Module:	6SENG005W Formal Methods
Module Leader:	K. Draeger/P Howells
	(email: `K.Draeger/P.Howells@westminster.ac.uk`)
Tutorial Exercises:	3
Subject:	Using **ProB** to Evaluate Set Expressions
Date:	2022/23

1. Using ProB to Evaluate Set Expressions

Assuming that:

1. The B machine `Sets.mch` has been syntax & type checked using **Atelier B**, & there are no errors.
2. The `Sets.mch` machine has been loaded into the **ProB** animator & no errors have occurred.

After loading the `Sets.mch` machine into **ProB** you can evaluate expressions in two ways.

1. Evaluate expressions by using the *Eval* terminal.

   First load & begin the animation of `Sets.mch`.

   You start the *Eval* terminal from **ProB**, using the mouse to *right-click* over the bottom left window.

   You can then type the expressions in using the AMN ASCII notation.

   A new window will start & then you type expressions at the `Eval` prompt "`>>>>`" for example using card & generalised union $\bigcup$:

   ```
   >>>> card( AA )
     8
   >>>> card( AA ) < 10
     TRUE
   >>>> union( { {1, 2}, { 3, 4 } } )
     {1,2,3,4}
   ```

   For help with **ProB**'s expression syntax, see the help under the "*Help > Summary of B Syntax*" menu.

2. Evaluate expressions by using the ASSERTIONS machine clause.

   You add this clause into the Sets.mch definition using the Atlier B editor.

   The ASSERTIONS clause is added after the INITIALISATION clause.

   You can try *true* & *false* *"assertions"*, for example:

   Assertions

   ```
   ASSERTIONS
       EE \/ FF <: { aa, bb, cc, dd, ee, ff, gg } ;
       EE /\ GG = {} ;
       FF /\ GG = { aa }
   ```

   the first two are true & the last is false.

2. Evaluate Expressions

After loading the Sets.mch machine into **ProB**, evaluate the following set, constant & type expressions given below.

2.1 Value Expressions

1. $homeland$
2. $Benelux$
3. AA
4. BB
5. CC
6. DD
7. $Even$
8. Odd
9. $Fives$
10. $card(Benelux)$
11. $card(\{\})$

12. $card(\{\ 1, 2, 3, 4, 5\ \})$

13. $card(AA)$

14. $card(BB)$

15. $card(Even)$

16. $card(Odd)$

17. $AA \cup BB$

18. $CC \cup \{\ aa\ \}$

19. $DD \cup \{\ pp, aa, uu, ll\ \}$

20. $card(AA \cup BB)$

21. $Even \cup Odd$

22. $AA \cap BB$

23. $card(AA \cap BB)$

24. $AA \cap CC$

25. $BB \cap DD$

26. $Even \cap Odd$

27. $AA \setminus BB$

28. $DD \setminus BB$

29. $CC \setminus \{\ xx\ \}$

30. $Benelux \setminus \{\ NL\ \}$

31. $Even \setminus \{\ \}$

32. $Odd \setminus Odd$

33. $\bigcup\{AA, BB, CC, DD\}$

34. $\bigcup\{Even, Odd\}$

35. $\bigcup\{\ \{2, 3, 4, 5\}, \{2, 4\}, \{\ \}\}\ \}$

36. $\bigcap\{Even, Odd\}$

37. $\bigcap\{Even, \{2,3,4,5,6,7\}\}$

38. $\bigcap\{\ \{2,3,4,5\}, \{2,4\}, \{\ \}\}\ \}$

2.2 Predicate Expressions

Evaluate these expressions using both approaches:

First use the `Eval` terminal.

Then use the `ASSERTIONS` clause method.

1. $aa \in AA$
2. $zz \in AA$
3. $tt \notin BB$
4. $yy \notin CC$
5. $xx \in (AA \cup CC)$
6. $zz \in (AA \cup BB)$
7. $tt \notin (CC \cup DD)$
8. $ee \notin (BB \cup CC)$
9. $BB \subset AA$
10. $CC \subset AA$
11. $AA \subset AA$
12. $AA \subseteq AA$
13. $BB \subseteq CC$
14. $\{\} \subseteq AA$
15. $\{\ aa, bb, cc\} \subseteq AA$
16. $\{\ xx, yy\ \} \subset CC$
17. $card(AA) \leq 10$
18. $card(BB) \leq 3$

19. $card(CC) = 3$

20. $card(DD) = 6$

21. $5 \neq (6 + 3)$

22. $5 = (2 + 3)$

23. $7 < (3 * 4)$

24. $3 < 6$

25. $3 > 6$

26. $(3 < 6) \wedge (2 \leq 10)$

27. $(3 = 6) \wedge (2 \leq 10)$

28. $(3 < 6) \wedge (2 > 10)$

29. $(3 = 6) \vee (2 \leq 10)$

30. $(3 = 6) \vee (2 = 10)$

31. $(aa \in AA) \wedge (\{ aa, bb, cc \} \subseteq AA)$

32. $(gg \in AA) \vee (gg \in AA)$

33. $tt \notin BB$

34. $yy \notin CC$