

Trabajo Práctico para Ingeniería de Software

Maximiliano Cristiá

1. Requerimientos

El sistema Agenda de Cumpleaños permite mantener un registro de los cumpleaños de los conocidos del usuario.

El sistema debe permitir asociar un nombre de persona con su fecha de cumpleaños.

Además debe permitir encontrar la fecha de cumpleaños de una persona dada y el nombre de todas las personas que cumplen años en una fecha dada.

2. Especificación

Comenzamos dando algunas designaciones.

n es un nombre $\approx n \in NAME$

d es una fecha $\approx d \in DATE$

k es el nombre de una persona cuyo cumpleaños hay registrar $\approx k \in known$

La fecha de cumpleaños de la persona $k \approx birthday\ k$

Entonces introducimos los siguientes tipos básicos.

$[NAME, DATE]$

Ahora podemos definir el espacio de estados de la especificación de la siguiente forma.

$BirthdayBook$

$known : \mathbb{P} NAME$

$birthday : NAME \rightarrow DATE$

El estado inicial de la agenda de cumpleaños es el siguiente.

$BirthdayBookInit$

$BirthdayBook$

$known = \emptyset$

$birthday = \emptyset$

El siguiente esquema describe los predicados que deberían ser invariantes de estado.

<i>BirthdayBookInv</i>
<i>BirthdayBook</i>
$known = \text{dom } birthday$

La primera operación que modelamos es cómo agregar una fecha de cumpleaños a la agenda. Como siempre modelamos primero el caso exitoso, luego los errores y finalmente integramos todo en una única expresión de esquemas.

<i>AddBirthdayOk</i>
$\Delta \text{BirthdayBook}$
$name? : \text{NAME}$
$date? : \text{DATE}$
$name? \notin \text{known}$
$\text{known}' = \text{known} \cup \{name?\}$
$birthday' = birthday \cup \{name? \mapsto date?\}$

<i>NameAlreadyExists</i>
$\exists \text{BirthdayBook}$
$name? : \text{NAME}$
$name? \in \text{known}$

$$\text{AddBirthday} == \text{AddBirthdayOk} \vee \text{NameAlreadyExists}$$

La segunda operación a especificar corresponde a mostrar el cumpleaños de una persona dada.

<i>FindBirthdayOk</i>
$\exists \text{BirthdayBook}$
$name? : \text{NAME}$
$date! : \text{DATE}$
$name? \in \text{known}$
$date! = birthday(name?)$

<i>NotAFriend</i>
$\exists \text{BirthdayBook}$
$name? : \text{NAME}$
$name? \notin \text{known}$

$$\text{FindBirthday} == \text{FindBirthdayOk} \vee \text{NotAFriend}$$

Finalmente tenemos una operación que nos lista los nombres de las personas cuya fecha de cumpleaños es hoy.

Remind

$\exists \text{BirthdayBook}$

$\text{today?} : \text{DATE}$

$\text{cards!} : \mathbb{P}\text{NAME}$

$\text{cards!} = \text{dom}(\text{birthday} \triangleright \{\text{today?}\})$

3. Simulaciones

La primera simulación es la siguiente:

```
birthdayBookInit(S0)           & addBirthday(S0,maxi,160367,S1) &
addBirthday(S1,'Yo',201166,S2)  & findBirthday(S2,'Yo',C,S3) &
addBirthday(S3,'Otro',201166,S4) & remind(S4,160367,Card,S5) &
remind(S5,201166,Card1,S6).
```

cuya primera respuesta es la siguiente:

```
S0 = {[known,{ }], [birthday,{ }]},
S1 = {[known,{maxi}], [birthday,{[maxi,160367]}]},
S2 = {[known,{maxi,Yo}], [birthday,{[maxi,160367],[Yo,201166]}]},
C = 201166,
S3 = {[known,{maxi,Yo}], [birthday,{[maxi,160367],[Yo,201166]}]},
S4 = {[known,{maxi,Yo,Otro}],
      [birthday,{[maxi,160367],[Yo,201166],[Otro,201166]}]},
Card = {maxi},
S5 = {[known,{maxi,Yo,Otro}],
      [birthday,{[maxi,160367],[Yo,201166],[Otro,201166]}]},
Card1 = {Yo,Otro},
S6 = {[known,{maxi,Yo,Otro}],
      [birthday,{[maxi,160367],[Yo,201166],[Otro,201166]}]}
```

