University of Westminster
School of Computer Science & Engineering

6SENG005W Formal Methods Coursework (2022/23)

Module Leader	Klaus Draeger (K.Draeger@westminster.ac.uk)
Unit	Coursework
Weighting:	50%
Qualifying mark	30%
Description	Develop a B specification of a *Robot moving around a* simple *Maze*, using the B tools Atelier B & ProB.
Learning Outcomes Covered in this Assignment:	The coursework assesses learning outcomes: LO1, LO2, LO3 & LO4.
Handed Out:	Week 07
Due Date	**13:00, Monday, 9th January 2023**
Expected deliverables	Electronic files: (a) B Specification Structure Diagram (.pdf) (b) Maze & Robot B Specification: B machine(s) (.mch) (c) Graph of ProB Animation Session History (.dot) (d) Screen shots demonstrating use of B tools to develop, animate and analyse the Maze & Robot B specification. (.jpg/.jpeg) (See section 3 for full details.) All files should be compressed into a single ZIP archive. The ZIP archive should be named using your surname & "CW", e.g. "howells_CW.zip"
Method of Submission:	**Online via Blackboard**
Type of Feedback and Due Date:	Verbal feedback in tutorial(s) before the assessment is submitted. Written feedback and marks 15 working days (3 weeks) after the submission deadline. **All marks will remain provisional until formally agreed by an Assessment Board.**

Assessment regulations

Refer to section 4 of the “How you study” guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:

http://www.westminster.ac.uk/study/current-students/resources/academic-regulations

Coursework Description

1. Introduction

This coursework requires you to develop a B specification of a *Robot moving around a* simple *Maze*, using the B tools Atelier B & ProB.

Figure 1. gives the layout of the rectangular shaped maze, the Robot is represented by "•" & its starting position is the entry square (1, 1).

The aim is to move the robot from the *entry square* through the maze using the various movement operations to get to the *exit square* of the maze (1, 5).

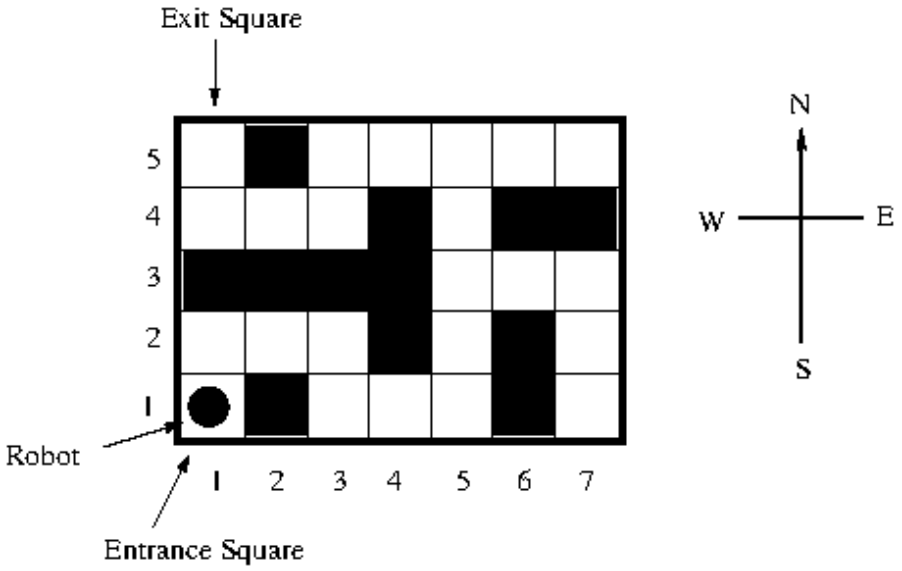

Figure 1. The Maze & Robot

Notes

- The squares of the maze form a grid of squares 7 wide by 5 high.
- The robot occupies only one square at a time & can only be in an "empty maze square", i.e. a square inside the maze and not one of the maze's internal walls.

  For example, the robot can be in square (5, 3), one of the maze's "path" squares, but not (4, 2) an "internal wall" square.
- The robot starts off in the *entrance square*, i.e. the bottom left square (1, 1).
- The robot can be moved around the maze by using one of four directions: *North* ↑, *South* ↓, *East* → and *West* ←; or it can "teleport" to one of the maze's "path" squares.
- Once the robot has found the exit square it can not make any further moves.

2. Develop a B Specification of a Robot moving around a Maze

Your B specification, i.e. collection of B machines, should include the following elements.

(a) Sets and Constants

Any sets and constants that are required to define the data and state of the maze and robot, and their properties.
(Hints: Represent the maze and the maze's internal walls as relations.)

(b) System State

The state variables required to represent the data for the robot and the maze. Including the state invariant and initialisation.

