

Modeling Discrete Event System Using DEVS

(2014 Fall)

Assignment 2 – Traffic flow Simulation in 1D Cellular Automaton Model

Student name: Zheng Xia

Student number: 7356005

University of Ottawa

Introduction

1. Problem Statement

Traffic problems have been widely concerned in recent years as the various automobiles are used in different occasions. In terms of the ways of studying the problem, the cellular automaton (also short for CA) have been implemented dramatically in simulation of traffic systems with the advantage of flexibility and simplicity. A prototype describing the traffic flow roughly in one-dimensional CA model is the rule-184 recalled in the Wolfram. In this rule, cars and roads have been regarded as different particles and discrete lattices respectively, as show in the Fig 1. All particles can go ahead at the same time in each time step as long as the next sites of theirs in the direction of advance are empty. This model, elaborately, illustrates a boost of phase transition between free movement phase and jammed phase.

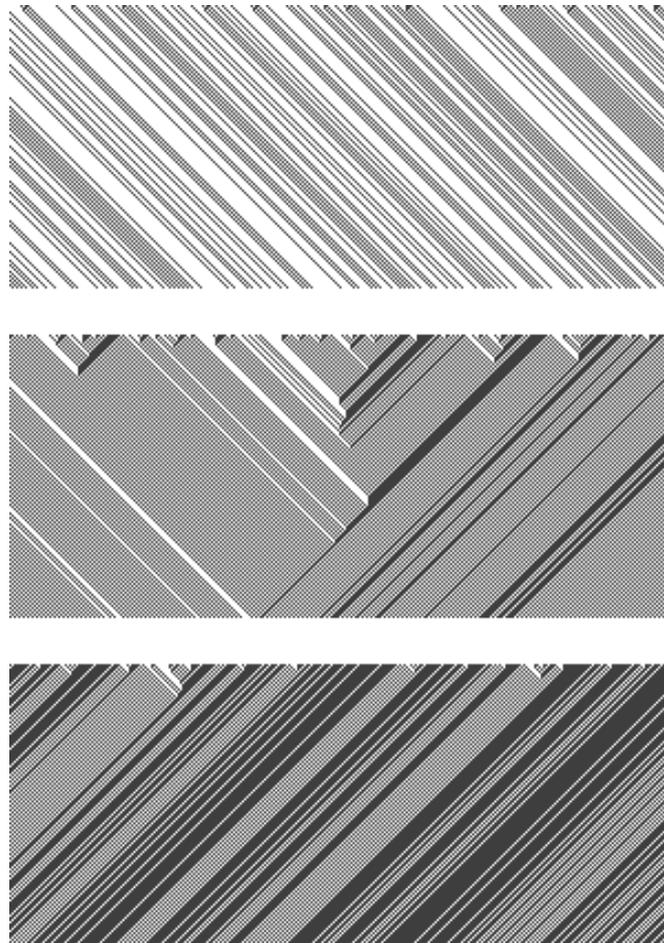


Fig 1. Rule-184 CA model simulation (after 128 steps in different conditions and density)

In this case, the rule-184 can be regarded as a reference for the further experiment in this report with disparity rule that is being used in simulation.

The experiment steps would be:

- Understanding the different meaning in distinguish CA rules
- To simulate the rule-184 and to get result
- Test other types of rules

Model specification

In this part, a formal specification of the traffic flow Cell-DEVS model will be presented first, followed by the rule of this model which has been illustrated in the research paper [1]. After the rules are being set, disparate values will also be applied as initial inputs. In addition, graphical outputs would also be shown and discussed in terms of the different outcomes in initial values and the rules. Last but not least, the conclusion will be drawn as the analysis of this cellular automaton model.

