EECS3342 Lab Assignment

Jinho Hwang (howden2@cse.yorku.ca)

25.03.2020

Revisions

Date	Revision	Description
25 March 2020	1.0	R, E descriptions, Event-B Model, Final code.

Contents

List of Figures

List of Tables

1 R-descriptions and E-descriptions

ENV1	There are known persons' name.	variable known
ENV2	Each known person's name is assigned to a birthday.	variable birthday
REQ3	The system can add a person's birthday record.	event AddBirthday
REQ4	The system can change a known person's birthday.	event ChangeBirthday
REQ5	The system can query a set of people who have a given birthday to send birthday cards.	event Remind
ENV6	There are predefined set of names.	carrier set NAME
ENV7	There are predefined set of birthdays.	carrier set NAME

2 Latex Documentation of Event-B models

2.1 Context

2.1.1 c0

CONTEXT c0

SETS

NAME ENV6 - There are predefined set of names.

DATE ENV7 - There are predefined set of birthdays.

END

2.2 Machine

2.2.1 m0

MACHINE m0

SEES c0

VARIABLES

known

birthday

cards_set

INVARIANTS

inv0_1: $known \in \mathbb{P}(NAME)$

ENV1 - There are birthdays for people.

inv0_2: $birthday \in NAME \nrightarrow DATE$

ENV2 - Each known person's name is assigned to a birthday.

inv0_3: $known = dom(birthday)$

inv0_4: $cards\_set \in \mathbb{P}(NAME)$

EVENTS

Initialisation

begin

act0_1: $known := \varnothing$

act0_2: $birthday := \varnothing$

act0_3: $cards\_set := \varnothing$

end

Event AddBirthday ⟨ordinary⟩ $\widehat{=}$

REQ3 - Birthday book can add a person's birthday record.

any

name

date

where

grd0_1: $name \in NAME$

grd0_2: $date \in DATE$

grd0_3: $name \notin known$

then

act0_1: $birthday := birthday \cup \{name \mapsto date\}$

act0_2: $known := known \cup \{name\}$

end

Event ChangeBirthday ⟨ordinary⟩ $\widehat{=}$

REQ 4 - The system can change a known person's birthday

any

name
new_date

where

grd0_1: $name \in NAME$
grd0_2: $name \in known$
grd0_3: $new\_date \in DATE$

then

act0_1: $birthday(name) := new\_date$

end

Event Remind ⟨ordinary⟩ $\widehat{=}$

REQ 5 - The system can query a set of people who have a given birthday to send birthday cards.

any

today

where

grd0_1: $today \in DATE$

then

act0_1: $cards\_set := dom(birthday \rhd \{today\})$

end

END

2.2.2 m1

MACHINE m1

REFINES m0

SEES c0

VARIABLES

cards_set

names

dates_func

count

INVARIANTS

inv1_1: $count \in \mathbb{N}$

inv1_2: $names \in 1 \mathinner{..} count \rightarrow NAME$

inv1_3: $dates\_func \in ran(names) \rightarrow DATE$

inv1_4: $ran(names) = dom(birthday)$

inv1_5: $ran(dates\_func) = ran(birthday)$

inv1_6: $\forall i \cdot (i \in 1 \mathinner{..} count) \Rightarrow (dates\_func(names(i)) = birthday(names(i)))$

inv1_7: $\forall i, j \cdot (i \in 1 \mathinner{..} count \land j \in 1 \mathinner{..} count \land i \neq j) \Rightarrow (names(i) \neq names(j))$

inv1_8: ⟨theorem⟩ $names \in 1 \mathinner{..} count \rightarrowtail NAME$

EVENTS

Initialisation

begin

act0_3: $cards\_set := \varnothing$

act1_1: $names := \varnothing$

act1_2: $dates\_func := \varnothing$

act1_3: $count := 0$

end

Event AddBirthday ⟨ordinary⟩ $\widehat{=}$

REQ3 - Birthday book can add a person's birthday record.

