6,fecha:f010223,hora:h1030,dni:d20016425,T7,P7) &
nombreTurno(T7,P7,fecha:f010323,hora:h1030,Nombre,T7,P7) &
turnosPaciente(T7,P7,dni:d40140078,Turnos,T7,P7) &
dec([T1,T2,T3,T4,T5,T6,T7],tur) & dec([P1,P2,P3,P4,P5,P6,P7],pac) &
dec(Turnos,set(fh)) & dec(Nombre,nombre).

```

Como puede observarse a continuación, la solución encontrada por $\{log\}$ es la esperada, donde en $T6$ no se agregó el turno ya que estaba reservado, y en $T7$ no se borró el turno porque el DNI brindado no corresponde al del turno.

```

T1 = {},
P1 = {},
T2 = {},
P2 = {[dni:d40140078,nombre:IgnacioCantore]},
T3 = {[fecha:f010323,hora:h1030],dni:d40140078}},

```

```

P3 = {[dni:d40140078,nombre:IgnacioCantore]},
T4 = {[fecha:f010323,hora:h1030],dni:d40140078},
      [[fecha:f100323,hora:h1215],dni:d40140078]},
P4 = {[dni:d40140078,nombre:IgnacioCantore]},
T5 = {[fecha:f010323,hora:h1030],dni:d40140078},
      [[fecha:f100323,hora:h1215],dni:d40140078]},
P5 = {[dni:d40140078,nombre:IgnacioCantore],
      [dni:d40769801,nombre:EloyYachini]},
T6 = {[fecha:f010323,hora:h1030],dni:d40140078},
      [[fecha:f100323,hora:h1215],dni:d40140078]},
P6 = {[dni:d40140078,nombre:IgnacioCantore],
      [dni:d40769801,nombre:EloyYachini]},
T7 = {[fecha:f010323,hora:h1030],dni:d40140078},
      [[fecha:f100323,hora:h1215],dni:d40140078]},
P7 = {[dni:d40140078,nombre:IgnacioCantore],
      [dni:d40769801,nombre:EloyYachini]},
Nombre = nombre:IgnacioCantore,
Turnos = {[fecha:f010323,hora:h1030],[fecha:f100323,hora:h1215]}

```

Demostraciones con $\{log\}$

Primera demostración con $\{log\}$. Se demuestra que *BajaPaciente* preserva el invariante *ConsultorioInv*, lo cual equivale al siguiente teorema:

theorem BajaPacientePI
 $ConsultorioInv \wedge BajaPaciente \Rightarrow ConsultorioInv'$

cuya negación se escribe en $\{log\}$ de la siguiente manera:

```

dec([T,T_],tur) & dec([P,P_],pac) & dec(D,dni) &
consultorioInv(T,P) &
bajaPaciente(T,P,D,T_,P_) &
dec([D1,D2],set(dni)) & ran(T_,D1) & dom(P_,D2) & nsubset(D1,D2).

```

Como $\{log\}$ no encuentra ninguna solución, la fórmula es insatisfacible, lo que demuestra que *BajaPaciente* preserva el invariante *ConsultorioInv*

Segunda demostración con $\{log\}$. Se demuestra que *AltaTurno* preserva el invariante $turnos \in _ \rightarrow _$, es decir:

theorem TurnosIsPFun

$$turnos \in _ \rightarrow _ \wedge AltaTurno \Rightarrow turnos' \in _ \rightarrow _$$

cuya negación se escribe en $\{log\}$ de la siguiente manera:

```
dec([T1,T2],tur) & dec([P1,P2],pac) & dec(F,fecha) &
dec(H,hora) & dec(D,dni) &
pfun(T1) &
altaTurno(T1,P1,F,H,D,T2,P2) &
npfun(T2) .
```

Como $\{log\}$ no encuentra ninguna solución, la fórmula es insatisfacible, lo que demuestra que *AltaTurno* preserva el invariante $turnos \in _ \rightarrow _$

Demostración con Z/EVES

Se demuestra que *AltaPaciente* preserva el invariante *ConsultorioInv*, lo cual equivale al siguiente teorema:

theorem AltaPacientePI

$$ConsultorioInv \wedge AltaPaciente \Rightarrow ConsultorioInv'$$

proof[*AltaPacientePI*]