Formal specification of the Cell-DEVS models

The more concrete Cell-DEVS formalism with port specifications is as follows:

$$CD = \langle X, Y, I, S, q, N, d, d_{int}, d_{ext}, t, l, D \rangle$$

- X: Set of input External Events
- Y: Set of output External Events
- I: $I = \langle \eta, \mu, P^x, P^y \rangle$ is the model's modular interface, with n neighboring cells
 - 3 total neighbors in this case for each cell
 - Neighbors: traf(0,-1), traf(0,0), traf(0,1)

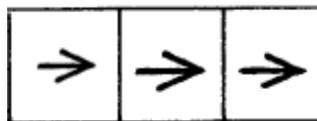


Fig 2. cell traf(0,-1)(left), traf(0,0)(center) and traf(0,1)(right) and the direction(arrow)

- S: possible state {0,1}
 - 0 means the cell is empty
 - 1 means the cell is occupied by a car
- The delay of the cell is 100 ms

Rules of reaction in Cell-DEVS model

Rules in the Cellular Automaton model define the general behavior of the cell through the time sequence. That is, it describes how the occupation situation of the neighbor of the cell affect the cell value after the delay time delay. There are two rules that describe best in the simple model of traffic flow and have been used in this experiment: rule-184 and rule-232.

- Rule-184

This rule has been widely used in the model of traffic flow simulation, especially the single lane of a highway. By applying this rule, vehicles (representing as particles) move in a single direction and the moving procedure is controlled by the car in front of them. The number of car in the simulation can remain unchanged [2]. Noticed that the rules have the same coding methodology as the Wolfram Cellular Automaton model. The rule describes that the model is deterministic. That is, there is no acceleration in the car moving process (which means the all the cars have only one speed, i.e. $V_{max} = 1$)

rule : 1 100 { (0,-1) = 1 and (0,0) = 1 and (0,1) = 1}
 rule : 0 100 { (0,-1) = 1 and (0,0) = 1 and (0,1) = 0}
 rule : 1 100 { (0,-1) = 1 and (0,0) = 0 and (0,1) = 1}
 rule : 1 100 { (0,-1) = 1 and (0,0) = 0 and (0,1) = 0}
 rule : 1 100 { (0,-1) = 0 and (0,0) = 1 and (0,1) = 1}
 rule : 0 100 { (0,-1) = 0 and (0,0) = 1 and (0,1) = 0}
 rule : 0 100 { (0,-1) = 0 and (0,0) = 0 and (0,1) = 1}
 rule : 0 100 { (0,-1) = 0 and (0,0) = 0 and (0,1) = 0}

- Rule-232

By changing two typical rules, the new rules would be shown as follows:

rule : 1 100 { (0,-1) = 1 and (0,0) = 1 and (0,1) = 1}
 rule : 1 100 { (0,-1) = 1 and (0,0) = 1 and (0,1) = 0}
 rule : 1 100 { (0,-1) = 1 and (0,0) = 0 and (0,1) = 1}
 rule : 0 100 { (0,-1) = 1 and (0,0) = 0 and (0,1) = 0}
 rule : 1 100 { (0,-1) = 0 and (0,0) = 1 and (0,1) = 1}
 rule : 0 100 { (0,-1) = 0 and (0,0) = 1 and (0,1) = 0}
 rule : 0 100 { (0,-1) = 0 and (0,0) = 0 and (0,1) = 1}
 rule : 0 100 { (0,-1) = 0 and (0,0) = 0 and (0,1) = 0}

The rules in red indicate the changes of the model comparing to the rule-184. Through the changing, a prospective view of the moving procedure can be described. This is that if the car has a neighbor behind and it finds the room in front is empty, both its neighbor and itself can move ahead. That is, in the case of 110 in traf(0,-1), traf(0,0), traf(0,1), the current cell with would still be occupied because of the car sequence in the next time step. In this way the original car will move out and the new car will come after the delay, which also could represent a result from a traffic model in the real world.

Graphical Output Result

In this session, graphical result will be shown with different rules implementation. In addition, different test cases will be used to compare in the next step.

