

---

## 6SENG001W Reasoning about Programs

### Tutorial 4: Logic Exercises

---

#### Introduction

These tutorial exercises refer to the notes for **Lecture 4: Logic**.

In this tutorial you are required to use the two B tools *Atelier B* & *ProB* to evaluate logical formula, create & extend B machines.

---

#### Exercise 4.1

1. Read the **Lecture 4: Logic** notes.
  2. Read & familiarise yourself with the **AMN versions of the Logical symbols**. See ProB's "Help" menu & the lecture notes.
- 

#### Exercise 4.2

Copy & complete the **truth table** on slide 16 of the **Lecture 4: Logic** notes, for the propositional logic formula in AMN:

$$( (P \ \& \ Q) \ \text{or} \ \text{not}(R) ) \Rightarrow P$$

---

#### Exercise 4.3

Copy & complete the **truth table** on slide 22 of the **Lecture 4: Logic** notes, for the propositional logic formula in AMN:

$$(P \ \& \ Q) \Rightarrow R$$

---

#### Exercise 4.4

See **Lecture 4: Logic** notes **slide 32**.

Create a B machine called **Logic** & add the following to it:

- definition of the sets **AA** & **BB**,  
(Hint: define **AA** & **BB** as a **CONSTANT** & set its value under the **PROPERTIES** clause.)
- the universally quantified formula (1) & the existentially quantified formula (2) as **ASSERTIONS**

Once these are type correct animate it in ProB & use the **ASSERTIONS** terminal to check what their truth value is.

---

### Exercise 4.5

See **Lecture 4: Logic** notes **slide 37**.

Add the first two quantified formula on the slide to your Logic machine.

Once they are type correct animate it in ProB & use the ASSERTIONS terminal to check what their truth value is.

---

### Exercise 4.6

See **Lecture 4: Logic** notes **slide 38**.

Type the two set comprehensions into ProB's "Eval" terminal to evaluate the sets correspond to their values given on the slide.

Try each set comprehension using different values:

- In the "*squares*" one replace the "5" by other numbers, e.g. 10, 15, 20, ...
- In the "*primes*" one replace the "50" by other numbers, e.g. 100, 150, 200, ...

ProB can only fully list sets up to a maximum size, try different values (especially in the "primes" case) to try to find the maximum set size it can deal with.

---

### Exercise 4.7

Give a **plain English** explanation, i.e. write a sentence, of the four predicates given on **slide 42** of the **Lecture 4: Logic** notes.

---

### Exercise 4.8

See **Lecture 4: Logic** notes **slide 44**.

Add the following to your Logic machine:

- the four DEFINITIONS: even, odd, smallPrime & ascending,
- the query operation areAscending.

Once these are type correct, animate it in ProB & execute the areAscending operation for different values of xx, yy & zz.

Define a second alternative version of the areAscending operation that returns the truth value of the ascending(xx, yy, zz ) definition using the bool function:

```
truthValue <-- areAscending( xx, yy, zz )
```

In addition, define operation versions of even(nn) and odd(nn) called isEven(nn) and isOdd(nn) in a similar style to the new version of the areAscending operation.

---

### Exercise 4.9

See **Lecture 4: Logic** notes **slides 45 - 50**.

Copy the [logic lecture machine](#) example in to Atelier B, type check it & then animate & analyse it in ProB.

Verify that the ASSERTIONS listed on slide 48, do **FAIL**.

---

*Last updated: 8/11/20*

---