

SIMULATION OF TRAFFIC FLOWS USING CELLULAR AUTOMATA

G. Y. Kazancev, G. A. Omarova

Institute of Computational Mathematics and Mathematical Geophysics SB RAS
630090, Novosibirsk, Russia

Simulation of transport systems is the demanded task in control of road networks helping to make decisions on further development and extension of transport system. In particular, simulation allows to define need for extension of a road network or adding of means of regulation. In this paper, traffic is simulated according to the cellular automaton of the Nagel-Scheckenberg model. It was the first model to take into account the imperfect behavior of human drivers and was thus the first model to explain the spontaneous formation of traffic jams. Now application of cellular automaton is also actual for simulation and research of different road situations and behavior of pedestrians. This model satisfies to traffic regulations in ideal conditions, when all machines move with the fixed speed and all drivers follow rules.

The Nagel-Scheckenberg model is a one-dimensional stochastic machine designed to simulate traffic. The model dimensional grid used in each cell is placed exactly one machine, the cell is either empty or contains a car. Time is discrete, the machine moves to forward an integral number of cells for each step of iteration. At each step of the iteration for each car in turn all the rules are applied. The first rule is responsible for aspiration of drivers to go as soon as possible, without violating the rule, the second rule doesn't allow collisions, and the third rule introduces an element of randomness in the motion of each driver. This rule set is the minimum set necessary for reproduction of basic properties of a transport flow. The main advantage of this model is its simplicity compared with other approaches; it is important in the transition to more complex structures, such as multi-row model or models including intersections. Both of these generalizations require a transition to a two-dimensional grid and seriously complicate the process of updating the machine state, as multiband movement also requires at each step to assess the need and possibility to change the band, and also requires a method of resolving the conflict when you attempt to move in one cell from two different neighboring bands. Creation of the intersection, results in need to process movement in the non-parallel directions through the same cells that can be realized through the cells containing only part of the machine. Besides creation of the intersection it is necessary to consider a priority of roads that can seriously complicate rules of transition or procedure of their application. For implementation of the intersection the following cellular automaton was defined.

Cellular automaton $CA = \langle A, C, \tau, X, T \rangle$, A — set of statuses of a cell, C — set of cells, τ — transition function that changes the state of the automaton, X — neighborhood relation, T — set of clock periods of time. Each object is characterized by two parameters — the direction of movement and the identifier. Cells belonging to the set C , are determined by two different sizes of the full and half. One object occupies one cell of the complete size or two cells of the half size. The neighborhood ratio X is not the same in different cells and strongly depends on the position of the cells relative to intersections. For determination of τ we introduce rules for all cells, after that we define application of τ on some configuration of the automatic machine as, application of rules of cells in a certain order.

Functions for operation with the cellular automaton: $Rules(c_i)$ — set of rules of a cell c_i , $Name(r_{ij})$ — rule name r_{ij} in $Rules(c_i)$. $Imp(r_{ij})$ and $Next(r_{ij})$ — set of important and following cells for the rule.

The basic rules have the general structure. The structure of additional rules for the second part of object is available only for cells of the half size. As for these cell it is necessary to distinguish objects from each other, for this purpose we enter the Id function returning an object identifier. The rule for an empty cell is trivial - the cell doesn't influence adjacent cells and doesn't change the statuses without influence of other cells. Each cell has rule set for movement through it in any the allowed direction, according to traffic regulations.

A cellular automaton for traffic simulation on a group of intersections was designed. The program of generation of the cellular automaton for different intersections was realized.

The models can be used to understand, predict and optimize different traffic situations and a2s examples for the various possible extensions and fields of applications.

Key words: model, cellular automata, distance, speed, acceleration, regular grid.

