УДК 004.4`2
Коцюба А.Ю., к.ф-м.н.
Луцький національний технічний університет

ЗАСТОСУВАННЯ ДВОВИМІРНИХ КЛІТИННИХ АВТОМАТІВ В МОДЕЛЮВАННІ ПРАВОСТОРОННЬОГО РУХУ ТРАНСПОРТНОГО ЗАСОБУ ПО СТВОРЕНІЙ КОРИСТУВАЧЕМ СХЕМІ ДОРІГ

Коцюба А.Ю. Застосування двовимірних клітинних автоматів в моделюванні правостороннього руху анспортного засобу по створеній користувачем схемі доріг. Засобами C++ Builder розроблено програму-конструктор, яка дозволяє будувати довільну схему доріг з перпендикулярними або паралельними напрямками переміщення. В даному програмному продукті реалізовано ще й так званий компілятор. Він не дозволить створити результат (web-сторінку з автоматично створеним на ній JavaScript-кодом, на якій відбувається безперервний правосторонній рух транспортного засобу та може вирішуватися задача пошуку найкоротшого маршруту) до того часу, доки не будуть виправлені всі помилки (недоліки при конструюванні, що можуть завадити безперервності руху). Створено алгоритм, за допомогою якого будується відповідний сконструйованій користувачем схемі доріг двовимірний клітинний автомат.
Ключові слова: моделювання, двовимірний клітинний автомат, правосторонній рух транспортного засобу, пошук найкоротшого маршруту, алгоритм Дейкстри.

Коцюба А.Ю. Применение двумерных клеточных автоматов в моделировании правостороннего движения транспортного средства по созданной пользователем схеме дорог. Средствами C++ Builder разработана программа-конструктор, которая позволяет строить произвольную схему дорог с перпендикулярными или параллельными направлениями перемещения. В данном программном продукте реализовано еще и так называемый компилятор. Он не позволит создать результат (web-страницу с автоматически созданным на ней JavaScript-кодом, на которой происходит непрерывное правостороннее движение транспортного средства и может решаться задача поиска кратчайшего маршрута) до тех пор, пока не будут исправлены все ошибки (недостатки при конструировании, что могут помешать непрерывности движения). Создан алгоритм, с помощью которого строится соответствующий сконструированной пользователем схеме дорог двумерный клеточный автомат.
Ключевые слова: моделирование, двумерный клеточный автомат, правостороннее движение транспортного средства, поиск кратчайшего маршрута, алгоритм Дейкстры.

Kotsyuba A.Yu. The use of the two-dimensional cellular automata in the right-hand vehicle traffic movement in agreement with a road scheme created by user. A design program is developed by C++ Builder tools that allows you to build an arbitrary scheme for the road with perpendicular or parallel directions to move. There also compiler is realized in this software. It doesn't above to create some results (web-page with automatically created JavaScript code in it and which the continuous right-hand vehicle traffic and the find of the shortest route problem can be solved) up to the time until all errors are corrected (the design mistakes that may interfere to the movement continuity). The algorithm is created by which, in agreement with the user constructed road scheme, a two-dimensional cellular automata is developed.
Keywords: modeling, two-dimensional cellular automata, right movement of the vehicle, finding the shortest route, algorithm of Dijkstra.

Постановка та актуальність проблеми. Оскільки на дорогах досить часто виникають проблеми (затори, аварійні ситуації, необхідність пошуку найкоротшого оптимального маршруту тощо), які заслуговують на те, щоб їх детально вивчали та вирішували, то моделюванню руху транспортного засобу по схемі доріг присвячено багато наукових праць [1-3]. Оскільки кількість таких проблем є досить значною, то це призводить до того, що і множина методів моделювання є також великою. З моменту публікації праці [4] в 1992-му році основною парадигмою побудови мікромоделей дорожнього руху став механізм з використанням клітинного автомата [5, 6] для представлення стану дорожнього полотна. Такий механізм має як недоліки, так і переваги. Мабуть, однією з найсуттєвіших переваг цього механізму є те, що для вирішення більшості проблем є можливість побудови математичної моделі і, як результат, побудови алгоритму та реалізації його у вибраному користувачем середовищі програмування (в даній роботі було вибрано середовище C++ Builder [7, 8] і приділено особливу увагу проблемі побудови довільної, хоча і дещо спрощеної, і, як результат, не завжди відповідної реальній, схеми доріг (рис. 1). З вищесказаного випливає, що слід звернути увагу ще й на один із найсуттєвіших недоліків, який полягає у тому, що для побудови довільної реальної схеми дорожнього полотна необхідно цей механізм ускладнювати до такого рівня, що моделювати за допомогою сучасних ПК буде або неможливо, або досить складно, або громіздкими і, як результат, повільними будуть відповідні програмні реалізації.