refines AddBirthday

any

name

date

where

grd0_1: $name \in NAME$

grd0_2: $date \in DATE$
grd1_1: $name \notin ran(names)$

then

act1_1: $names(count + 1) := name$
act1_2: $dates\_func(name) := date$
act1_3: $count := count + 1$

end

Event ChangeBirthday ⟨ordinary⟩ $\widehat{=}$

REQ 4 - The system can change a known person's birthday

refines ChangeBirthday

any

name
new_date

where

grd1_1: $name \in NAME$
grd1_2: $name \in ran(names)$
grd1_3: $new\_date \in DATE$

then

act1_1: $dates\_func(name) := new\_date$

end

Event Remind ⟨ordinary⟩ $\widehat{=}$

REQ 5 - The system can query a set of people who have a given birthday to send birthday cards.

refines Remind

any

today

where

grd0_1: $today \in DATE$

then

act0_1: $cards\_set := dom(dates\_func \rhd \{today\})$

end

END

2.2.3 m2

MACHINE m2

REFINES m1

SEES c0

VARIABLES

cards

names

count

dates

lock

cursor

CB_name

CB_new_date

RM_today

cards_count

INVARIANTS

inv2_1: $dates \in 1 \mathinner{..} count \rightarrow DATE$

inv2_3: $\forall i \cdot (i \in 1 \mathinner{..} count) \Rightarrow (dates\_func(names(i)) = dates(i))$

inv2_4: $lock \in \{0, 1, 2\}$

inv2_5: $cursor \in \mathbb{N}$

inv2_6: $CB\_name \in NAME$

inv2_7: $CB\_new\_date \in DATE$

inv2_8: $RM\_today \in DATE$

inv2_9:

$(lock = 0) \Rightarrow ($
$(cursor \in 0 \mathinner{..} count))$

0 means AddBirthday / ChangeBirthdayStart / RemindStart is possible

inv2_10:

$(lock = 1) \Rightarrow ($
$count \geq 0 \land$
$CB\_name \in ran(names) \land$
$CB\_name \notin names[1 \mathinner{..} cursor - 1] \land$
$CB\_name \in names[cursor \mathinner{..} count] \land$
$cursor \in 1 \mathinner{..} count)$

1 means ChangeBirthday is happening

inv2_11: $cards\_count \in 0 \mathinner{..} count$

inv2_12: $cards \in 1 \mathinner{..} cards\_count \to NAME$

inv2_13:

$(lock = 2) \Rightarrow ($
$count \geq 1 \land$
$cursor \in 1 \mathinner{..} (count + 1) \land$
$cards\_count < cursor$
)

2 means Remind is happening

VARIANT

$count - cursor$

EVENTS

Initialisation

begin

act1_1: $names := \varnothing$
act1_3: $count := 0$
act2_1: $dates := \varnothing$
act2_2: $lock := 0$
act2_3: $cursor := 0$
act2_4: $CB\_name :\in NAME$
act2_5: $CB\_new\_date :\in DATE$
act2_6: $RM\_today :\in DATE$
act2_7: $cards\_count := 0$
act2_8: $cards := \varnothing$

end

Event AddBirthday ⟨ordinary⟩ $\widehat{=}$

REQ3 - Birthday book can add a person's birthday record.

refines AddBirthday

any

name
date

where

grd0_1: $name \in NAME$
grd0_2: $date \in DATE$
grd1_1: $name \notin ran(names)$
grd2_1: $lock = 0$

then

act1_1: $names(count + 1) := name$
act1_2: $dates(count + 1) := date$
act1_3: $count := count + 1$

end

Event ChangeBirthdayStart ⟨ordinary⟩ $\widehat{=}$

any

name
new_date

where

grd1_1: $name \in NAME$
grd1_2: $name \in ran(names)$
grd1_3: $new\_date \in DATE$
grd2_1: $lock = 0$
grd2_2: $count \geq 1$

then

act2_1: $CB\_name := name$
act2_2: $CB\_new\_date := new\_date$
act2_3: $lock := 1$
act2_4: $cursor := 1$

end

Event ChangeBirthdayProgress ⟨convergent⟩ $\widehat{=}$

when

grd2_1: $lock = 1$
grd2_2: $CB\_name \neq names(cursor)$
grd2_3: $cursor < count$