(c) Robot Movement Operations in the Maze

The following operations are the basic movements that all move the robot one square in the appropriate direction in the maze.

- *MoveNorth*
- *MoveSouth*
- *MoveEast*
- *MoveWest*

Note that if any attempted movement can not be performed, either because of the boundary wall of the maze, i.e. attempting to move North when in square (4, 5), or an internal maze wall, i.e. attempting to move East when in square (3, 2), then an error message is output indicating the reason it could not make the move.

The Robot can also "teleport" to a "path" square within the maze.

- *Teleport*

The operation inputs a square that the Robot attempts to teleport to. If the input square is suitable for the Robot to teleport to it does so and reports success. However, if it is not a suitable square for it to teleport to, it outputs an error message indicating why.

Note that all operations must report the outcome of an attempted movement, that is, either it was successful, failed due to the maze's boundary, failed due to an internal wall, or for some other reason.

(d) Enquiry Operations about Robot in the Maze

You must also specify the following enquiry operations, that all output information about the state of the Robot:

- *getPosition* - outputs the current position of the robot in the maze.
- *foundExit* - outputs *yes* if the robot is currently in the *exit* square of the maze, *no* otherwise.
- *hasVisitedSquare* - inputs a square and reports yes if that square has been previously visited by the robot, no otherwise.
  Note that a square is designated as having being visited by the Robot if it has been on the square more than once. That is, a square "has been visited" only after the robot has been on it and then has moved off it.
  (Hint: you will need to keep a record of the squares visited.)
- *robotsRoute* - outputs the **sequence of squares** the Robot has visited, in the order visited, i.e. its *route* through the maze.

(e) Reset Operation

A *Reset* operation to reinitialise the system, i.e. moves the robot back to the entry square, deletes the robots route and visited squares.

General Requirements

The B specification should use the appropriate features to define the data & operations in any machines that you define. If you use a collection of machines then you must use the appropriate B structuring features to combine them.

The specification **must** be **syntactically & type correct**, as checked by using the **Atelier B** tool.

The specification **must** be **animated** by **ProB**. That is it must *initialise* correctly & all operations can be *animated* successfully & can be used to move the Robot from the entry square to the exit square using a **combination of all of the movement operations**.

3. Blackboard Submission

The following 4 components are to be submitted via **Blackboard**:

(1) The Structure Diagram of the Maze & Robot System, in terms of the abstract machine(s) used to specify it. You must also include as a note with the diagram a "Plain English" explanation of the "state invariants" of the system. Examples of Structure Diagrams can be found in the lecture notes and in the tutorial exercises.
SUBMIT: 1 ".pdf" file. **[10 MARKS]**

(2) The B Specification of the Maze & Robot System, i.e. the abstract machine(s).
SUBMIT: all B machine plain ASCII Text files: ".mch". **[65 MARKS]**

(3) A Graph representation of a complete ProB Animation Session history. Using ProB perform an animation session that shows how the Robot is moved from the entry square to the exit square **using a combination of all of the movement operations**, e.g. *MoveNorth, MoveSouth, MoveEast, MoveWest* and *Teleport*, the route does not need to be the shortest. View this Animation Session as a "DOT" graph and then save it. Note you should just submit this ".dot" file as it is, **do not open** it in a word processor and then submit that file, as it may get corrupted by the word processor.
SUBMIT: 1 graph file: ".dot". **[10 MARKS]**

(4) Screen shot images (.png or .jpg or .jpeg) showing the use of the B tools to develop, animate and analyse the Maze & Robot B specification:

(a) Atelier B syntax & type checking: 1 image showing the Atelier B main **type check** window and **editor** window showing the specification. (File: AtelierB.png)

(b) ProB initialisation: 1 image showing the **initialised** specification in ProB. (File: ProB_initialisation.png)

(c) ProB animation: 1 image showing ProB at the **end of the animation** that corresponds to the journey from the entry square to the exit square, i.e. the final state represented in the DOT graph. (File: ProB_animation .png)

(d) ProB analysis: 1 (or more) images showing ProB's "**Evaluation View**" terminal displaying the value of the INVARIANT, PROPERTIES and OPERATIONS. (File(s): ProB_EV_1 .png, ProB_EV_2 .png, …)

(e) ProB analysis: 1 image showing the use of ProB's "**Eval**" terminal to showing the value of the variables at the end of the journey animation. (File: ProB_Eval .png)

SUBMIT: 5 – 7 screen shot image files: ". png" or ".jpg" or ".jpeg" **[15 MARKS]**

NOTE: All files should be compressed into a single ZIP archive. The ZIP archive should be named using your surname & "CW", e.g. "howells_CW.zip".