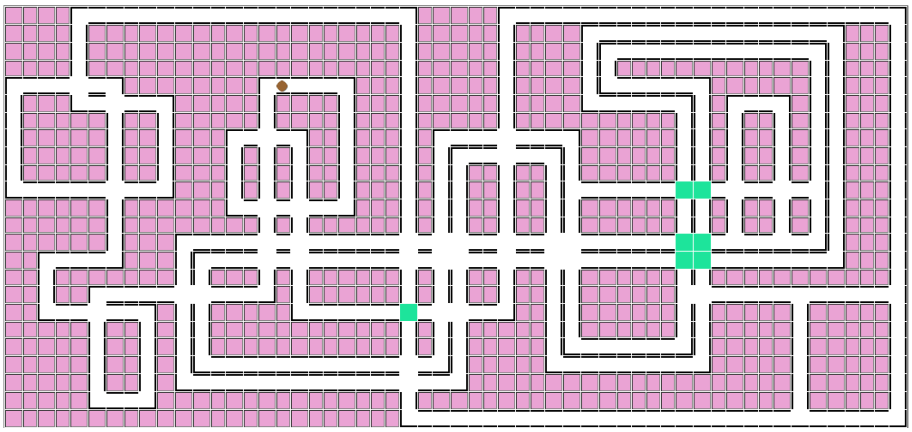

Рис. 1. Один із випадків схеми доріг

Актуальність даної роботи полягає у тому, що мало хто вирішував проблему побудови довільної схеми доріг таким чином, щоб користувач, далекий до програмування, зміг би самостійно сконструювати дорожнє полотно, на якому можна буде моделювати будь-які дорожні проблеми. Наприклад, пошук найкоротшого маршруту за правилами правостороннього руху від місцезнаходження транспортного засобу у даний момент часу до місця, вказаного користувачем, за умови, що транспортний засіб весь час (до того часу, як користувач задав кінцевий пункт, і після того, як цей пункт буде пройдено) знаходиться у неспинному випадковому русі також згідно правил правостороннього руху. Власне вирішенню цієї проблеми і присвячена дана робота.

Таким чином **метою** даної роботи є побудова методики, за допомогою якої програма-конструктор схеми доріг (рис. 2), що розроблена в середовищі C++ Builder, зможе самостійно створювати такі web-сторінки (рис. 1) з відповідним javaScript-кодом, на яких буде вирішуватися за допомогою алгоритму Дейкстри вищеописана проблема. Основною парадигмою для побудови цієї методики буде реалізація двовимірного клітинного автомата. Але спосіб такої реалізації буде оригінальним. І докладніше його опишемо нижче.

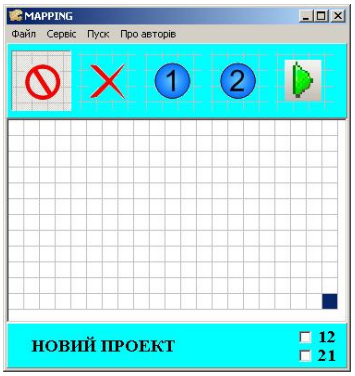

Рис. 2. Загальний вигляд програми-конструктора схеми доріг

Зауважимо, що розроблена програма містить у собі компілятор, який не дозволить програмі створити web-сторінку до того часу, доки всі помилки не будуть виправлені (деякі з них виправляються автоматично, а з іншими необхідно користувачу попрацювати самостійно). Детальніше про можливі помилки буде розказано далі.

Побудова двовимірних клітинних автоматів. Як правило, розглядають клітинний автомат, який є детермінованим автоматом Мура, тобто окремим випадком автомата Мілі, для якого функція переходів є функцією, а не відношенням. Покажемо, що цей підхід не завжди себе оправдовує. Для цього спочатку введемо ці поняття за допомогою визначень.

П'ятірка (A, X, Y, f_A, g_A) називається автоматом Мілі, якщо вона складається із множини станів автомата A, множини вхідних символів X, множини вихідних символів Y, функції переходів $f_A : A \times X \to A$ і функції виходів $g_A : A \times X \to Y$. Множини X і Y називають відповідно вхідним і вихідним алфавітами автомата. Як правило, автомат позначають символом

множини його станів, тобто $A=(A,X,Y,f_A,g_A)$. Коли відомо, про який автомат іде мова, то індекс A в символах функцій переходів і виходів опускають.