References

1. Kravchenko P.S., Omarova G.A. Mikroskopicheskie matematicheskie modeli transportnyh potokov. Analiticheskij obzor // Zhurnal „Problemy informatiki“. N 1. 2014. P. 71–78.
2. Gasnikov A. V. i dr. Vvedenie v matematicheskoe modelirovanie transportnyh potokov: MFTI Moskva. 2010.
3. Shvecov V. I. Matematicheskoe modelirovanie transportnyh potokov. Avtomatika i Telemekhanika. 2003. N 11. P. 3–46
4. Dzh. fon Nejman. Teorija samovosproizvodjashhihsja avtomatov. M.: „Mir“, 1971.
5. Cremer M., Ludwig J. A fast simulation model for traffic flow on the basis of Boolean operations // Math. Comp Simul. 1986. V. 28. P. 297–303.
6. Nagel K., Schreckenberg M. A cellular automation model for freeway traffic // Phys. I France. 1992. V. 2. P. 2221–2229
7. Simon, P. M., Gutowitz, H. A. A Cellular Automaton Model for Bi-Directional Traffic. Phys. Rev. E 57, 2441, 1998.
8. Nowak S., Schadschneider A. A Cellular Automaton Approach for Lane Formation in Pedestrian Counterflow // Traffic and Granular Flow '11. P. 149–160.
9. Torsten Held, Stefan Bittihn. Cellular automata for traffic simulation — Nagel-Schreckenberg model // Project report in Computational Physics. Bonn, 2011.
10. Hafstein S. F., Chrobok R., Pottmeier A., Wahle J., Schreckenberg M. A High Resolution Cellular Automata Traffic Simulation Model with Application in a Freeway Traffic Information System // Computer-Aided Civil and Infrastructure Engineering, 2004. V. 19. I. 5. P. 338–350.
11. Zamith M., C'elia R., Leal-Toledo P., Kischinhevsky M., Clua E., Brand'ao D., Montenegro A., B. Lima Edgar. A Probabilistic Cellular Automata Model for Highway Traffic Simulation // Procedia Computer Science 1. 2012. P. 337–345.
12. Omarova G. A., Kazancev G. Ju. Primenenie kletochnyh avtomatov dlja modelirovanija transportnyh potokov // Problemy informatiki. N 3. 2015. P. 15–21.

МОДЕЛИРОВАНИЕ ТРАНСПОРТНЫХ ПОТОКОВ С ПРИМЕНЕНИЕМ КЛЕТОЧНЫХ АВТОМАТОВ

Г. Ю. Казанцев, Г. А. Омарова

Институт вычислительной математики и математической геофизики СО РАН
630090, Новосибирск, Россия

УДК 519.179.2+512.23

Рассмотрена и реализована модель Нагеля-Шрекенберга на прямой. Разработана схема работы клеточных автоматов для различного вида перекрестков.

Ключевые слова: модель, клеточный автомат, расстояние, скорость, ускорение, регулярная решетка.

Введение. Моделирование транспортных систем — это востребованная задача в управлении дорожными сетями, помогающая принимать решения о дальнейшем развитии и расширении транспортной системы. В частности, моделирование позволяет определить потребность в расширении дорожной сети или добавлении средств регуляции. Грамотное планирование позволяет снизить нагрузку на транспортную сеть, избежать пробок и уменьшить среднее время в пути для водителей [1].

Для исследования и моделирования транспортных систем используют различные макро- и микромоделли для оценки объемов загрузки и интенсивности потоков [2]. В 50-е годы на волне развития газовой и гидродинамики появились первые макроскопические модели транспортной сети на основе уравнений гидродинамики, в которых транспортный поток уподоблялся сжимаемой жидкости (М. Лайтхилл и Дж. Уизем, П. Ричардс). Тогда же появилась и первая микроскопическая модель — модель следования за лидером, в которой явно выписываются уравнения движения для каждого автомобиля (А. Решель, Л. Пайпс и др.) [3]. В дальнейшем класс макро- и микроскопических моделей был значительно расширен. Одним из способов создания микроскопических моделей является использование клеточных автоматов (СА) [4–7]. В настоящее время применение СА также актуально для моделирования и исследования различных дорожных ситуаций и поведения пешеходов. Необходимость решения предложенной задачи на индивидуальных вариантах состоит в отличии дорог, рельефа местности, правил дорожного движения в России [8–10], либо описания поведения водителя различными вероятностными функциями [11].

