

## Coursework 7 (Scala, Knight's Tour)

This coursework is worth 10%. The first part is due on 23 November at 11pm; the second, more advanced part, is due on 30 November at 11pm. You are asked to implement Scala programs that solve various versions of the *Knight's Tour Problem* on a chessboard.

### Disclaimer

It should be understood that the work you submit represents your own effort. You have not copied from anyone else. An exception is the Scala code I showed during the lectures or uploaded to KEATS, which you can freely use.

### Background

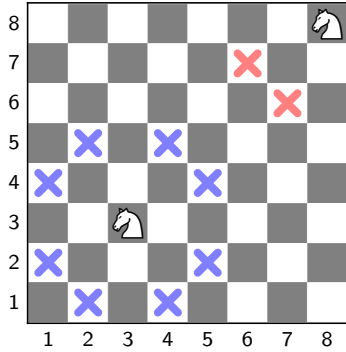
The *Knight's Tour Problem* is about finding a tour such that the knight visits every field on a  $n \times n$  chessboard once. For example on a  $5 \times 5$  chessboard, a knight's tour is as follows:

5	24	11	6	17	0
4	19	16	23	12	7
3	10	5	18	1	22
2	15	20	3	8	13
1	4	9	14	21	2
	1	2	3	4	5

The tour starts in the right-upper corner, then moves to field (4,3), then (5,1) and so on. There are no knight's tours on  $2 \times 2$ ,  $3 \times 3$  and  $4 \times 4$  chessboards, but for every bigger board there is.

A knight's tour is called *closed*, if the last step in the tour is within a knight's move to the beginning of the tour. So the above knight's tour is not closed (it is open) because the last step on field (1,5) is not within the reach of the first step on (5,5). It turns out there is no closed knight's tour on a  $5 \times 5$  board. But there are on a  $6 \times 6$  board.

If you cannot remember how a knight moved in chess, below are all potential moves indicated for two knights, one on field (3,3) and another on (8,8):



### Disclaimer

It should be understood that the work you submit represents your own effort. You have not copied from anyone else. An exception is the Scala code I showed during the lectures or uploaded to KEATS, which you can freely use.

### Task

The task is to implement a regular expression matcher based on derivatives of regular expressions. The implementation should be able to deal with the usual (basic) regular expressions

**Important!** Your implementation should have explicit cases for the basic regular expressions, but also explicit cases for the extended regular expressions. That means do not treat the extended regular expressions by just translating them into the basic ones. See also Question 2, where you are asked to explicitly give the rules for *nullable* and *der* for the extended regular expressions.