```
invoke AltaPaciente;
split AltaPacienteOk;
cases;
prove by reduce;
next;
prove by reduce;
next;
```

■

El comando **invoke** reemplaza *AltaPaciente* en el antecedente por su definición. Luego, **split** *AltaPacienteOk* genera dos casos: uno donde se satisfacen las condiciones de *AltaPacienteOk* y otro donde no. **cases** separa estos casos para poder demostrarlos individualmente. El primer caso se demuestra usando **prove by reduce**. Luego, se avanza al segundo caso con **next**. Se procede nuevamente con **prove by reduce**. Finalmente, el comando **next** intenta avanzar al siguiente caso, y al no haber más se completa la prueba.

Casos de prueba

Se generan casos de prueba para la operación *BajaTurno*, para lo cual se utilizan los siguientes comandos de Fastest:

```
loadspect fastest.tex
selop BajaTurno
genalltt
addtactic BajaTurno_DNF_1 SP \ndres \{(f?,h?)\} \ndres turnos
genalltt
genalltca
showtt
showsched -tca
```

Es decir, se carga la especificación, se selecciona la operación y se genera el árbol de pruebas, el cual aplica la táctica de testing FND. Ésta la separa en tres clases: *BajaTurno_DNF_1*, *BajaTurno_DNF_2* y *BajaTurno_DNF_3*, las cuales corresponden a *BajaTurnoOk*, *TurnoNoAsignado* y *TurnoNoCoincide*, respectivamente.

Luego, se agrega la táctica de Particiones Estándar para el operador \Leftarrow sobre la expresión $\{(f?, h?)\} \Leftarrow turnos$ en *BajaTurno_DNF_1*. Se genera nuevamente el árbol para aplicar esta táctica, y se generan los casos de prueba, además de podar las clases de prueba insatisfacibles.

Finalmente, con `showtt` se puede ver el árbol de pruebas generado, donde las hojas corresponden a los casos de prueba generados:

```
BajaTurno_VIS
!_____BajaTurno_DNF_1
|   !_____BajaTurno_SP_3
|   |   !_____BajaTurno_SP_3_TCASE
|   |
|   !_____BajaTurno_SP_4
|       !_____BajaTurno_SP_4_TCASE
|
|
|_____BajaTurno_DNF_2
|   !_____BajaTurno_DNF_2_TCASE
|
```

```
!_____BajaTurno_DNF_3
  !_____BajaTurno_DNF_3_TCASE
```

Y con `showsch -tca` se obtienen los esquemas Z correspondientes a los casos de prueba generados:

$\frac{BajaTurno_SP_3_TCASE}{BajaTurno_SP_3}$ $f? = fECHA1$ $h? = hORA2$ $pacientes = \emptyset$ $turnos = \{((fECHA1 \mapsto hORA2) \mapsto dNI3)\}$ $d? = dNI3$
--

$\frac{BajaTurno_SP_4_TCASE}{BajaTurno_SP_4}$ $f? = fECHA1$ $h? = hORA2$ $pacientes = \emptyset$ $turnos = \{((fECHA1 \mapsto hORA2) \mapsto dNI3), ((fECHA1 \mapsto hORA4) \mapsto dNI5)\}$ $d? = dNI3$

$\frac{BajaTurno_DNF_2_TCASE}{BajaTurno_DNF_2}$ $f? = fECHA1$ $h? = hORA2$ $pacientes = \emptyset$ $turnos = \emptyset$ $d? = dNI3$
--

<i>BajaTurno_DNF_3_TCASE</i>	
<i>BajaTurno_DNF_3</i>	
<i>f?</i> = <i>fECHA3</i>	
<i>h?</i> = <i>hORA1</i>	
<i>pacientes</i> = \emptyset	
<i>turnos</i> = $\{((fECHA1 \mapsto hORA1) \mapsto dNI2)\}$	
<i>d?</i> = <i>dNI1</i>	