Данная работа является продолжением работы авторов [12]. В предлагаемом исследовании рассматривается клеточно-автоматная модель и продолжены исследования предыдущих работ авторов, отличающаяся тем, что длины клеток различны и рассматривается не перпендикулярный перекресток.

Дальнейшее изложение организовано следующим образом. В первом параграфе рассматриваются модель Нагеля-Шрекенберга и ее алгоритм для прямого участка, во втором рассматривается построение клеточного автомата для нескольких видов перекрестка и в последнем параграфе — описание реализации алгоритма на языке программирования C#.

Алгоритм Нагеля-Шрекенберга для прямой. Модель Нагеля-Шрекенберга представляет собой стохастический одномерный автомат, предназначенный для моделирования дорожного движения [6]. Данная модель имитирует движение по одной полосе прямой дороги без возможности обгона и перестроения. В модели Нагеля-Шрекенберга используется одномерная сетка, в каждую ячейку помещается ровно одна машина, ячейка является либо пустой, либо содержит автомобиль. Время дискретно, за каждый шаг итерации машина перемещается на целое число ячеек вперед. Для простоты описания правил введем $n_i(t)$ и $v_i(t)$ — номер ячейки, в которой находится машина с номером i и ее скорость (число ячеек, которые будут пройдены за один такт времени) на шаге итерации t , машина с большим номером едет перед машиной с меньшим номером, v_{max} — максимальная допустимая скорость на участке.

Тогда правила, единые для всех клеток, имеют следующий вид:

1) *Ускорение.* Если скорость i -го автомобиля меньше v_{max} , то автомобиль ускоряется

$$v_i(t) = \min \{v_i(t-1) + 1, v_{max}\}.$$

2) *Торможение.* Если расстояние до едущего впереди автомобиля меньше скорости, то значение скорости приравнивается этому расстоянию

$$v_i(t) = \min \{v_i(t), n_{i+1}(t-1) - n_i(t-1)\}$$

3) *Случайные возмущения.* С заданной вероятностью p водитель замедляется.

$$v_i(t) = \begin{cases} \max \{v_i(t) - 1, 0\}, & p \\ v_i(t), & 1 - p. \end{cases}$$

4) *Движение.* Изменение положения автомобиля согласно новой скорости

$$n_i(t) = n_i(t-1) + v_i(t).$$

На каждом шаге итерации к каждому автомобилю поочередно применяются все правила. При этом первое отвечает за стремление водителей ехать как можно быстрее, не нарушая правила, второе не допускает столкновений, а третье вносит элемент случайности в движение каждого водителя. Данный набор правил является минимальным набором, необходимым для воспроизведения базовых свойств транспортного потока. Численные эксперименты показали, что при малых плотностях потока данная модель устойчива и теряет устойчивость при высоких плотностях [3]. Впрочем, если вероятность случайного замедления равна нулю, поток остается устойчивым при любой плотности, что является недостатком по сравнению с гидродинамическими моделями, в которых единичная флуктуация способна приводить к нестабильности потока. В более поздних моделях эта проблема была решена теми или иными способами. Как можно заметить, основным достоинством данной модели является ее простота по сравнению с другими подходами, что является важным при переходе к более сложным структурам, например многорядной модели или модели, включающей перекрестки. Оба этих обобщения требуют перехода к двумерной сетке и серьезно усложняют процесс обновления состояния автомата, так как многополосное движение требует также на каждом шаге оценивать необходимость и возможность

смены полосы, а также требует создания способа решения конфликта при попытке перейти на одну клетку с двух различных соседних полос. Создание перекрестка же, приводит к необходимости обрабатывать движение по непараллельным направлениям через одни и те же клетки, что может быть реализовано через клетки, содержащие лишь часть машины. Кроме того, создание перекрестка добавляет необходимость учитывать приоритет дорог, что может серьезно усложнить правила перехода или процедуру их применения.

