

TP Final Ing Software II

Nicolás, Uriel Navall

Legajo: N-1159/2

May 2022

Requerimientos

Se tiene una libreta de un paseador de perros en la cual se agenda las salidas de este para pasear perros. El paseador solo puede pasear cierta cantidad de perros a la vez, dependiendo esto de los tamaños de los perros a pasear. Además el paseador no quiere pasear más de un perro de raza conflictiva a la vez, ni quiere pasear perros de diferentes sexos al mismo tiempo.

Se necesitan operaciones para poder agregar perros a un turno, sacar perros de un turno y obtener los perros a pasear dado un turno.

Especificación

Se comienza dando las siguientes designaciones:

n es el nombre de un perro $\approx n \in NOMBRE$

f representa un día en particular $\approx f \in FECHA$

h es una hora del día $\approx h \in HORA$

r es una raza de perro $\approx r \in RAZA$

s representa un sexo $\approx s \in SEXO$

t es el tamaño de un perro $\approx t \in TAMANO$

Conjunto de perros a pasear en el turno asociado a la fecha f y la hora $h \approx agenda(f,h)$

Luego introducimos los tipos a utilizar en la especificación

$[NOMBRE, FECHA, HORA, RAZA]$

$SEXO ::= hembra \mid macho$

$TAMANO ::= chico \mid medio \mid grande$

Presentaremos la siguiente definición axiomática para representar el "peso" sumado máximo de perros que el paseador puede pasear al mismo tiempo.

$\mid MAX_PESO : \mathbb{N}$

Además también utilizaremos la definición axiomática *razas_peligrosas* para representar el conjunto de razas de perros que el paseador considera peligrosa.

$\mid razas_peligrosas : \mathbb{P} RAZA$

También será necesario definir el tipo esquema *Perro* y lo hacemos de la siguiente manera:

| |
|---|
| $Perro$ $nombre : NOMBRE$ $raza : RAZA$ $tamano : TAMANO$ $sexo : SEXO$ |
|---|

Ahora podemos definir el espacio de estados de la especificación y su estado inicial:

| |
|---|
| $Sist$ $agenda : FECHA \times HORA \rightarrow \mathbb{P} Perro$ |
|---|

| | |
|----------------------|-------|
| $SistInit$ | _____ |
| $Sist$ | _____ |
| $agenda = \emptyset$ | _____ |

Como no tiene sentido tener guardado en la agenda un turno en la que no haya ningún perro (que surge al eliminar el único perro de un turno) pasaremos entonces a eliminar los turnos cuando se elimina el último perro de este, lo que deriva en la siguiente invariante de estado que nuestra especificación deberá cumplir:

| | |
|---|-------|
| $SistInv$ | _____ |
| $Sist$ | _____ |
| $\emptyset \notin (\text{ran } agenda)$ | _____ |

Comenzaremos ahora a modelar las operaciones de nuestro sistema empezando con la operación para agregar un perro a un turno. Se definiría primero el esquema del caso exitoso y luego los esquemas que corresponden a los errores, para al final integrar todo en un mismo esquema.

| | |
|---|-------|
| $AgregarPerroSalidaOk$ | _____ |
| $\Delta Sist$ | _____ |
| $p? : Perro$ | _____ |
| $f? : FECHA$ | _____ |
| $h? : HORA$ | _____ |
| $(f?, h?) \in (\text{dom } agenda) \wedge p? \notin (agenda(f?, h?))$ $p?.raza \notin razas_peligrosas \vee$ $razas_peligrosas \cap \{p : Perro \mid p \in (agenda(f?, h?)) \bullet p.raza\} = \emptyset$ $p?.sexo \in \{x : Perro \mid x \in (agenda(f?, h?)) \bullet x.sexo\}$ $\#\{x : Perro \mid x \in (agenda'(f?, h?)) \wedge x.tamano = chico \bullet x\} +$ $\#\{x : Perro \mid x \in (agenda'(f?, h?)) \wedge x.tamano = medio \bullet x\} * 2 +$ $\#\{x : Perro \mid x \in (agenda'(f?, h?)) \wedge x.tamano = grande \bullet x\} * 3 < MAX_PESO$ $agenda' = agenda \oplus \{(f?, h?) \mapsto (agenda(f?, h?)) \cup \{p?\}\}$ | _____ |