Якщо $f(a,x)=a'$, то кажуть, що автомат A під дією вхідного сигналу $x\in X$ переходить в стан a', або сигнал x переводить автомат A із стану a в стан a'. Якщо $g(a,x)=y$, то говорять, що автомат A перетворює в стані а вхідний сигнал $x\in X$ у вихідний сигнал $y\in Y$.

Інколи трапляються автомати, в яких компонент f – не функція, а деяке відношення, тобто в таких автоматах не виконується умова однозначності переходу. Автомати такого типу називаються недетермінованими. Якщо ж відношення f є функцією, то автомат називається детермінованим. Отже, для детермінованого автомата A з початковим станом a однозначно знаходиться стан b, в який автомат перейде під дією слова $p\in F(X)$. Ясно, що клас детермінованих автоматів є підкласом недетермінованих автоматів.

Автомат $A=(A,X,Y,f,g)$ називається скінченним, якщо всі три множини – A, X і Y – скінченні, і нескінченним, якщо хоч одна з них нескінченна.

Як уже було зазначено автомати Мура є окремим випадком автоматів Мілі. Автомат (A,X,Y,f,g) називається автоматом Мура, якщо його функція виходів $g(a,x)$ виражається функцією переходів $f(a,x)$ за допомогою рівняння $g(a,x)=h(f(a,x))$, де $h:A\to Y$. Функція h називається функцією відміток автомата, а її значення $h(a)$ на стані a – відміткою цього стану.

Закони функціонування скінченного автомата Мілі можна представити у вигляді

$$a(t+1)=f(a(t),x(t)),\ y(t)=g(a(t),x(t)); \qquad (1.1)$$

де t – поточний момент часу; $t+1$ – наступний момент часу; $a(t+1)$ – стан автомата в наступний момент часу; $a(t)$, $x(t)$, $y(t)$ – елементи опису автомата в поточний момент часу.

Закони функціонування скінченного автомата Мура можна представити у вигляді

$$a(t+1)=f(a(t),x(t)),\ y(t)=g(a(t)). \qquad (1.2)$$

Для представлення необхідних для вирішення вищеописаної проблеми автоматів, визначимо множини A, X та Y. Для цього приймемо, що схема доріг є скінченною та має розмірність $m\times n$, тобто множину станів можна представити як підмножину множини усіх елементів матриці $\{a_{i,j}\}_{m\times n}$, де $i=\overline{0,m-1}$, $j=\overline{0,n-1}$, множину вхідних символів задамо у вигляді $X=\{\downarrow,\uparrow,\leftarrow,\rightarrow\}$, а вихідних – $Y=\{\Downarrow,\Uparrow,\Leftarrow,\Rightarrow\}$. Оскільки для кожної схеми доріг існує свій автомат, який може і не бути детермінованим, то опишемо спочатку лише всі можливі правила переходу з одного стану в інший, які є характерними для правостороннього руху транспортного засобу і для кожного з яких можна побудувати функції переходів та виходів (табл. 1)

Таблиця 1 – Правила, що характеризують всі входи та виходи з клітини

№	Правила	Коментар
0		Один вхід – один вихід.
1		
2		
3		
4		
5		

№	Правила	Коментар
6		
7		
8		
9		
10		
11		

№	Правила	Коментар
12		Один вхід – два можливих виходи, другий вхід – також два можливих виходи.
13		
14		
15		
16		
17		
18		Один вхід – два можливих виходи.
19		
20		
21		Один вхід – один вихід, другий вхід – також один вихід.
22		
23		
24		Один вхід – два можливих виходи.
25		
26		
27		Один вхід – один вихід, другий вхід – також один вихід.
28		
29		
30		Один вхід – два можливих виходи.
31		
32		
33		Один вхід – один вихід, другий вхід – також один вихід.
34		
35		
36		Один вхід – два можливих виходи.
37		
38		

№	Правила	Коментар
39		Один вхід – один вихід, другий вхід – також один вихід.
40		
41		
42		Один вхід – три можливих виходи.
43		Один вхід – один вихід, другий вхід – один вихід, третій вхід – також один вихід.
44		Один вхід – три можливих виходи.
45		Один вхід – один вихід, другий вхід – один вихід, третій вхід – також один вихід.
46		Один вхід – три можливих виходи.
47		Один вхід – один вихід, другий вхід – один вихід, третій вхід – також один вихід.
48		Один вхід – три можливих виходи.
49		Один вхід – один вихід, другий вхід – один вихід, третій вхід – також один вихід.
50		Назвемо такі клітини мостами, у них: один вхід – один вихід, другий вхід – один вихід з тією лише різницею, що для різних входів – різні виходи.
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		
61		
62		Порожня клітина: вона ніколи не може бути активована ($\xi \equiv 0$).

