UC Davis STA 242 2015 Spring Assignment 2

Wenhao Wu, 9987583

April 29, 2015

1 Data Structure and Algorithm Design

A `BMLGrid` instance contains 3 components:

`grid` A `r`-by-`c` integer matrix. If `grid[i,j]==0`, then there is no car on the crossing of *i*-th row and *j*-th column; if `grid[i,j]==1`, there is a red car on that grid; if `grid[i,j]==2`, there is a blue car on that. In our program, when `grid` is indexed, it is treated as a vector (1-D).

`blue` An integer vector contains the 1-D indices of all blue cars in `grid`.

`red` An integer vector contains the 1-D indices of all red cars in `grid`.

We define 2 key functions that returns a vector of 1-D indices in `grid`

`idx_right()` Given an input vector of 1-D indices in `grid`, return a vector of 1-D indices in `grid` for grids to the *right* of the input grids.

`idx_up()` Given an input vector of 1-D indices in `grid`, return a vector of 1-D indices in `grid` for grids to the *up* of the input grids.

Upon each step, we use `idx_up()`(`idx_right()`) to check in `grid` whether the grids to the up(right) of the grids represented by `blue`(`red`) is occupied, then update the cars' indices `blue`(`red`) and the grid state `grid` accordingly.

2 Simulation Results

2.1 Behavior of the BML model

2.2 Code Performance

3 Build BMLGrid Package