AgregarPerroSalidaTurnoNuevoOk _____

$\Delta Sist$

$p? : Perro$

$f? : FECHA$

$h? : HORA$

$(f?, h?) \notin \text{dom } agenda$

$\# \{x : Perro \mid x \in (agenda'(f?, h?)) \wedge x.tamano = chico \bullet x\} +$

$\# \{x : Perro \mid x \in (agenda'(f?, h?)) \wedge x.tamano = medio \bullet x\} * 2 +$

$\# \{x : Perro \mid x \in (agenda'(f?, h?)) \wedge x.tamano = grande \bullet x\} * 3 < MAX_PESO$

$agenda' = agenda \oplus \{(f?, h?) \mapsto \{p?\}\}$

AgregarPerroSalidaYaExiste _____

$\Xi Sist$

$p? : Perro$

$f? : FECHA$

$h? : HORA$

$(f?, h?) \in \text{dom } agenda \wedge p? \in agenda(f?, h?)$

AgregarPerroSalidaConflictoRaza _____

$\Xi Sist$

$p? : Perro$

$f? : FECHA$

$h? : HORA$

$(f?, h?) \in (\text{dom } agenda) \wedge p? \notin (agenda(f?, h?))$

$p?.raza \in razas_peligrosas \wedge razas_peligrosas \cap \{p : agenda(f?, h?) \bullet p.raza\} \neq \emptyset$

AgregarPerroSalidaConflictoSexo _____

$\Xi Sist$

$p? : Perro$

$f? : FECHA$

$h? : HORA$

$(f?, h?) \in (\text{dom } agenda) \wedge p? \notin (agenda(f?, h?))$

$p?.sexo \notin \{x : agenda(f?, h?) \bullet x.sexo\}$

| | |
|--|--|
| $\frac{\text{AgregarPerroSalidaMuyPesado}}{\Xi \text{Sist}$ $p? : \text{Perro}$ $f? : \text{FECHA}$ $h? : \text{HORA}$ | |
| $(f?, h?) \in (\text{dom agenda}) \wedge p? \notin (\text{agenda}(f?, h?))$ $\# \{x : \text{Perro} \mid x \in (\text{agenda}(f?, h?)) \cup \{p?\} \wedge x.tamano = \text{chico} \bullet x\} +$ $\# \{x : \text{Perro} \mid x \in (\text{agenda}(f?, h?)) \cup \{p?\} \wedge x.tamano = \text{medio} \bullet x\} * 2 +$ $\# \{x : \text{Perro} \mid x \in (\text{agenda}(f?, h?)) \cup \{p?\} \wedge x.tamano = \text{grande} \bullet x\} * 3 \geq \text{MAX_PESO}$ | |

$$\text{AgregarPerroSalida} \hat{=} \text{AgregarPerroSalidaOk} \vee \text{AgregarPerroSalidaTurnoNuevoOk} \vee$$

$$\text{AgregarPerroSalidaYaExiste} \vee \text{AgregarPerroSalidaConflictoRaza} \vee$$

$$\text{AgregarPerroSalidaConflictoSexo} \vee \text{AgregarPerroSalidaMuyPesado}$$