Роз'яснимо, наприклад, як трактувати правило № 41. Це правило означає, що для фіксованого стану $a_{i,j}$ функція переходу може задаватися залежностями: $f\left(a_{i,j},\rightarrow\right)=a_{i+1,j}$ або $f\left(a_{i,j},\leftarrow\right)=a_{i+1,j}$. При цьому автомат зберігатиме свою детермінованість лише за умови, що існує стан $a_{i+1,j}$ (компілятор програми-конструктора повинен перевірити, щоб ще й існувало

значення функції $f\left(a_{i+1,j},\downarrow\right)$). В разі, коли стан $a_{i+1,j}$ не існує, то користувач повинен задати скінченну множину таких станів $\left\{a_{k,l}\right\}$, для яких існує відповідна множина значень функції $\left\{f\left(...,\downarrow\right)\right\}$ і при цьому повинні не існувати стани $\left\{a_{k-1,l}\right\}$. У цьому випадку стан $a_{i,j}$ називатимемо виходом, а множину станів $\left\{a_{k,l}\right\}$ входами, що відповідають даному виходу. Схему, у якій немає виходів, називають замкненою (якщо є виходи, то повинні бути і входи, хоча може бути, що є так званні початкові входи, а виходів немає; компілятор це відслідковує), Програма випадковим чином сама повинна вибирати один із заданих користувачем входів (при цьому компілятор не допустить, щоб виходу була поставлена у відповідність $\varnothing$, бо це призведе до зупинки транспортного засобу, а ми зацікавленні, щоб рух був неспинний). Очевидно, що при цьому, якщо множина входів не є одноелементною, автомат втрачає свою детермінованість. Вона втрачається ще й для всіх схем, у яких зустрічаються правила, що мають декілька символів виходу, крім так званих мостів (правила №50-61), для них характерною є така особливість, що у функціях виходів значення залежать від символу входу взаємнооднозначно (тобто кожному символу входу відповідає лише один символ виходу). Щодо функцій виходів, то для заданого правила вони матимуть вигляд $g\left(a_{i,j},\rightarrow\right)=\Downarrow$ або $g\left(a_{i,j},\leftarrow\right)=\Downarrow$.

Аналогічно можна проаналізувати і всі інші правила.

Зазначимо, що для випадків декількох можливих входів або виходів алгоритм будується таким чином, що в разі пошуку оптимального найкоротшого маршруту випадковість вимикається, можливі варіанти вибираються обдумано, при досягненні кінцевого пункту випадковість знову вмикається і транспортний засіб без затримки прямує далі.

Висновки. Отже, клітинний автомат використовується для моделювання руху транспортного засобу вже понад 20 років. При цьому дана проблема у зв'язку з великою різноманітністю задач, які можна вирішити таким способом не перестає бути актуальною і на даний час. Спосіб застосування (чи побудови відповідного) клітинного автомата у більшості праць присвячених даній тематиці зазвичай відрізняється. В даній роботі розроблено власний спосіб застосування, за допомогою якого можна вирішувати будь-яку з вищеописаних проблем, а не лише пошук найкоротшого маршруту. Але якщо зупинитися лише на останній проблемі, то для її вирішення можна використовувати не лише алгоритм Дейкстри (існує велика кількість алгоритмів, які дозволяють шукати найкоротший маршрут: мурашиний алгоритм, алгоритми, що базуються на нейронних мережах тощо). І порівняльний аналіз, оптимізація пошуку тощо – це ще одні із напрямків досліджень, які можна проводити за допомогою розробленої в роботі програми-конструктора.

Список використаних джерел:

1. http://www.transport.ru/1/12/i31_4004p0.htm
2. http://www.kommersant.ru/doc.aspx?DocsID=801719
3. http://googleblog.blogspot.com/2007/02/stuck-in-traffic.html
4. Nagel K. A cellular automaton model for freeway traffic / K. Nagel, M. Schreckenberg // J. Physique I France. – 1992, vol. 2 – P. 2221-2229.
5. http://cas.ssu.runnet.ru/sgnp/data/papers/Train/CellAutomat.pdf
6. Капітонова Ю.В. Основи дискретної математики / Ю.В Капітонова, С.Л Кривий, О.А Летичевський та ін. – Київ: "Наукова думка", 2002. – 580 с.
7. Пахомов Б. Самоучитель С/С++ и С++ Builder 2007 / Б. Пахомов. – Санкт-Петербург: "БХВ Петербург", 2008. – 672 с.
8. Бруно Бабэ. Просто и ясно о Borland C++ / Бабэ Бруно. – М., 1996. – 400 с.