- Rule-184
 - Initial value of row: 11111010110001101010

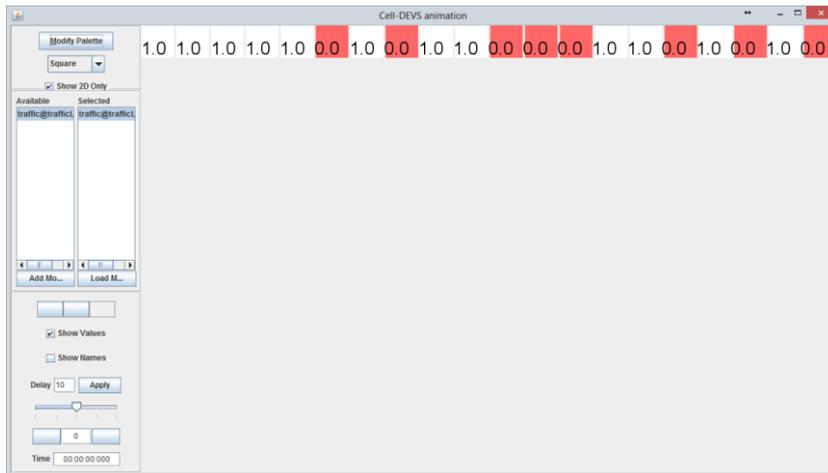


Fig 3. Initial state (time = 0ms)

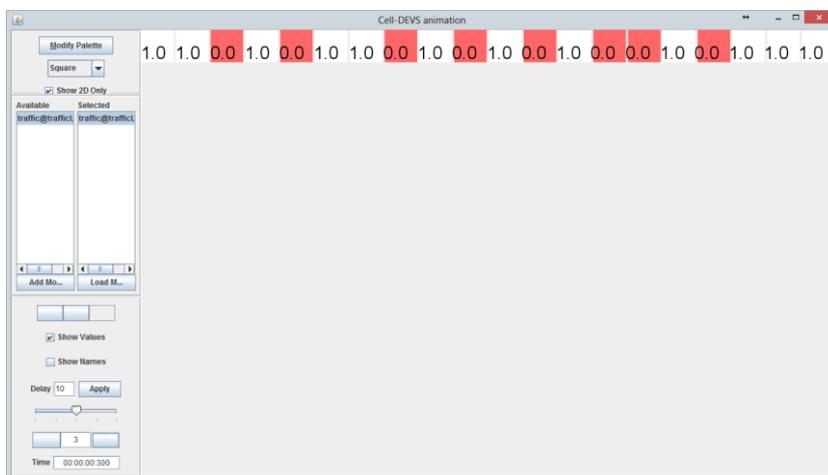


Fig 4. Three-frames state (time = 300ms)



Fig 5. Eleven-frames state (time = 1100ms)

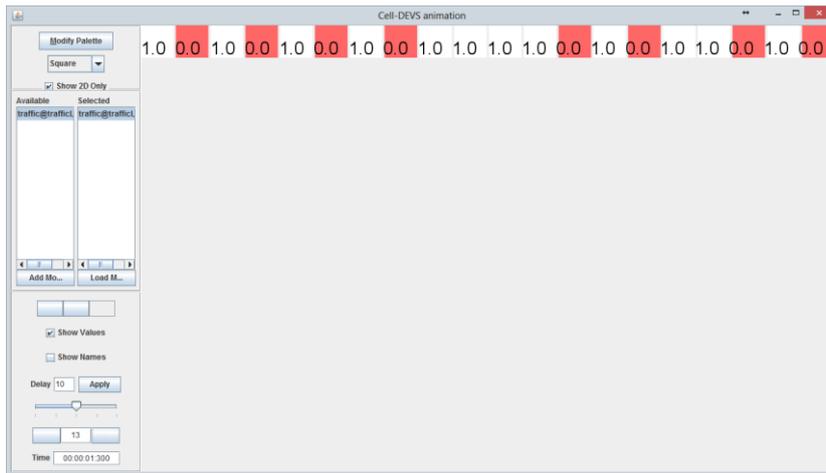


Fig 6. Thirteen-frames state (time = 1300ms)

- Initial value of row: 010101010101010101



Fig 7. With different initial value of the rows (time = 0ms)