Ahora definiremos la operación para remover a un perro de un turno:

| | |
|---|--|
| $\frac{\text{QuitarPerroSalidaOk}}{\Delta \text{Sist}$ $p? : \text{Perro}$ $f? : \text{FECHA}$ $h? : \text{HORA}$ | |
| $(f?, h?) \in \text{dom agenda}$ $p? \in \text{agenda}(f?, h?)$ $\text{agenda}(f?, h?) \neq \{p?\}$ $\text{agenda}' = \text{agenda} \oplus \{(f?, h?) \mapsto \text{agenda}(f?, h?) \setminus \{p?\}\}$ | |

| | |
|--|--|
| $\frac{\text{QuitarPerroSalidaYTurnoOk}}{\Delta \text{Sist}$ $p? : \text{Perro}$ $f? : \text{FECHA}$ $h? : \text{HORA}$ | |
| $(f?, h?) \in \text{dom agenda}$ $p? \in \text{agenda}(f?, h?)$ $\text{agenda}(f?, h?) = \{p?\}$ $\text{agenda}' = \{(f?, h?)\} \triangleleft \text{agenda}$ | |

| | |
|---|-------|
| $TurnoNoExiste$ | _____ |
| $\Xi Sist$ $f? : FECHA$ $h? : HORA$ | |
| $(f?, h?) \notin \text{dom } agenda$ | |

| | |
|---|-------|
| $QuitarPerroSalidaNoExistePerro$ | _____ |
| $\Xi Sist$ $p? : Perro$ $f? : FECHA$ $h? : HORA$ | |
| $(f?, h?) \in \text{dom } agenda$ $p? \notin (agenda(f?, h?))$ | |

$$QuitarPerroSalida \hat{=} QuitarPerroSalidaOk \vee QuitarPerroSalidaYTurnoOk \vee TurnoNoExiste \vee QuitarPerroSalidaNoExistePerro$$

Y por último definiremos una operación que nos devuelve los perros a pasear de un turno dado:

| | |
|--|-------|
| $PerrosAPasearOk$ | _____ |
| $\Xi Sist$ $f? : FECHA$ $h? : HORA$ $prs! : \mathbb{P} Perro$ | |
| $(f?, h?) \in \text{dom } agenda$ $prs! = agenda(f?, h?)$ | |

$$PerrosAPasear \hat{=} PerrosAPasearOk \vee TurnoNoExiste$$

Simulaciones

Simulación no tipada

En la simulación agrego un par de perros en un mismo turno, y luego intento agregar uno de los perros ya ingresados en el mismo turno. Luego agrego este perro en otro turno e intento eliminar un perro que no está en este último turno. Intento devolver los perros de un turno que no existe y para terminar devuelvo los perros del último turno que modifiqué.

Simulación:

```
sistInit(Agenda1) &
Razas_Peligrosas = {pitbull} & Max_Peso = 10 &
agregarPerroSalida(Razas_Peligrosas, Max_Peso, Agenda1, Agenda2,
[yill,ovejero_aleman,macho,medio], 34557274,1725892) &
agregarPerroSalida(Razas_Peligrosas, Max_Peso, Agenda2, Agenda3,
[zack,golden_retriever,macho,medio], 34557274,1725892) &
agregarPerroSalida(Razas_Peligrosas, Max_Peso, Agenda3, Agenda4,
[zack,golden_retriever,macho,medio], 34557274,1725892) &
agregarPerroSalida(Razas_Peligrosas, Max_Peso, Agenda4, Agenda5,
[zack,golden_retriever,macho,medio], 204986,246892) &
quitarPerroSalida(Agenda5,Agenda6,[yill,ovejero_aleman,macho,medio],
204986,246892) &
perrosAPasear(Agenda6,620867,240876,Prs_o1) &
perrosAPasear(Agenda6,204986,246892,Prs_o2).
```

En donde la primer respuesta fue la siguiente:

```
Agenda1 = {},
Razas_Peligrosas = {pitbull},
Max_Peso = 10,
Agenda2 = {[34557274,1725892],[yill,ovejero_aleman,macho,medio]}},
Agenda3 = {[34557274,1725892],[yill,ovejero_aleman,macho,medio],
[zack,golden_retriever,macho,medio]}},
Agenda4 = {[34557274,1725892],[yill,ovejero_aleman,macho,medio],
[zack,golden_retriever,macho,medio]}},
Agenda5 = {[204986,246892],[zack,golden_retriever,macho,medio]},
[[34557274,1725892],[yill,ovejero_aleman,macho,medio],
```

```
[zack,golden_retriever,macho,medio]]}],
    Agenda6 = {[[204986,246892],[[zack,golden_retriever,macho,medio]]],
[[34557274,1725892],[[yill,ovejero_aleman,macho,medio],
[zack,golden_retriever,macho,medio]]]},
    Prs_o2 = {[zack,golden_retriever,macho,medio]}
```

