Automat komórkowy ze zmianą reguł.

1. Co to są automaty komórkowe?

Automaty komórkowe są to narzędzia wykorzystywane do modelowania

układów dynamicznych. Przykładem tego są, proste zasady i lokalne

oddziaływania mogące prowadzić do bardzo różnorodnych i skomplikowanych zachowani. Zwykle jako automaty komórkowe wyróżnia się:

– regularną 1D, 2D ... siatkę;  
– każdy węzeł siatki znajduje się w jednym z *N* dyskretnych stanów. Np. w modelu SIR moglibyśmy mieć :

– stany węzłów uaktualniane są synchronicznie w dyskretnych momentach

– stan każdego węzła w chwili *t + 1* jest deterministyczną funkcją stanu

sąsiadów w chwili *t*

1.1. Typ komórki: W najprostszym przypadku jest to pojedyncza wartość logiczna przyjmująca jeden z dwóch stanów – prawdy lub fałszu. Może to być wartość wyliczeniowa, liczba całkowita, zmiennoprzecinkowa, wektor czy nawet rozbudowana struktura złożona z wielu pól.

1.2. Stan początkowy: Może to być wypełnienie całej tablicy określoną wartością, wartościami losowymi, zapamiętanymi wcześniej w pliku czy wczytanymi z innego źródła (np. na podstawie obrazu graficznego) albo wygenerowanymi przez pewien algorytm.

1.3. Funkcja przejścia: Tak jak dyskretną przestrzeń w automatach komórkowych stanowi tablica komórek, tak dyskretny czas wyznaczają kolejne cykle przeliczania nazywane dalej iteracjami. Algorytm może w każdej iteracji, według pewnych określonych reguł przeliczać kolejno wszystkie komórki, tylko jedną (na pewnej pozycji pamiętanej między iteracjami jako bieżąca), jeden wiersz tablicy itp. Dla każdej przeliczanej komórki wyliczany jest jej nowy stan na podstawie pewnych parametrów, przede wszystkim jej stanu aktualnego i stanu komórek sąsiednich.

1. Historia powstania automatów komórkowych.

Twórca automatów komórkowych jest jeden z największych myślicieli ery komputerowej – *John von Neumann*. Docelowo chciał stworzyć model maszyny samosterującej, tzn. takiej, iż powielałaby ona swoja budowę i przekazywała swoje cechy. Na przełomie lat czterdziestych i pięćdziesiątych Neumann opracował swoja teorie opierając się na maszynie Turinga. Opracował on piec modeli

samo-replikujących się automatów, realizacja jednak okazała się zbyt trudna jak na owe czasy. Pracami Neumanna zainteresował się dopiero Edgar Frank Codd, który

uczynił automaty możliwymi do wykorzystania. Codd zaprojektował automat komórkowy, który mógł obliczyć wszystkie możliwe funkcje, i który mógł się rozmnażać. Jednak mimo ze ten projekt zawierał o wiele prostsza koncepcje od pomysłu von Neumanna, również nie został zrealizowany. Posłużył natomiast do skonstruowania powszechnie stosowanej *Gry w życie* (J.H. Conway). Mimo ze w obu przypadkach brakowało realizacji projektów, prace obu teoretyków uważa się za fundamenty powstania automatów komórkowych. Następnym przełomowym wydarzeniem w historii automatów komórkowych było sklasyfikowanie ich. Po wcześniejszych czysto teoretycznych projektach w 1983 roku Stephen Wolfram dokonuje klasyfikacji automatów komórkowych.

1. Opis reguł gry w życie.  
    Przepis na nowy stan węzła opisywany jest przez reguły. Jako przykład weźmy 1D automat dwustanowy i sasiedztwo .

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stan sąsiedztwa dla *t* | 111 | 110 | 101 | 100 | 011 | 010 | 001 | 000 |
| stan węzła dla *t* | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

Tak zapisaną regułę można odczytać jako liczbę binarną i przedstawić ją w systemie dziesiętnym jako 110. Z powyższej tabelki widzimy, że tak rozumianych reguł w tym przypadku mamy . Używana terminologia:

* legalcałe sąsiedztwo 0 maouje się na stan 0, i mamy izotropie
* totalisticstan węzła zależy tylko od sumy sąsiedztwa
* perypherialstan węzła nie wpływa na swój przyszły stan

Reguły można też podzielić pod względem wzorców do których ewoluują automaty komórkowe im podlegające. Wolfram (1984) zdefiniował 4 klasy:

1. ewoluują do ustalonego jednorodnego stanu;
2. ewoluują do stanu niejednorodnego lub cyklicznego;
3. ewoluują do zachowania chaotycznego lub aperiodycznego;
4. ewoluują do skomplikowanych zlokalizowanych struktur.
5. Diagram klas przedstawia klasy (typów) obiektów w programie, w odróżnieniu od diagramu obiektów, który pokazuje jedynie egzemplarze (instancje) obiektów i ich zależności istniejące w konkretnym momencie. Diagramów klas używa się do modelowania statycznych aspektów perspektywy projektowej. Diagramy klas pozwalają na sformalizowanie specyfikacji danych i metod. Mogą także pełnić role graficznego środka pokazującego szczegóły implementacji klas. Możliwe jest  korzystanie z ustawień parametrów gry, bez logowania się do systemu. W diagramie jak i w programie istnieje 8 klas. Większość z nich dziedziczy z głównej klasy „zmienne”, reszta z nich nie miała takiej potrzeby, ponieważ związane były z innymi okienkami w aplikacji.
6. Ogólny opis systemu:  
   - przycisk „Nowa symulacja” - przygotowuje okno do symulacji;

- przycisk „Pomoc” - wyświetla okno pomocy;

- przycisk „Rozpoczęcie symulacji” - rozpoczyna symulację;

- przycisk „Zatrzymanie symulacji” - zatrzymuję symulację;

- przycisk „Wyczyść planszę” - kasuje wykonaną symulację;

- przycisk „Następny krok symulacji” - przechodzi do następnego kroku symulacji;

- przycisk „Zmień szybkość symulacji” - zwiększa lub zmniejsza szybkość symulacji;

- przycisk „Zmiana rozmiaru siatki” - zwiększa lub zmniejsza rozmiar siatki;

- przycisk „Zmień szybkość symulacji” - zwiększa lub zmniejsza szybkość symulacji;

- przycisk „Zmiana skali siatki” - zwiększa lub zmniejsza skalę siatki;

- przycisk „Pokazanie/Ukrycie siatki” - pokazuję lub ukrywa siarkę;

- przycisk „Zmiana grubości siatki” - zwiększa lub zmniejsza grubość siatki;

- przycisk „Zmiana grubości siatki” - zwiększa lub zmniejsza grubość siatki;

- przycisk „Wybór koloru komórek” - wyznacza danej komórce kolor;

- przycisk „Wybór reguł” - wybieramy zdefiniowane reguły przez autora;

- przycisk „Zdefiniowanie własnej reguły” - definiujemy własną regułę;

- przycisk „Wczytanie poprzedniego stanu” - wczytuje wcześniej zapisaną symulację;

- przycisk „Zapisanie obecnego stanu” - zapisuje obecny stan symulacji.

6. Diagram Sekwencji zwany również diagramem interakcji. Umożliwia modelowanie aspektów aplikacji. Przedstawia działanie kilku obiektów do realizacji określonego zadania z podkreśleniem perspektywy czasowej z zaznaczeniem kolejności wystąpień w czasie. Biorą w nim udział obiekty, opisane nazwą obiektu oraz klasą, które wymieniają miedzy sobą komunikaty.