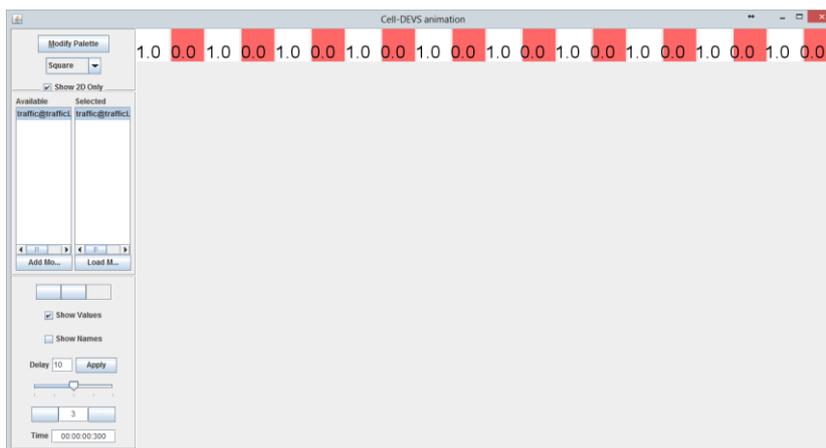


Fig 8. With different initial value of the rows (time = 300ms)

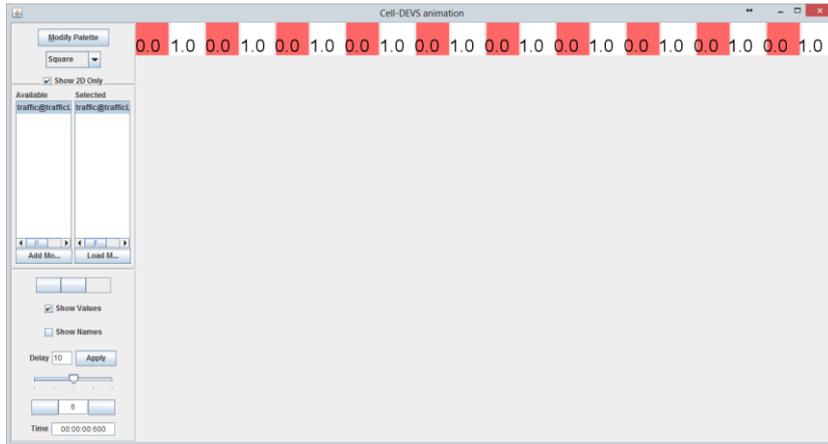


Fig 9. With different initial value of the rows (time = 600ms)

- Rule-232

- Initial value of row: 11111010110001101010

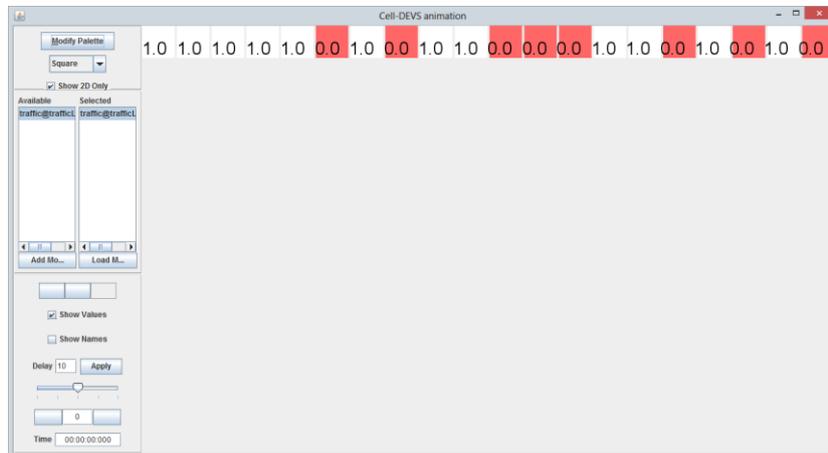


Fig 10. Result by applying the rule-232 (time = 0ms)

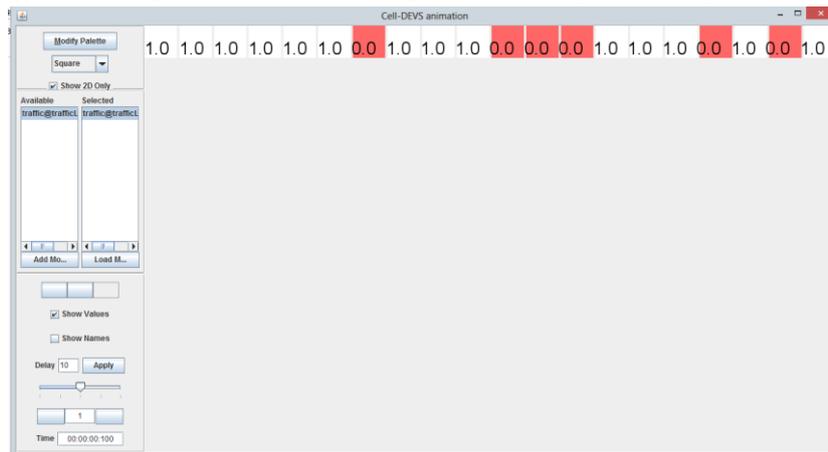


Fig 11. Result by applying the rule-232 (time = 100ms)

