

Automaty komórkowe

1 Wstęp

1.1 Definicja

Automaty komórkowe są narzędziem wykorzystywanym czasem do modelowania układów dynamicznych. Są przykładem tego, że proste zasady i lokalne oddziaływania mogą prowadzić do bardzo różnorodnych i skomplikowanych zachowań. Zwykle jako automaty komórkowe rozumie się:

- regularną 1-D, 2-D ... siatkę.
- Każdy węzeł siatki znajduje się w jednym z N dyskretnych stanów. Np. w modelu SIR moglibyśmy mieć

$$x(i, j) = \begin{cases} 0 & \text{dla stanu podatnego} \\ 1 & \text{dla stanu zainfekowanego} \\ 2 & \text{dla stanu ozdrowiałego} \end{cases} \quad (1)$$

- stany węzłów uaktualniane są synchronicznie w dyskretnych momentach
- stan każdego węzła w chwili $t + 1$ jest deterministyczną funkcją stanu sąsiadów w chwili t
- najczęściej przyjmuje się okresowe warunki brzegowe (żeby nie było kłopotów z nierówną ilością sąsiadów na brzegach). Prowadzi to do topografii okręgu (1-D) lub torusa (2-D).

Łatwo wyobrazić sobie uogólnienia tego klasycznego sformułowania na stany ciągłe w każdym węźle i funkcje probabilistyczne na zmiany stanów.

1.2 Historia

Twórcą automatów komórkowych jest jeden z największych myślicieli ery komputerowej, który wprowadził koncepcję samo-reprodukcji - *John von Neumann*. Docelowo chciał stworzyć model maszyny samosterującej, tzn. takiej, iż powielalaby ona swoją budowę i przekazywała swoje cechy. Na przełomie lat czterdziestych i pięćdziesiątych Neumann opracował swoją teorię opierając się na maszynie Turinga. Opracował on pięć modeli samo-replikujących się automatów, realizacja jednak okazała się zbyt trudna jak na owe czasy. Trafiły one na półkę.

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 że ten projekt zawierał o wiele prostszą koncepcję od pomysłu von Neumanna, również nie został zrealizowany. Posłużył natomiast do skonstruowania powszechnie stosowanej *Gry w życie* (J.H. Conway).

Mimo że 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.

1.3 Sąsiedztwo

Sąsiedztwo von Neumanna Jeśli oznaczymy kierunki na zasadzie różnic wiatrów: N, S, E, W oraz kierunki pośrednie NW, NE, SE, SW, to sąsiedztwem von Neumanna będzie zbiór czterech komórek: N, S, E, W

Sąsiedztwo Moore'a Posługując się powyższymi oznaczeniami, sąsiedztwem Moore'a będzie zbiór wszystkich ośmiu komórek dookoła komórki centralnej

Sąsiedztwo Margolusa Stosuje się je w automatach do symulacji spadającego piasku, czy też interakcji cząsteczek gazu. Reguły przejścia opierają się na kwadratowych blokach tworzonych przez cztery sąsiadujące komórki. Stany tych sąsiednich komórek zmieniają się jednocześnie, przy czym komórki przyjmują wartości binarne 1 i 0. Tak dzieje się na całej siatce automatu. W następnym kroku reguły są obliczane podobnie, tylko że zmieniają się grupy komórek. Bloki tworzące owe grupy przesuują się o jeden w prawo i w dół.

1.4 Warunki brzegowe

Periodyczne Definiują one zamkniętą siatkę w taki sposób, że np. symulując poruszającą się cząstkę po dojściu do krawędzi pojawi się ona z drugiej strony. Komórka znajdująca się na brzegu siatki ma za sąsiada komórkę leżącą po drugiej stronie siatki.

Zamknięte pochłaniające siatka jest zdefiniowana w taki sposób, że brzeży siatki wypełnione są z góry ustaloną wartością, która poprzez funkcję przejścia ustala wpływ na zachowanie automatu. W praktyce, symulując np. cząstkę gazu, po przekroczeniu krawędzi siatki przestaje ona istnieć.

Zamknięte odbijające Warunki brzegowe na krawędzi siatki tworzą barierę, od której symulowane cząstki się odbijają. Stosowane do symulacji zamkniętych przestrzeni doświadczalnych.

1.5 Reguły

Przepis na nowy stan węzła x opisywany jest przez reguły. Jako przykład weźmy 1-D atomat dwustanowy $N = 2$ i sąsiedztwo $r = 1$.

stan sąsiedztwa dla t	111	110	101	100	011	010	001	000	Tak
stan węzła dla t	0	1	1	0	1	1	1	0	

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 $2^8 = 256$. Używana terminologia:

- *legal* całe sąsiedztwo 0 maouje się na stan 0, i mamy izotropię
- *totalistic* stan węzła zależy tylko od sumy sąsiedztwa $x(t+1) = f\left(\sum_{\text{sąsiad}} x(t)\right)$
- *peripheral* stan 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