Модель перекрестка.

Постановка задачи и определение СА. Необходимо разработать модели СА для моделирования организации движения на дороге, представленной в виде графа, содержащего Т-образный, крестообразный и неперпендикулярный перекресток, согласно заданному набору условий проезда перекрестка (знаки приоритета, разрешенные направления движения).

Определим СА как структуру $CA = \langle A, C, \tau, X, T \rangle$, где $A = \{0,1\}$ — множество состояний клетки, C — множество клеток, τ — функция перехода, меняющая состояние автомата, X — отношение соседства, T — множество тактов времени. Для определения состояния автомата в такт времени t применим τ к предыдущему такту $t-1$. Введем M — множество объектов, движущихся в системе, и $m(c,t)$ — функцию от клетки автомата и такта времени.

$$m(c,t) = \begin{cases} \text{empty}, & \text{if } A(c,t) = 0, t \in T, \\ m(c,t) \in M, & \text{else.} \end{cases}$$

Каждый объект характеризуется двумя параметрами — направлением движения и идентификатором. Клетки, принадлежащие множеству C , определяются двумя разными размерами полного и половинного. Один объект занимает одну клетку полного размера или две клетки половинного размера. Отношение соседства X неодинаково у разных клеток и сильно зависит от положения клетки относительно перекрестков. Для задания τ введем правила для каждой клетки, после чего применение τ на какой-то конфигурации автомата определим как применение правил клеток в определенном порядке.

Пусть $Rules(c_i)$ — множество правил клетки c_i , $Name(r_{ij})$ — имя правила $r_{ij} \in Rules(c_i)$. $Imp(r_{ij})$ и $Next(r_{ij})$ — множество важных и следующих клеток для правила r_{ij} , $Imp(r_{ij}) \in X(c_i)$. Количество клеток $Imp(r_{ij})$ неограниченно, тогда как мощность $Next(r_{ij})$ один, если происходит переход на клетку полного размера, и два, если половинного.

Основные правила также имеют общую структуру:

$$\begin{array}{ll} \text{if } (\forall c_k \in Imp(r_{ij})) A(c_k, t) = 0 & \\ \text{then} & A(Next(r_{ij}), t+1) = 1; \\ & m(Next(r_{ij}), t+1) = m(c_i, t); \\ & A(c_i, t+1) = 0; \\ \text{else} & A(c_i, t+1) = 1; \\ & m(c_i, t+1) = m(c_i, t). \end{array}$$

Структура дополнительных правил для второй части объекта имеется только у клеток половинного размера. Так как для этих клеток необходимо иметь возможность отличать объекты друг от друга, введем функцию $Id(m)$, возвращающую идентификатор объекта. $Id(\text{empty}) = \text{empty}$. Тогда правило для второй части имеет следующий вид:

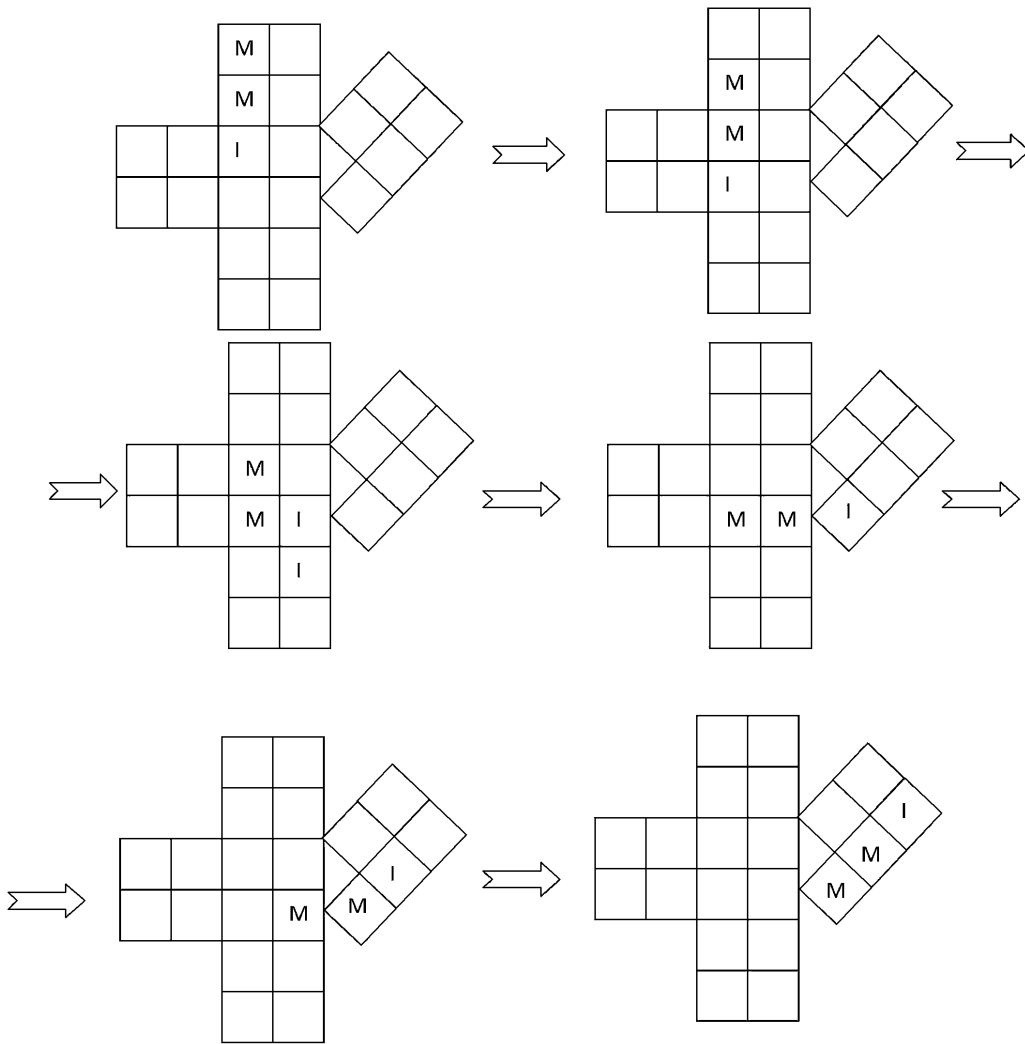


Рис. 1. Схема движения правила поворота налево, где M — клетка, содержащая машину, I — клетка из множества $Imp(r_{ij})$ описываемого правила

$$\begin{array}{ll}
 \text{if } (\forall c_k \in Imp(r_{ij})) Id(m(c_i, t) \neq Id(m(c_i, t))) & \\
 \text{then} & A(c_i, t + 1) = 0; \\
 \text{else} & A(c_i, t + 1) = 1; \\
 & m(c_i, t + 1) = m(c_i, t).
 \end{array}$$

Правило для пустой клетки тривиально — клетка не влияет на соседние клетки и не меняет свое состояние без воздействия других клеток. Каждая клетка имеет набор правил для движения через нее в любом разрешенном, согласно правилам дорожного движения, направлении. Для выбора подходящего правила используется функция $Dir(m_i, t)$, возвращающая направление движения объекта m_i в такт времени t . Области значений функций Dir и $Name$ совпадают, и на каждом шаге выбирается то правило, у которого $Name$ совпадает с Dir объекта, содержащейся в этой клетке.