Simulación tipada

En la simulación agrego un par de perros en un mismo turno, luego agrego otro perro en otro y quito el primer perro que agregue en el turno original, para luego agregar el perro del segundo turno en el primero también. Por último pido la lista de perros del primer turno al que agregue perros.

Simulación:

```
sistInit(Agenda1) &
Razas_Peligrosas = {raza:pitbull} & Max_Peso = 10 &
agregarPerroSalida(Razas_Peligrosas,Max_Peso,Agenda1,Agenda2,
[nombre:mora,raza:caniche,hembra,chico], fecha:249862,hora:293847) &
agregarPerroSalida(Razas_Peligrosas,Max_Peso,Agenda2,Agenda3,
[nombre:pelusa,raza:san_bernardo,hembra,grande],fecha:249862,hora:293847) &
agregarPerroSalida(Razas_Peligrosas,Max_Peso,Agenda3,Agenda4,
[nombre:zack,raza:golden_retriever,macho,medio],fecha:456982,hora:3583) &
quitarPerroSalida(Agenda4,Agenda5,[nombre:mora,raza:caniche,hembra,
chico],fecha:249862,hora:293847) &
agregarPerroSalida(Razas_Peligrosas,Max_Peso,Agenda5,Agenda6,
[nombre:zack,raza:golden_retriever,macho, medio],fecha:249862,hora:293847) &
perrosAPasear(Agenda6,fecha:249862,hora:293847,Prs_o) &
dec([Agenda1,Agenda2,Agenda3,Agenda4,Agenda5,Agenda6],ag) &
dec(Prs_o,set(pr)) & dec(Razas_Peligrosas,set(raza)) & dec(Max_Peso,int).
```

En donde la primer respuesta fue la siguiente:

```
Agenda1 = {},
Razas_Peligrosas = {raza:pitbull},
Max_Peso = 10,
Agenda2 = {[[fecha:249862,hora:293847],[[nombre:mora,raza:caniche,hembra,
chico]]]},
Agenda3 = {[[fecha:249862,hora:293847],[[nombre:mora,raza:caniche,hembra,
```



```

chico], [nombre: pelusa, raza: san_bernardo, hembra, grande]]}],
  Agenda4 = {[[fecha: 456982, hora: 3583], {[nombre: zack, raza: golden_retriever,
macho, medio]}], [[fecha: 249862, hora: 293847], {[nombre: mora, raza: caniche, hembra,
chico], [nombre: pelusa, raza: san_bernardo, hembra, grande]}]}],
  Agenda5 = {[[fecha: 456982, hora: 3583], {[nombre: zack, raza: golden_retriever,
macho, medio]}], [[fecha: 249862, hora: 293847], {[nombre: pelusa, raza: san_bernardo,
hembra, grande]}]}],
  Agenda6 = {[[fecha: 249862, hora: 293847], {[nombre: pelusa, raza: san_bernardo,
hembra, grande], [nombre: zack, raza: golden_retriever, macho, medio]}], [[fecha: 456982,
hora: 3583], {[nombre: zack, raza: golden_retriever, macho, medio]}]}],
  Prs_o = {[nombre: pelusa, raza: san_bernardo, hembra, grande], [nombre: zack, ra
za: golden_retriever, macho, medio]}

```

Demostraciones en $\{log\}$

Demostración n°1

Se demostrará a continuación que la operación *AgregarPerroSalida* preserva la invariante *SistInv*, es decir:

theorem *AgregarPerroSalidaPI*
 $SistInv \wedge AgregarPerroSalida \Rightarrow SistInv'$

Teorema el cual se escribe en $\{log\}$ de la siguiente manera:

```

sistInv(Agenda) &
agregarPerroSalida(Razas_Peligrosas, Max_Peso, Agenda, Agenda_, P_i, F_i, H_i) &
ran(Agenda_, R) & {} in R &
dec([Agenda, Agenda_], ag) & dec(P_i, pr) & dec(F_i, fecha) & dec(H_i, hora) &
dec(Razas_Peligrosas, set(raza)) & dec(R, set(set(pr))) & dec(Max_Peso, int).

