

CIS 721 - Real-Time Systems
Homework #4
Fall 2015

Due: Friday, Nov. 13, 2015 at 11:59 pm (25 points)

1. Vikings' Problem

Adjust the solution to the Vikings' Problem given in bridge.xml to handle eight vikings with crossing times of 1, 2, 3, 4, 5, 6, 7, and 8, respectively. Determine the minimum time required for all eight vikings to cross the bridge. Repeat the problem for the first 7 vikings with crossing times of 1 through 7, then the first 6, 5, 4, and 3. Hint: You only need to change the verification query for each of these additional steps. Develop a general formula for any number of vikings, say n , for any n greater than or equal to 3.

2. Mini Lights Out Problem

Model and analyze the following Mini Lights Out problem in UPPAAL. The Lights Out puzzle consists of a grid of cells, which have lights in them. By toggling a switch inside a cell, the lights in the cell, and in the non-diagonally adjacent cells (to the left or right or up or down), change to the opposite of what they were. The puzzle starts with some of the lights on. The goal is to press the correct cells to turn all lights out.



See: <http://www.jaapsch.net/puzzles/lights.htm#java>

The Mini Lights Out version looks like a normal Lights Out played on a 4×4 square. The main difference is that the board has no edges - the left and right columns are considered to be adjacent, as are the top and bottom rows. Every light therefore has exactly 4 neighbours, and so every move changes exactly 5 lights. For example, pressing the top-left button changes its own light, the light below, the light to the right, the light to the 'left' (the top right corner), and the light 'above' (the bottom left corner).

Consider the example shown below starting with four cells lit as shown in the upper left. A total of four clicks are required to turn all cells off:



Construct an UPPAAL model and corresponding query to determine the minimum number of clicks needed to solve the following puzzle:



What to Submit: Submit an electronic copy of your final system models (UPPAAL .xml files). Also, submit a list of the queries (.q file) used to determine the optimal solutions, and the formula derived for part 1. **Hints:** Uppaal supports typedefs and forall; e.g., `typedef int[0,3] position;` to declare a new type “position” representing an integer of 0,1,2, or 3. Also, you can use: `forall (i:position) (forall (j:position) board[i][j]==0)` in a guard to see if all board[i][j] positions are 0.