# Obliczenia inspirowane Naturą

## Wykład 01 – Od maszyn Turinga do automatów komórkowych

Jarosław Miszczak

IITiS PAN Gliwice

03/03/2016

# Czego dowiedzieliśmy się na poprzednim wykładzie

1. ...
2. ...
3. ...

# Maszyny Turinga
## Maszyny Turinga

Maszyna Turinga dysponuje bardzo skromnymi środkami obliczeniowymi...

- Wszystkie operacje maszyny wykonywane są na taśmie, na której znajdują się symbole z pewnego skończonego zbioru $\Sigma$.
- Maszyna w jednym kroku może pobrać za pomocą czytnika jeden symbol $x \in \Sigma$ z taśmy, wykonać (lub nie) ruch nad taśmą i zmienić stan wewnętrzny.

...lecz pomimo to potrafi wykonać wiele skomplikowanych zadań.

# Maszyny Turinga
## Deterministyczna maszyna Turinga

### Definicja

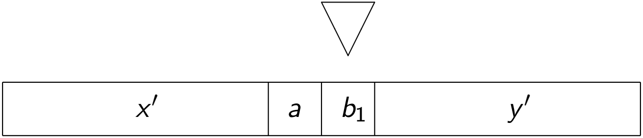
*Deterministyczną maszyną Turinga nazywamy czwórkę uporządkowaną* $M = (K, \Sigma, \delta, q_0)$ *złożoną z:*

- *skończonego zbioru* $K$ *stanów zawierającego stan początkowy* $q_0 \in K$
- *skończonego zbioru* $\Sigma$ *symboli (alfabetu) zawierającego symbol pusty* $\sqcup$ *i symbol końcowy* $\triangleleft$.
- *Funkcji* $\delta\colon K \times \Sigma \to (K \cup \{\text{„stop”}, \text{„tak”}, \text{„nie”}\}) \times \Sigma \times \{\leftarrow, -, \rightarrow\}$

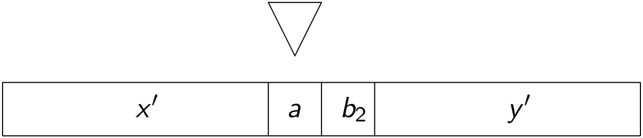
*gdzie* $\leftarrow, \rightarrow, -$ *oznaczają odpowiednio ruch w prawo, w lewo i pozostanie na miejscu.*

# Maszyny Turinga

Przykład: funkcja przejścia $\delta(q_i, b_1) = (q_2, b_2, -1)$




(a) Konfiguracja $(q_i, x'a, b_1 y')$




(b) Konfiguracja $(q_j, x', ab_2 y')$

# Maszyny Turinga
## Deterministyczna maszyna Turinga

- Najprostszym rozszerzeniem tego modelu jest dodanie możliwości operowania na kilku ciągach danych – otrzymujemy wówczas maszynę Turinga z wieloma taśmami, którą trochę łatwiej się *programuje*.
- Najważniejszą wersją tego modelu jest *niedeterministyczna* maszyna Turinga.

# Maszyny Turinga
## Niedeterministyczna maszyna Turinga

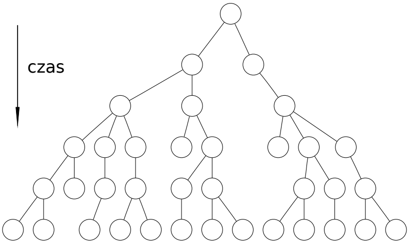
### Definicja

*Niedeterministyczną maszyną Turinga nazywamy czwórkę uporządkowaną* $M = (K, \Sigma, \delta, q_0)$ *złożoną z:*

- *skończonego zbioru* $K$ *stanów zawierającego stan początkowy* $q_0 \in K$
- *skończonego zbioru* $\Sigma$ *symboli (alfabetu) zawierającego symbol pusty* $\sqcup$ *i symbol końcowy* $\triangleleft$.
- *Relacji* $\delta \subset K \times \Sigma \times (K \cup \{\text{„stop”}, \text{„tak”}, \text{„nie”}\}) \times \Sigma \times \{\leftarrow, -, \rightarrow\}$

# Maszyny Turinga

## Jak działa niedeterministyczna maszyna Turinga?




C.H. Papadimitriou, *Złożoność obliczeniowa*, WNT, 2007.

Jak to interpretować?

1. Maszyna wybiera jedną z dróg a nastepnie zachowuje się tak jak maszyna deterministyczna.
2. Wiele maszyn wykonuje program równocześnie.

Ale każda z tych interpretacji wskazuje że niedeteministyczna maszyna Turinga nie jest rozsądnym modelem obliczeń.