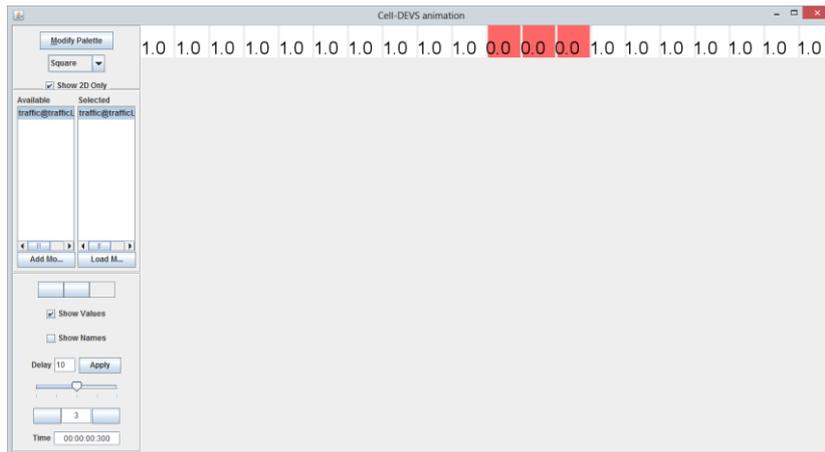


Fig 12. Result by applying the rule-232 (time = 300ms and it ends)

- Initial value of row: 010101010101010101



Fig 12. Result by applying the rule-232 with different initial value (time = 0ms)

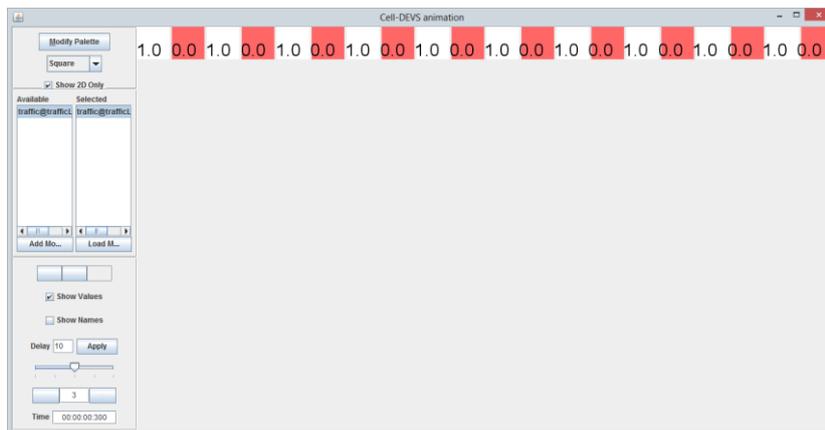


Fig 13. Result by applying the rule-232 with different initial value (time = 300ms)



Fig 14. Result by applying the rule-232 with different initial value (time = 600ms)

Conclusion

As we can see from the graphical result, the Cell-DEVS model using different rule gives the enormous difference in some outcomes when the same initial value has been set. This result indicates the rule-232 might not fit the traffic flow model because it may meet the problem when the car moving in this way. That is, the collision might happens when the cell output a car and a car comes into the cell at the same time, which will lead to “car-rewrite” or “cell-empty” situation depending on the order of moving.

Reference

- [1]. Fukui, Minoru; Ishibashi, Yoshihiro. Traffic Flow in 1D Cellular Automaton Model Including Cars Moving with High Speed[J]. Journal of the Physical Society of Japan, 1996, 65(6):1868
- [2]. http://en.wikipedia.org/wiki/Rule_184