Module: 6SENG001W Reasoning about Programs
Module Leader: K. Draeger/P Howells
(email: K.Draeger/P.Howells@westminster.ac.uk)
Tutorial Exercises: 5
Subject: Evaluate Relation expressions using **Atelier B** & **ProB**
Date: 22/10/18

1 Introduction

Using the B tools load & type check the following B relation definitions B machine called `Relations` into **Atelier B** & then animate/evaluate the expressions given below using **ProB**'s "*Eval terminal*" or add them to the `Relations` Machine directly as `ASSERTIONS` & use **ProB**'s "*Eval Assertions terminal*" to check if they are true or false.

2 Evaluate the following expressions ✓

Load the `Relations.mch` machine into **Atelier B** & then **ProB** & evaluate the following expressions.

2.1 Value Expressions

1. $AAxXX$
2. $favourite$
3. $speaks$
4. $alphabet$
5. $\mathrm{card}(favourite)$
6. $\mathrm{card}(AAxXX)$
7. $\mathrm{card}(\{\ (1,2),(3,4)\ \})$

8. $\text{card}(speaks)$
9. $\text{card}(alphabet)$

2.2 Predicate Expressions ✓

1. $(Paul, purple) \in favourite$
2. $(Paul, pink) \in favourite$
3. $(Paul, blue) \notin favourite$
4. $(Wales, French) \in speaks$
5. $(Canda, Welsh) \notin speaks$

2.3 Cartesian Products

1. $XX \times AA$
2. $AA \times AA$
3. $XX \times XX$
4. $AA \times COLOUR$
5. $\text{prj}_1(COUNTRY, LANGUAGE)(Wales, Welsh)$
6. $\text{prj}_2(COUNTRY, LANGUAGE)(Wales, Welsh)$

2.4 Relational Domain & Range

1. $\text{dom}(favourite)$
2. $\text{ran}(favourite)$
3. $\text{dom}(speaks)$
4. $\text{ran}(speaks)$
5. $\text{dom}(AAxXX)$
6. $\text{ran}(AAxXX)$

7. $\mathrm{dom}(R1)$
8. $\mathrm{ran}(R1)$
9. $\mathrm{dom}(R2)$
10. $\mathrm{ran}(R2)$
11. $\mathrm{dom}(R3)$
12. $\mathrm{ran}(R3)$
13. $\mathrm{dom}(RR)$
14. $\mathrm{ran}(RR)$
15. $\mathrm{dom}(QQ)$
16. $\mathrm{ran}(QQ)$
17. $\mathrm{dom}(RR) \cap \mathrm{dom}(QQ)$

2.5 Relational Image Operator

1. $favourite\ [\ \{\ Paul, Sue\ \}\ ]$
2. $speaks\ [\ \{\ Canada, Wales\ \}\ ]$
3. $R1\ [\ \{\ aa, bb, cc\ \}\ ]$
4. $R2\ [\ \{\ cc, dd\ \}\ ]$
5. $alphabet\ [\ \{\ aa, bb, cc\ \}\ ]$
6. $QQ\ [\ \mathrm{dom}(RR)\ ]$

2.6 Relational Restriction Operators

1. $\{Jim, Ian\} \lhd favourite$
2. $favourite \rhd \{blue, red\}$
3. $\{\ Wales,\ Scotland,\ England,\ NIreland\ \} \lhd speaks$
4. $speaks \rhd \{French\}$

5. $\{\ Wales,\ Scotland,\ England,\ NIreland\ \} \lhd speaks$
6. $speaks \rhd \{\ English\ \}$
7. $\{\ 2, 3\ \} \lhd RR$
8. $RR \rhd \{3, 5\}$
9. $\{\ 2, 3\ \} \lhd RR$
10. $RR \rhd \{\ 3, 5\ \}$

2.7 Relational Composition

Using the $alphabet$ relation calculate the following compositions:

1. $alphabet\ ;\ alphabet$
2. $alphabet\ ;\ alphabet$
3. $alphabet\ ;\ alphabet\ ;\ alphabet$
4. $alphabet^5$
5. $alphabet^9$
6. $alphabet^{10}$
7. $RR\ ;\ QQ$
8. $QQ\ ;\ RR$
9. $RR\ ;\ RR$
10. $QQ\ ;\ QQ$

2.8 Relational Overriding

Try evaluating the following "override" expressions using both the $\lhd\!\!+$ operator & using its definitions:

$$RR \lhd\!\!+ QQ\ =\ (\ \mathrm{dom}(QQ) \lhd RR\ )\ \cup\ QQ$$

2.8.1 Overriding Expression

1. $R1 \lhd\!\!+ \{\ aa \mapsto 10\ \}$
2. $R1 \lhd\!\!+ \{\ bb \mapsto 9\ \}$
3. $R2 \lhd\!\!+ \{\ dd \mapsto 2,\ dd \mapsto 10\ \}$
4. $R2 \lhd\!\!+ \{\ aa \mapsto 9,\ bb \mapsto 10\ \}$
5. $R3 \lhd\!\!+ \{\ gg \mapsto 9,\ hh \mapsto 6,\ hh \mapsto 10,\ zz \mapsto 99\ \}$

2.8.2 Using Overriding

Using the two relations R & Q, work out the new relation given by R overriding Q ($Q \lhd\!\!+ R$) then compare this with $R \lhd\!\!+ Q$ given in the lecture notes.

$$RR \lhd\!\!+ QQ \quad = \quad \{\ (0,1),\ (1,2),\ (2,3),\ (3,3),\ (4,5),\ (4,6),\ (5,5),\ (6,7)\ \}$$

Finally, compare these two with the relation you get by just unioning the two relations: $RR \cup QQ$.

Show how the following people's choice of their *favourite colour(s)* can be modifying, i.e. the $favourite$ relation is modified using the overriding operator $\lhd\!\!+$.

1. $Paul$'s favourite colour is now $blue$.
2. Sue's favourite colours are now $pink$ & $purple$.
3. Ian's favourite colours are now $green$ & $yellow$.

3 Analyse the Hotel Rooms B Specification

Download the `HotelRooms.mch` B specification used in Relations Lecture.
Type check it using **Atelier B**.
Animate it in **ProB** & execute a sample of the operations so that several rooms have guests.
Then use **ProB**'s *Eval terminal* to evaluate the relation expressions used in the specification. For example:

1. In INITIALISATION, first:
   ROOM
   then
   ROOM * { empty }
2. In operation guestsCheckIn:
   { rm2 } * { Ian, Sue, Tom }
   First:
   guests
   then
   guests <+{ rm2 } * { Ian, Sue, Tom }
3. In operation guestsCheckOut:
   guests <+ { rm2 |-> empty }
4. In operation roomOccupants:
   ran({ rm1 } <| guests)
   guests[{ rm1 }]
   guests[{ rm2 }]
   guests[{ rm3 }]
   guests[{ rm4 }]
   guests[{ rm5 }]
5. In operation hasGuestCheckedIn:
   dom(guests)
   ran(guests)
6. In operation guestsSwapRoom pick two rooms that have guests in them.
   Assuming that rm1 & rm3 have guests then:
   guests[{ rm1 }]
   guests[{ rm3 }]
   { rm1 } * guests[{ rm3 }]
   { rm3 } * guests[{ rm1 }]
   { rm1 } * guests[{ rm3 }] \/ { rm3 } * guests[{ rm1 }]
   guests
   guests <+ ({ rm1 } * guests[{ rm3 }] \/ { rm3 } * guests[{ rm1 }