

Automaty komórkowe

Projekt 1

Trzystanowy automat komórkowy von Neumann'a

Piotr Moszkowicz

W ramach pierwszego projektu moim zadaniem była implementacja trzystanowego automatu komórkowego von Neumann'a w technologiach webowych. W tym celu wykorzystany został język TypeScript („nakładka” na język JavaScript pozwalająca na statyczne typowanie w momencie kompilacji) oraz framework React. Do wizualizacji danych (tj. rysowania wykresów) wykorzystano bibliotekę react-vis utworzoną przez inżynierów z firmy Uber. Wchodząc na stronę aplikacji ukazuje nam się ekran podzielony na dwie części – z lewej możemy zarządzać symulacją – mamy możliwość ustawienia reguł symulacji, wystartowanie / zastopowanie symulacji oraz zresetowanie. Natomiast po prawej stronie znajduje się wykres dystrybucji danych komórek oraz pod nim ostatnie 10 kroków symulacji. Każdy kolejny krok symulacji wykonuje się co 1 sekundę, natomiast ilość komórek to 120.

Simulation settings

Start Reset

000: 0: 1: 2:

001: 0: 1: 2:

002: 0: 1: 2:

010: 0: 1: 2:

011: 0: 1: 2:

012: 0: 1: 2:

020: 0: 1: 2:

021: 0: 1: 2:

022: 0: 1: 2:

100: 0: 1: 2:

101: 0: 1: 2:

102: 0: 1: 2:

110: 0: 1: 2:

111: 0: 1: 2:

112: 0: 1: 2:

120: 0: 1: 2:

121: 0: 1: 2:

122: 0: 1: 2:

200: 0: 1: 2:

201: 0: 1: 2:

202: 0: 1: 2:

210: 0: 1: 2:

211: 0: 1: 2:

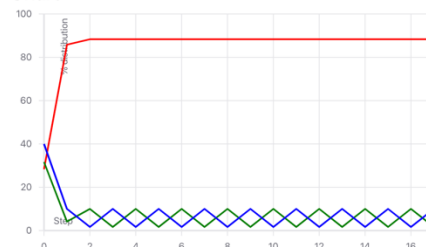
212: 0: 1: 2:

220: 0: 1: 2:

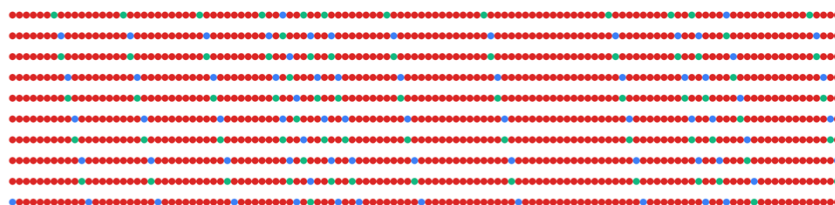
221: 0: 1: 2:

222: 0: 1: 2:

Chart



Simulation



Szukając ciekawych reguł dość szybko natrafiłem na regułę, która pozwala na symetryczne wyniki zielonych oraz niebieskich komórek. Na wykresie widać „wymianę” stanu pomiędzy nimi.

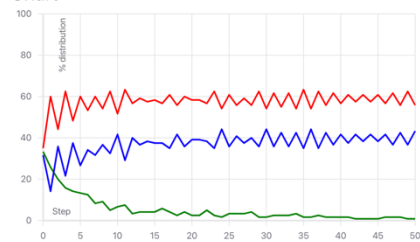
Simulation settings

Start

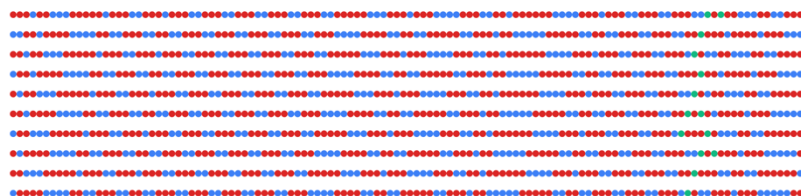
Reset

000: 0: ☐ 1: ☐ 2: ☒
001: 0: ☐ 1: ☒ 2: ☐
002: 0: ☒ 1: ☐ 2: ☐
010: 0: ☒ 1: ☐ 2: ☐
011: 0: ☒ 1: ☐ 2: ☐
012: 0: ☐ 1: ☒ 2: ☐
020: 0: ☒ 1: ☐ 2: ☐
021: 0: ☒ 1: ☐ 2: ☐
022: 0: ☒ 1: ☐ 2: ☐
100: 0: ☐ 1: ☐ 2: ☒
101: 0: ☒ 1: ☐ 2: ☐
102: 0: ☐ 1: ☒ 2: ☐
110: 0: ☐ 1: ☐ 2: ☐
111: 0: ☒ 1: ☐ 2: ☐
112: 0: ☐ 1: ☐ 2: ☐
120: 0: ☐ 1: ☐ 2: ☐
121: 0: ☐ 1: ☒ 2: ☐
122: 0: ☐ 1: ☐ 2: ☐
200: 0: ☐ 1: ☐ 2: ☒
201: 0: ☒ 1: ☐ 2: ☐
202: 0: ☐ 1: ☒ 2: ☐
210: 0: ☒ 1: ☐ 2: ☐
211: 0: ☒ 1: ☐ 2: ☐
212: 0: ☐ 1: ☐ 2: ☒
220: 0: ☒ 1: ☐ 2: ☐
221: 0: ☐ 1: ☒ 2: ☐
222: 0: ☒ 1: ☐ 2: ☐

Chart



Simulation



Kolejną interesującą symulacją jest ta pokazana na powyższym rzucie ekranu. Po chwili można dostrzec ustabilizowanie się komórek czerwonych i niebieskich (ich zależność jest liniowa), mimo, iż komórki zielone dalej istnieją.