then

act2_1: $cursor := cursor + 1$

end

Event ChangeBirthdayFinish ⟨ordinary⟩ $\widehat{=}$

REQ 4 - The system can change a known person's birthday

refines ChangeBirthday

when

grd2_1: $lock = 1$
grd2_2: $CB\_name = names(cursor)$

with

name: $name = CB\_name$
new_date: $new\_date = CB\_new\_date$

then

act2_1: $dates(cursor) := CB\_new\_date$
act2_2: $lock := 0$
act2_3: $cursor := 0$

end

Event RemindStart ⟨ordinary⟩ $\widehat{=}$

any

today

where

grd0_1: $today \in DATE$
grd2_1: $lock = 0$
grd2_2: $count \geq 1$

then

act2_1: $lock := 2$
act2_2: $cursor := 1$
act2_3: $RM\_today := today$
act2_4: $cards := \varnothing$
act2_5: $cards\_count := 0$

end

Event RemindProgressCollect ⟨ordinary⟩ $\widehat{=}$

when

grd2_1: $lock = 2$
grd2_2: $cursor \leq count$
grd2_3: $dates(cursor) = RM\_today$

then

act2_1: $cursor := cursor + 1$
act2_2: $cards(cards\_count + 1) := names(cursor)$
act2_3: $cards\_count := cards\_count + 1$

end

Event RemindProgressIgnore ⟨ordinary⟩ $\widehat{=}$

when

grd2_1: $lock = 2$
grd2_2: $cursor \leq count$
grd2_3: $dates(cursor) \neq RM\_today$

then

act2_1: $cursor := cursor + 1$

end

Event RemindFinish ⟨ordinary⟩ $\widehat{=}$

REQ 5 - The system can query a set of people who have a given birthday to sendbirthday cards.

refines Remind

when

grd2_1: $lock = 2$

grd2_2: $cursor > count$

with

today: $today = RM\_today$

then

act2_1: $lock := 0$

act2_2: $cursor := 0$

end

END

3 Final Code For Remind Event Using Merging Rules

3.1 Original Event-B Events from m2

RemindStart
any $today$ **where**
$today \in \text{DATE}$
$today = 0$
$count \geq 1$
then
$lock := 2$
$cursor := 1$
$RM\_today := today$
$cards := \varnothing$
$cards_count := 0$
end

RemindProgressCollect
when
$lock = 2$
$cursor \leq \text{count}$
$dates(cursor) = RM\_today$
then
$cursor := cursor + 1$
$cards(cards\_count + 1) := names(cursor)$
$cards\_count := cards\_count + 1$
end

RemindProgressIgnore
when
$lock = 2$
$cursor \leq \text{count}$
$dates(cursor) \neq \text{RM\_today}$
then
$cursor := cursor + 1$
end

RemindFinish
refines $Remind$
when
$lock = 2$
$cursor > count$
then
$lock := 0$
$cursor := 0$
end

3.2 If-Merge RemindProgressCollect and RemindProgressIgnore to RemindProgress

RemindProgress

———————-

when

lock = 2

cursor ≤ count

then

if (dates(cursor) = RM_today) **then**

cards(cards_count + 1) := names(cursor)

cards_count := cards_count + 1

end

cursor := cursor + 1

end

3.3 While-Merge RemindProgress and RemindFinish to RemindProgressFinish

RemindProgressFinish

---

when

lock = 2

then

while (cursor ≤ count) **then**

if (dates(cursor) = RM_today) **then**

cards(cards_count + 1) := names(cursor)

cards_count := cards_count + 1

end

cursor := cursor + 1

end

lock := 0

cursor := 0

end

3.4 Init-Merge RemindStart and RemindProgressFinish to RemindRoutine

RemindRoutine (today: DATE)
Precondition: lock = 0 and names.count ≥ 1

---

lock := 2
cursor := 1
RM_today := today
cards.make_empty
cards_count := 0
while (cursor ≤ count) **then**
    if (dates(cursor) = RM_today) **then**
        cards(cards_count + 1) := names(cursor)
        cards_count := cards_count + 1
    end
    cursor := cursor + 1
end
lock := 0
cursor := 0