La segunda simulación es la siguiente:

```
S0 = {[known,{maxi,caro,cami,alvaro}],
      [birthday,{[maxi,160367],[caro,201166],[cami,290697],[alvaro,110400]}]} &
addBirthday(S0,'Yo',160367,S1) & remind(S1,160367,Card,S1).
```

cuya primera respuesta es la siguiente:

```
S0 = {[known,{maxi,caro,cami,alvaro}],
      [birthday,{[maxi,160367],[caro,201166],[cami,290697],[alvaro,110400]}]},
S1 = {[known,{maxi,caro,cami,alvaro,Yo}],
      [birthday,{[maxi,160367],[caro,201166],[cami,290697],[alvaro,110400],
                  [Yo,160367]}]},
Card = {maxi,Yo}
```

4. Demostraciones con {log}

Primera demostración con {log}. Demuestro que *AddBirthday* preserva el invariante *BirthdayBookInv*, o sea el siguiente teorema:

theorem AddBirthdayPI

$$BirthdayBookInv \wedge AddBirthday \Rightarrow BirthdayBookInv'$$

el cual en {log} se escribe de la siguiente forma:

S = {[known,K],[birthday,B]} &
 S_ = {[known,K_],[birthday,B_]} &
 dom(B,K) &
 addBirthday(S,N,C,S_) &
 ndom(B_,K_).

Segunda demostración con {log}. Demuestro que *AddBirthday* preserva el invariante *birthday* \in *NAME* \rightarrow *DATE*, o sea el teorema:

theorem BirthdayIsPfun

$$birthday \in NAME \rightarrow DATE \wedge AddBirthday \Rightarrow birthday' \in NAME \rightarrow DATE$$

el cual en {log} se escribe de la siguiente forma:

S = {[known,K],[birthday,B]} &
 S_ = {[known,K_],[birthday,B_]} &
 dom(B,K) &
 pfun(B) &
 addBirthday(S,N,C,S_) &
 npfun(B_).

donde tuve que agregar como hipótesis que el dominio de *birthday* es igual a *known*.

5. Demostración con Z/EVES

theorem AddBirthdayPI

$$BirthdayBookInv \wedge AddBirthday \Rightarrow BirthdayBookInv'$$

proof[AddBirthdayPI]

invoke AddBirthday;
 split AddBirthdayOk;
 cases;
 prove by reduce;
 next;
 prove by reduce;
 next;

■

6. Casos de prueba

El script que usé para generar casos de prueba con Fastest es el siguiente:

```
loadspec /home/mcristia/fceia/is/material/setlog/bb.tex
selop AddBirthday
genalltt
addtactic AddBirthday_DNF_1 SP \cup birthday \cup \{name? \mapsto date?\}
genalltt
genalltca
```

Es decir que generé casos de prueba para la operación *AddBirthday* aplicando DNF y SP la expresión $birthday \cup \{name? \mapsto date?\}$ pero solo para particionar la clase de prueba *AddBirthday_DNF_1*.

De esta forma Fastest generó casos de prueba para todas las clases satisfacibles. Los casos de prueba son los siguientes:

AddBirthday_SP_2_TCASE

AddBirthday_SP_2

name? = *NAMENameInput*
known = \emptyset
birthday = \emptyset
date? = *DATEDateInput*

AddBirthday_SP_4_TCASE

AddBirthday_SP_4

name? = *NAMENameInput*
known = \emptyset
birthday = \emptyset
date? = *DATEDateInput*

AddBirthday_SP_6_TCASE

AddBirthday_SP_6

name? = *NAMENameInput*
known = \emptyset
birthday = $\{(NAMENameInput \mapsto DATEDateInput), (NAME1014 \mapsto DATE1020)\}$
date? = *DATEDateInput*

AddBirthday_SP_7_TCASE

AddBirthday_SP_7

name? = *NAMENameInput*
known = \emptyset
birthday = $\{(NAMENameInput \mapsto DATEDateInput)\}$
date? = *DATEDateInput*

AddBirthday_DNF_2_TCASE

AddBirthday_DNF_2

name? = *NAMEN**NameInput*

known = {*NAMEN**NameInput*}

birthday = \emptyset

date? = *DATED**DateInput*