

# 1D celulárny automat

Jedno-dimenzionálny celulárny automat pozostáva z vektoru buniek a pravidiel ktoré určujú, ako sa bunky zmenia v ďalšom kroku. Bunky nadobúdajú dva možné stavy. 0-mŕtva bunka, 1-živá bunka.

## Pravidlo 110

Každá bunka sa vyvíja podľa svojho stavu a stavu dvoch susedných buniek. Napríklad bunka so stavom 1, ktorá je obklopená dvoma ďalšími bunkami so stavmi 1, zmení v ďalšom kroku svoj stav na 0. Prvá a posledná bunka budú mať vždy stav 1.

## Zoznam pravidiel

111	110	101	100	011	010	001	000
↓	↓	↓	↓	↓	↓	↓	↓
<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>

### Príklad výpočtu pre stav 11101

1. nastavíme prvú a poslednú bunku na 1

$$\begin{array}{cccccc} & & & & & \\ & & & & & \\ & & & & & \\ \hline & 1 & 1 & 1 & 0 & 1 \\ & & & & & \\ & & & & & \\ & & & & & \\ \hline & \mathbf{1} & \_ & \_ & \_ & \mathbf{1} \\ & & & & & \\ & & & & & \\ & & & & & \\ \hline \end{array}$$

2. vypočítame nový stav pre bunku na pozícii 1 (číslujeme od 0). Použijeme pravidlo pre 111.

$$\begin{array}{cccccc} & & & & & \\ & & & & & \\ & & & & & \\ \hline & 1 & 1 & 1 & 0 & 1 \\ & & & & & \\ & & & & & \\ & & & & & \\ \hline & 1 & \mathbf{0} & \_ & \_ & 1 \\ & & & & & \\ & & & & & \\ & & & & & \\ \hline \end{array}$$

3. vypočítame nový stav pre bunku na pozícii 2. Použijeme pravidlo pre 110.

$$\begin{array}{cccccc} & & & & & \\ & & & & & \\ & & & & & \\ \hline & 1 & 1 & 1 & 0 & 1 \\ & & & & & \\ & & & & & \\ & & & & & \\ \hline & 1 & 0 & \mathbf{1} & \_ & 1 \\ & & & & & \\ & & & & & \\ & & & & & \\ \hline \end{array}$$

4. vypočítame nový stav pre bunku na pozícii 3. Použijeme pravidlo pre 101.

$$\begin{array}{cccccc} & & & & & \\ & & & & & \\ & & & & & \\ \hline & 1 & 1 & 1 & 0 & 1 \\ & & & & & \\ & & & & & \\ & & & & & \\ \hline & 1 & 0 & 1 & \mathbf{1} & 1 \\ & & & & & \\ & & & & & \\ & & & & & \\ \hline \end{array}$$

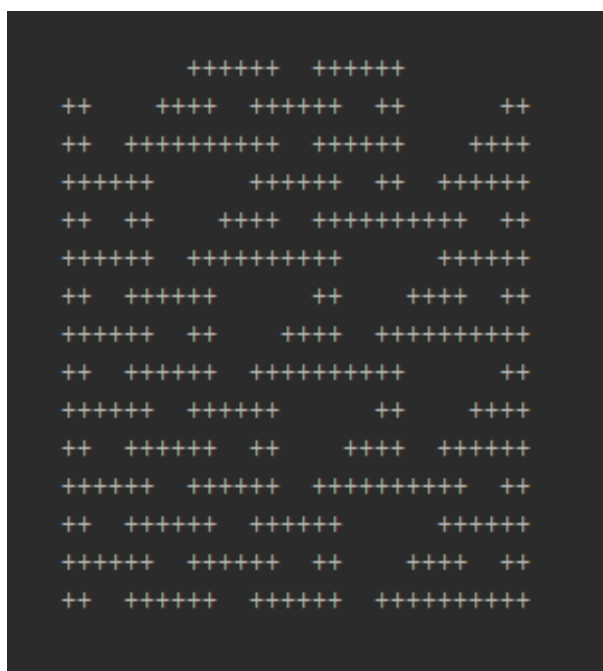
- Napíšte funkciu, ktorá vykreslí stav na príkazový riadok. Stav 0 vykreslite ako dve medzery a stav 1 vykreslite ako 2x plus ++. Napríklad stav [1, 0, 1] sa má vykresliť na ++ ++.

```
def print_state(cells):  
    # TODO: implement
```

- Napíšte funkciu, ktorá bude brať počiatočný stav automatu a počet krokov a vráti list stavov automatu v krokoch 0 až  $n$  (počiatočný stav, stav automatu v prvom kroku, v druhom kroku, ...).

```
def cellular_automaton(cells, n):  
    # TODO: implement automaton
```

- Vyskúšajte vykresliť prvých 15 krokov pre počiatočný stav [0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0]. Mal by sa vám vypísať nasledujúci výstup.



Obr. 1: Očakávaný výsledok