
6SENG001W Reasoning about Programs

Tutorial 2. Using Atelier B & ProB to Evaluate Set Expressions

Introduction

This tutorial is based on the *sets*, *types* & *constants* introduced in Lecture 2.

It is intended to introduce you to using AMN sets notation & evaluating sets in an Abstract Machine using the B tools.

ProB is the main tool we shall use to evaluate set, constant & type expressions.

For more information about how to set up & use the B tools see the manuals & Tutorial 1.

Exercise 2.1

- Using Atelier B, creating a new B project.
- Do this from the Atelier B "Atelier B > New > Project" menu.

Enter a *Project Name*, e.g.

tutorial2

Select "Project Type" as "*Software Development*".

- Add a component to your **tutorial2** project. Do this from the "Atelier B > New > Component" menu.

Enter a new component name:

Sets.mch

add it to the project.

You will then use it to explore set & constant expressions.

- If everything has gone properly you should see an **orange box** with "Sets" in it.
-

Exercise 2.2

Using the Atelier B built in editor, type in the following **Sets** machine, that provides the definitions of various set & constant definitions:

MACHINE Sets

SETS

EU = { BEL, NL, LUX, FR, DK, POR, GBR, ITA, IRL, DUT, ESP, GRE } ;

LETTER = { aa, bb, cc, dd, ee, ff, gg, hh, ii, jj, kk, ll, mm,
nn, oo, pp, qq, rr, ss, tt, uu, vv, ww, xx, yy, zz }

CONSTANTS

Benelux,

AA, BB, CC, DD,
Even, Odd, Fives

PROPERTIES

```

Benelux <: EU & Benelux = { BEL, LUX, NL }
&
AA <: LETTER & AA = { aa, bb, cc, dd, ee, ff , gg, hh }      &
BB <: LETTER & BB = { aa, ee, ii, oo, uu }                    &
CC <: LETTER & CC = { xx, yy, zz }                             &
DD <: LETTER & DD = { ff, oo, rr, mm, aa, ll, ee, tt, hh, dd, ss }
&
Even <: NAT & Even = { ev | ev : NAT & ev mod 2 = 0 & ev <= 20 }
&
Odd <: NAT & Odd = { od | od : NAT & od mod 2 = 1 & od <= 20 }
&
Fives <: NAT & Fives = { fi | fi : NAT & fi mod 5 = 0 & fi <= 20 }

```

VARIABLES

homeland, EE, FF, GG

INVARIANT

```

homeland : EU
&
EE <: LETTER & FF <: LETTER & GG <: LETTER

```

INITIALISATION

```

homeland, EE, FF, GG := GBR, { ee }, { ff }, { gg }

```

END

Exercise 2.3

Syntax & Type Checking the specification , using **Atelier B**.

You can either syntax & type check the **Sets** specification as you type it in or after you have finished typing it in.

The **Atelier B** tool will type check it automatically immediately after you have saved any changes.

Error messages will be displayed in the "Outline" sub-window & underlined in red in the specification.

Alternatively, you can "force" type checking by either:

- pressing the *blue circular "Tc" button* at the top of the tool's main screen.
- pressing Control-T, i.e. hold down the "Ctrl" (Control) key & at the same time press the "T" key.

Exercise 2.4

Once the **Sets** machine has been syntax & type checked & there are **no errors**, you can *animate* it using the **ProB** animator.

To do this start the **ProB** animator from the Ubuntu "*Applications > Programming*" menu.

Then *open* the **Sets.mch** file from the "*File > Open*" menu, by using the "*Browse*" option.

If there are **no errors** then you should see:

1. The specification in the top window.
2. In the bottom "*Enabled operations*" window, you should see - "**INITIALISATION({})**".

To begin the animation "*double-click*" on **INITIALISATION({})**.

Note that there are **NO enabled operations listed** as this machine has none, you are simply going to use the definitions to evaluate set expressions.

Exercise 2.5

After loading the **Sets.mch** machine into **ProB** you can now use it to evaluate set expressions.

For details on how to do this & the list of expressions that you are to evaluate see these notes: [Evaluating Set expressions](#).

Last updated: 1/10/17