```

Demostración n°2

Se demostrará a continuación que la operación *QuitarPerroSalida* preserva el invariante $agenda \in _ \rightarrow _$. Lo que se traduce al siguiente teorema:

theorem AgendaIsPFun

$agenda \in _ \rightarrow _ \wedge \text{QuitarPerroSalida} \Rightarrow agenda' \in _ \rightarrow _$

Teorema el cual se escribe en $\{log\}$ de la siguiente manera:

```
pfun(Agenda) &
quitarPerroSalida(Agenda,Agenda_,P_i,F_i,H_i) &
npfun(Agenda_) &
dec([Agenda,Agenda_],ag) & dec(P_i,pr) & dec(F_i,fecha)&
dec(H_i,hora).
```

La respuesta al correr los teoremas es no, demostrando así los teoremas presentados.

Demostración en Z/EVES

theorem notInRanSubset $[X, Y]$

$\forall f : X \leftrightarrow Y; y : Y; C : \mathbb{P} X \bullet y \notin (\text{ran } f) \Rightarrow y \notin \text{ran}(C \triangleleft f)$

proof[*notInRanSubset*]

prove;

apply ranDefinition;

prove;

■

Casos de prueba

Se busca generar casos de prueba para la operación *AgregarPerroSalida*. Para obtenerlos se utilizarán los siguientes comandos en *Fastest*:

```
loadspect fastest.tex
selop AgregarPerroSalida
genalltt
addtactic AgregarPerroSalida_DNF_2 SP \cap razas\_ peligrosas \cap
\{ p : Perro | p \in agenda ( f? , h? ) @ p . raza \} = \{ \}
genalltt
genalltca
showsche -tcl -o test_classes.txt
```

En primera instancia aplicamos la táctica DNF que separara la operación en las clases *AgregarPerroSalida_DNF₁*, *AgregarPerroSalida_DNF₂*, *AgregarPerroSalida_DNF₃*, *AgregarPerroSalida_DNF₄*, *AgregarPerroSalida_DNF₅*, *AgregarPerroSalida_DNF₆*, *AgregarPerroSalida_DNF₇*.

Luego aplicamos la técnica de partición estándar sobre \cap en la expresión $razas_peligrosas \cap \{p : agenda(f?, h?) \bullet p.raza\} \neq \{\}$ para particionar la clase de prueba *AgregarPerroSalida_DNF₂*.

Y de esta manera, obtenemos el árbol de clases de prueba generado por Fastest que puede observarse en la Figura 1.

Por errores de java al ejecutar el comando *genalltca* en Fastest no se pudo obtener casos de prueba.

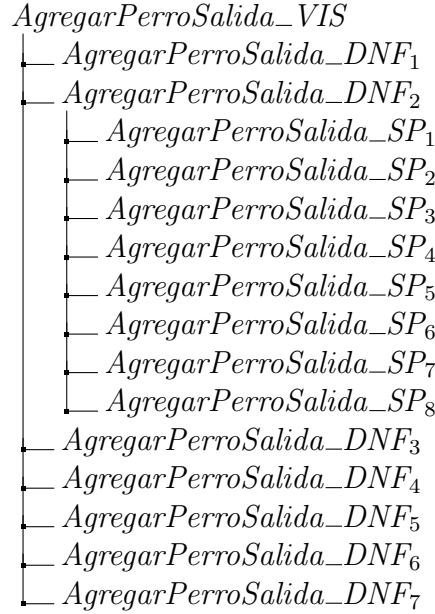


Figura 1: Árbol de clases de prueba.