Зададим порядок применения правил на каждом такте. Будем считать, что правило, меняющее состояние клетки, помечает ее, и в дальнейшем для этой клетки правила

на этом шаге не применяются. Порядок применения правил для одной полосы прямого участка — против движения объектов по этой полосе. Порядок применения правил для каждого перекрестка и прилегающих к нему прямых участков следующий — сначала применяются правила для полос прямых участков, ведущих с перекрестка, затем обрабатывается перекресток, затем полосы, ведущие на этот перекресток. Будем считать, что от смены порядка, в котором обрабатываются перекрестки и соответственно прилежащие к ним прямые участки, результат обновления не сильно меняется. Правила клеток не учитывают приоритет дорог на перекрестке, поэтому для его реализации будем использовать порядок обхода клеток.

Построение СА на различных участках

Прямые отрезки. Клеточная структура прямых содержит только клетки полного размера. Правило на прямых отрезках только одно — если следующая клетка по ходу движения пуста, переместить объект в следующую клетку. У последней по ходу движения клетки прямого участка правило немного отличается, из-за того что объект с нее переходит не на клетку полного размера, а на клетки половинного размера. Как следствие, Next этого правила содержит две клетки, а не одну.

Перекресток. Правила на перекрестке можно условно поделить на три группы — движение прямо, движение вправо и движение налево. Движение прямо почти не отличается от правила прямого участка — если клетка перед машиной пуста, то переместиться на одну клетку вперед. Главным отличием от правил прямого участка является сниженная скорость движения, так как за один шаг объект сдвигается только на половину своего размера. Правила поворота направо имеют схожую идею — проверить, свободна ли следующая по траектории клетка, и переместить объект в следующие клетки. Правила движения налево отличаются, так как необходимо учитывать возможную помеху справа. На основе данной модели также созданы две модификации — Т-образный перекресток [12] и перекресток с неперпендикулярными дорогами. Т-образный перекресток — это упрощение случая крестообразного перекрестка, у которого отсутствует одна из второстепенных дорог, в то время как для перекрестка с углами, отличающимися от 90 на угол от 30 до 60 градусов, создаются дополнительные клетки для предотвращения резких скачков. На рис. 1 приведена схема движения правила поворота налево, где M — клетка, содержащая машину, I — клетка из множества $Imp(r_{ij})$ описываемого правила.

Реализация. Модель перекрестка, описанная выше, была реализована с использованием языка C#. В программе используются два основных класса для хранения клеток — *intersite* и *line*, реализующие соответственно перекресток и полосу. Оба этих класса содержат какое-то количество объектов класса *cell*, реализующего сами клетки. Для хранения правил используется класс *rule*. Объекты классов *intersite* и *line* передаются в класс *intersitegraphic*, осуществляющий визуализацию текущего состояния и подающий команду на смену состояния автомата.

Класс *cell* содержит 4 поля. Поле *rules* используется для хранения правил клетки, поля *machine* и *sd* используются для реализации функций, связывающих клетку и машину, содержащую ее, поле *finished* используется при вычислении значения клетки на следующем шаге автомата, а именно, когда идет обновление какой-либо клетки, т.е. применяется правило, то у всех клеток, затронутых этим правилом, поле *finished* переходит в состояние *true*, и в дальнейшем правила этих клеток не используются до следующего шага.

Основные методы класса cell. Метод *move()* — это реализация функционирования основных правил клетки, кроме того, так как эта же функция содержит реализацию выбора правила для конкретной машины, в нее встроена и реализация функции *Dir* модели.

Метод *move_back()* — соответственно реализация дополнительных правил для второй части, в нем также имеется реализация функции выбора правила для машины.

Метод *get_machineid()* — это реализация функции *Id* из модели, также этот метод используется для визуализации для обработки половин машин.

Метод *nextDest()* — меняющий направление машины, содержащейся в клетке, на следующее согласно маршруту этой машины, этот метод используется при покидании машиной перекрестка, но так как данные о машине хранятся в клетке, реализован он также в самой клетке.

Также у класса *cell* есть группа служебных методов, использующихся для получения данных для других компонент или изменения состояния клетки другими компонентами, такие как метод для сброса флага обработки, создание машины в клетке и другие.