# Maszyny Turinga
## Klasy złożoności

- **P** (ang. *polynomial time*) – wszystkie problemy dla których istnieje deterministyczna maszyna Turinga kończąca swoje działanie w czasie wielomianowym względem wielkości problemu.
- **NP** (ang. *nondeterministic polynomial time*) – wszystkie problemy dla których istnieje niedeterministyczna maszyna Turinga kończąca swoje działanie w stanie „tak” w czasie wielomianowym względem wielkości problemu.

**Hipoteza**

**P**$\neq$**NP**

# Maszyny Turinga
## Klasyfikacja języków

Ograniczając możliwości maszyny Turinga możemy wprowadzić hierarchię języków.

- języki **rekurencyjnie przeliczalne** – języki dla których istnieje akceptująca je maszyna Turinga (nie musi się zatrzymywać!);
- języki **rekurencyjne** – języki dla których istnieje rozpoznająca je maszyna Turinga;
- języki **bezkontekstowe** – języki dla których istnieje rozpoznający je (niedeterministyczny) automat ze stosem ($\equiv$ gramatyki bezkontekstowe $\equiv$ Backus-Naur Form);
- języki **regularne** – języki dla których istnieje rozpoznający jest automat skończony ($\equiv$ wyrażenia regularne – sed, gawk, pcre);

J.E.F. Friedl, *Wyrażenia regularne*, Helion, Gliwice, 2001.

# Inne warianty

Maszyna Turinga jest w dużej mierze określona przez określenie relacji przejścia. Ważne warianty to:

- **obliczenia odwracalne**, które pozwalają na:
  - niski pobór energii;
  - możliwość cofnięcia się do stanu przed wystąpieniem błędu (*np.* przy programowaniu robotów);
- **obliczenia kwantowe**, które pozwalają na
  - wykorzystanie superpozycji (czyli równoczesnego wykonywania operacji na wielu danych);
  - fizyczne operowanie na nanostrukturach (pojedyńczych atomach lub dobrze zdefiniowanych grupach atomów);
  - wykorzystanie pojedyńczych fotonów w kryptografii.

# Automaty komórkowe

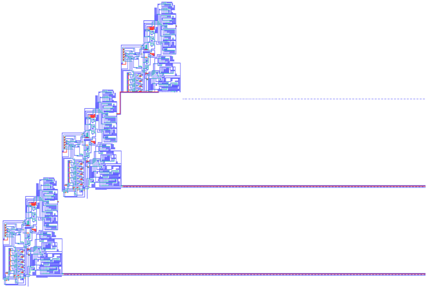
## Krótka historia

- Automaty komórkowe to jeden z pierwszych i najpopularniejszych modeli obliczeniowych inspirowanych zjawiskami obserwowanymi w przyrodzie.
- Jest to model układów dynamicznych w którym czas i przestrzeń są dyskretne.
- Model automatów komórkowych został wymyślony przez Johna von Neumana i Stanisława Ulama w latach czterdziestych XX wieku. Jednak stał się on popularny dopiero w latach '70 dzięki grze *Life* wymyślonej przez Johna Conwaya.

# Automaty komórkowe

## Uniwersalny konstruktor

John von Neumana opracował w latach '40 automat nazywany **uniwersalnym konstruktorem**.

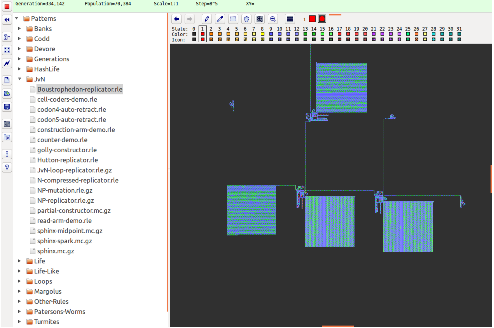


(https://en.wikipedia.org/wiki/Von_Neumann_universal_constructor)

- maszyna najpierw powiela taśmę z instrukcją
- powielona taśma jest programem do budowy nowej maszyny.

# Automaty komórkowe
## Uniwersalny konstruktor

Replikator z dwuwymiarową taśmą.



# Automaty komórkowe

## Znaczenie

Dlaczego automaty komórkowe są ciekawe?

- Jest to prosty model pozwalający na odtworzenie wielu struktur (fraktale, proce urodzin i śmierci).
- Mają zdolność do samopowielania i mogą służyć za model ewolucji.
- Jest to model obliczeń równoważny uniwersalnej maszynie Turinga – wiele problemów ma swoje odpowiedniki (np. problem stopu i problem rajskich ogrodów).