1.6 Przykład

Poniżej przedstawiam prosty model 1-D automatu komórkowego, z sąsiedztwem o promieniu 1 i stanach $\in \{0, 1\}$. Przestrzeń ma topografię okręgu. Proszę eksperymentować z różnymi wartościami reguł i zaobserwować częstość występowania różnych klas.

```
clear all
close all
regula = 110;          % numer reguly
w = 100;              % ile wezlow
T = 100;              % ile iteracji

%init = round(rand(1,w)); % losowa inicjalizacja
init = zeros(1,w);    % deterministyczna inicjalizacja
init(w/2) = 1;
X(1,:) = init;

r = bitget(regula,8:-1:1); % konwersja reguly na ciag bitow
vec = [4 2 1];
% wektor pomocny przy odczytywaniu stanu sasiedztwa
% z tabelki reguly odpowiada zapalonym bitom
% 1- bit o 1 w prawo,
% 2- bit srodkowy,
% 4 - bit o jeden w lewo (porownaj tabelka)

for t=1:T
    Y = X(t,:);
    d(1,:) = [Y(end) Y(1:(end-1))]; % kopia stanu w
    % chwili t przesunięta o 1 w prawo
    % zakładamy topografie okregu wiec
    % przeklejamy ostani element na pierwszy
    d(2,:) = Y; % kopia stanu w chwili t
    d(3,:) = [Y(2:end) Y(1)]; % kopia stanu w
    % chwili t przesunięta o 1 w lewo
    % zakładamy topografie okregu wiec przeklejamy
    % pierwszy element na ostani

    X(t+1,:) = r(8-vec*d(:,,:));
bar(X(t+1,:))
pause(0.1)
drawnow
end

imagesc(X);
axis equal
axis tight
xlabel('przestrzen')
ylabel('czas')
title(['Regula: ', num2str(regula)])
```

1.7 Przykład dwuwymiarowego modelu SIR realizowanego przez automaty komórkowe

Założenia:

1. choroba trwa a jednostek czasu
2. po wyzdrowieniu uzyskuje się odporność na b jednostek czasu
3. po czasie $a + b$ od początku infekcji ponownie staje się podatnym
4. zarażeniu ulega każdy kto ma w *otoczeniu* przynajmniej jednego chorego

Poniższy kod proszę zasymulować po kilka razy dla następujących kombinacji parametrów $a \in \{2, 4, 8\}$, $b \in \{2, 4, 8\}$ i $p \in \{0.01, 0.05, 0.10\}$. Spróbujcie zinterpretować obserwowane wzory 2-D i przebiegi czasowe S , I , R .

```
clear all
close all

n = 100;          % rozmiar sieci n na n
a = 5;           % ilosc stanow infekcyjnych: 1,...,a
b = 14;          % ilosc stanow z odpornoscia: a+1,...,a+b
T=300;
p = 0.2;         % poczatkowa czestosc wystepowania stanow infekcyjnych i opornych
%Przygotowanie COLORMAP

X0 = [0 1 0];    % podatni sa zieloni
Xa = colormap(autumn(a)); % zarazeni sa zolto-czerwoni
Xb = colormap(winter(b)); % odporni sa niebiesko-zieloni
X= cat(1,X0,cat(1,Xa,Xb)); % cala colormap

% Stan poczatkowy

D1 = ceil((a+b)*rand(n,n)); % macierz wypełniona liczbami z rozkładu płaskiego od 1 do a+b
D2 = rand(n,n) < p;        %wybieramy wezly zarazeni i odporni
D = D1.*D2;

%% PROGRAM
image(D); colormap(X); colorbar ; % stan poczatkowy
axis square;

t = 1;
S(t) = sum(sum(D==0)); % poczatkowa ilosc podatnych
I(t) = sum(sum((D>0)&(D<a))); % zarazonych (D>0)&(D<b))
R(t) = sum(sum(D>a)); % odpornych

for t=2:T
```

```

D1 = D(:, [n 1:n-1]);          % stan sasiadow w lewo
D2 = D(:, [2:n 1]);           % stan sasiadow w prawo
D3 = D([n 1:n-1], :);        % stan sasiadow powyzej
D4 = D([2:n 1], :);          % stan sasiadow ponizej

sick_neighbors = ((D1>0)&(D1<a))+((D2>0)&(D2<a)) + ((D3>0)&(D3<a)) + ((D4>0)&(D4<a));
stay_healthy = ((D==0)&(sick_neighbors==0)); % podatni ktorzy nie zachoruja
get_sick = ((D==0)&(sick_neighbors>0));      % podatni ktorzy zachoruja
get_recovered = (D>=(a+b));                 % odporni ktorzy stana sie podatni
others = (D>0)&(D<a+b);                     % wszyscy pozostali

D(get_sick) = 1;                            % uaktualnianie stanu podatnych zaczynajacych chorobe
D(get_recovered) = 0;                        % uaktualnianie stanu odpornych tracacych odpornosc
D(others) = D(others)+1;                    % uaktualnianie stanu pozostalych

% rysujemy stan aktualny
image(D); colorbar; title(['t = ', num2str(t)])
drawnow
S(t) = sum(sum(D==0));                       % sumujemy podatnych
I(t) = sum(sum((D>0)&(D<a)));                 % sumujemy zakazonych
R(t) = sum(sum(D>a));                         % sumujemy odpornych
end

figure % S,I i R
t=1:T;
plot(t,S,t,I,t,R)
legend('podatni', 'zakazeni', 'odporni')
xlabel('czas')

```

1.8 Zadanie: Zaimplementować model Greenberga i Hastingsa.

Jest to prosty model typu reakcja-dyfuzja Założenia:

1. komórka $k_{i,j}$ znajduje się w jednym ze stanów $k_{i,j} \in \{0, 1, 2\}$, 0- stan równowagi, 1- stan przejściowy, 2 - stan wysycenia.
2. stan komórki jest sumą dwóch składników:

$$k_{i,j}(t+1) = R_{i,j}(t) + D_{i,j}(t)$$

reakcji

$$R_{i,j}(t) = \begin{cases} 2 & \text{dla } k_{i,j}(t) = 1 \\ 0 & \text{dla } k_{i,j}(t) = 0 \vee 2 \end{cases}$$

oraz dyfuzji

$$D_{i,j}(t) = \begin{cases} 1 & \text{dla } k_{i,j}(t) = 0 \text{ i co najmniej jeden z sąsiadów w stanie 1} \\ 0 & \text{dla pozostałych przypadków} \end{cases}$$