Класс *line* используется для хранения клеток одной полосы прямого участка. Поле класса *cellLine* отвечает за собственно хранение клеток этой полосы. Все клетки этого массива являются полноразмерными клетками, нумерация клеток соответствует направлению движения машин на этой полосе. Поля *begin* и *end* содержат перекрестки, соединенные этой полосой. Одно из этих полей может быть пустым для полос, связанных с объектами вне обрабатываемой группы, эти поля используются для генерации маршрута для машин. Поле *key* содержит идентификатор полосы, идентификаторы полос используются для генерации правил движения и для связи с соответствующими входными и выходными клетками перекрестка.

Основными методы класса *line*.

Метод *createRules* создает правила для движения по прямой во всех клетках полосы, кроме последней, правила для которой генерируются перекрестком.

Метод *update* применяет правила в клетках на полосе в порядке, обратном направлению движения на полосе.

Класс *intersite* является классом, реализующим перекресток, именно экземпляры этого класса содержат клетки половинного размера.

Поле *connected* содержит крайние клетки полос, соседствующих с перекрестком, и служит для связи перекрестков и полос при генерации правил. Поля *from* и *middle* содержат клетки половинного размера. Поля *from* — это отрезки по две или три клетки половинного размера, соответствующие продолжению полосы, входящей и выходящей из перекрестка, в то время как поле *middle* — это массив клеток, лежащих в центре перекрестка. В текущей реализации перекрестка этих клеток всегда четыре. Поля *position* и *angle* используются для визуализации перекрестка и содержат соответственно координаты перекрестка и угол поворота.

Основные методы класса *intersite*.

Метод *createRules* — это метод, генерирующий правила для движения через перекресток. Он использует три метода для создания правил для движения прямо, направо и налево для двух заданных полос: *createRulesLeft*, *createRulesRight*, *createRulesForForward*.

Роль метода *update* осталась прежней, но алгоритм метода изменился:

— обновить полосы, выходящие из перекрестка,

— обработать правила движения клеток, примыкающие к полосам, выходящим из перекрестка,

и список всех полос и всех перекрестков подается конструктору окна приложения. После каждого обновления вычисляются координаты клеток на экране, соответствующих непустым клеткам, после чего клетки, соответствующие передней и задней частям автомобиля, объединяются в одну с использованием взаиморасположения передней и задней частей машины.

На рис. 2. представлена UML диаграмма классов программы.

Заключение. В ходе работы проведено подробное исследование клеточных автоматов для моделирования транспортных потоков. Данная модель удовлетворяет правилам дорожного движения в идеальных условиях, когда все машины движутся с фиксированной скоростью и все водители соблюдают правила.

Разработан клеточный автомат для моделирования движения транспорта для группы перекрестков, также написана программа, реализующая генерацию автомата для группы перекрестков и использующая этот автомат для моделирования движения на этом перекрестке.

Программа написана на языке C# и состоит из двух основных частей — создание автомата для заданной группы перекрестков и моделирование движения и визуализация текущего состояния. Для визуализации текущего состояния используются средства библиотеки .NET Framework 4, а именно классы Windows Forms. Для визуализации модели на разных шагах итерации используется графический интерфейс, созданный с использованием библиотеки WinAPI.

Список литературы

1. Кравченко П. С., Омарова Г. А. Микроскопические математические модели транспортных потоков. Аналитический обзор // Проблемы информатики. № 1. 2014.; С. 71–78.
2. Гасников А. В. и др. Введение в математическое моделирование транспортных потоков. МФТИ: Москва. 2010.
3. Швецов В. И. Математическое моделирование транспортных потоков // Автоматика и Телемеханика. 2003. № 11. С. 3–46.
4. Дж. фон Нейман. Теория самовоспроизводящихся автоматов. М.: „Мир“, 1971.
5. Cremer M., Ludwig J. A fast simulation model for traffic flow on the basis of Boolean operations // Math. Comp Simul. 1986. V. 28. P. 297–303.
6. Nagel K., Schreckenberg M. A cellular automaton model for freeway traffic // Phys. I France. 1992. V. 2. P. 2221–2229.
7. Simon, P. M., Gutowitz, H. A. A Cellular Automaton Model for Bi-Directional Traffic. Phys. Rev. E 57. 2441. 1998.
8. Nowak S., Schadschneider A. A Cellular Automaton Approach for Lane Formation in Pedestrian Counterflow // Traffic and Granular Flow '11. P. 149–160.
9. Torsten Held, Stefan Bittihn: Cellular automata for traffic simulation — Nagel-Schreckenberg model // Project report in Computational Physics. Bonn, 2011.
10. Hafstein S. F., Chrobok R., Pottmeier A., Wahle J., Schreckenberg M. A High Resolution Cellular Automata Traffic Simulation Model with Application in a Freeway Traffic Information System // Computer-Aided Civil and Infrastructure Engineering. V. 19. I. 5. P. 338–350. 2004.
11. Zamith M., C'elia R., Leal-Toledo P., Kischinhevsky M., Clua E., Brand'ao D., Montenegro A., B. Lima Edgar. A Probabilistic Cellular Automata Model for Highway Traffic Simulation // Procedia Computer Science 1. 2012. P. 337–345.
12. Омарова Г. А., Казанцев Г. Ю. Применение клеточных автоматов для моделирования транспортных потоков // Проблемы информатики. № 3. 2015. С. 15–21.



Казанцев Григорий Юрьевич — студент НГУ; e-mail: kazantsev.grigorij@gmail.com.

Казанцев Григорий получил диплом бакалавра на механико-математическом факультете Новосибирского государственного университета в 2016 году. В настоящее время учится в магистратуре механико-математического факультета Новосибирского государственного университета. Его текущим исследовательским интересом являются клеточные автоматы и их применение для моделирования транспортных потоков.

Kazantsev Grigory received his bachelor degree in Mathematics from Novosibirsk State University (2016). Now he is learning for magister degree in Novosibirsk State University. His current research interests include cellular automata and their implementation for modeling traffic.



Омарова Гульзира Алимовна — канд. физ.-мат. наук, науч. сотр. Института вычислительной математики и математической геофизики СО РАН; e-mail: gulzira@rav.sccc.ru.

Гульзира Омарова окончила механико-математический факультет Новосибирского государственного университета в 1995 году. В 1997 г. окончила магистратуру механико-математический факультет Новосибирского государственного университета. В 2007 году за-

щитила кандидатскую диссертацию по специальности 05.13.18 в Институте вычислительной математики и математической геофизики СО РАН. С 2000 года по настоящее время работает в Институте вычислительной математики и математической геофизики СО РАН. Г. Омарова занимается математическим и имитационным моделированием в области сетей и графов. Ей опубликовано более 30 работ в таких областях, как построение и исследование алгоритмических моделей на сетях и графах, математическое и имитационное моделирование в исследовании различных транспортных ситуаций. В данное время основным проектом Г. Омаровой является микро- и макро моделирование транспортных потоков.

Gulzira Omarova graduated from mekhaniko-mathematical faculty of Novosibirsk State University in 1995, received her M. S. degree in Mathematics from the State University of Novosibirsk (1997), Ph.D. degree in Computer Science (2007) from the Institute of Computational Mathematics and Mathematical Geophysics of the Siberian Branch of the Russian Academy of Sciences (SB RAS). Since 2000 till present works at the Institute of Computational Mathematics and Mathematical Geophysics of SB RAS. G. Omarova is engaged in mathematical and simulation modeling in the field of networks and graphs. She published over 30 papers in such areas as construction and research of algorithmic models on networks and graphs, mathematical and simulation modeling in research of different transport situations. Now is engaged mathematical models of transport flows.

Дата поступления — 29.06.2016