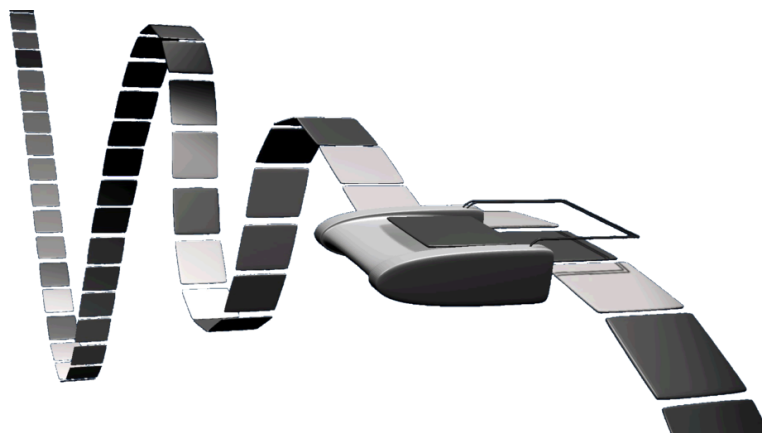


Wojciech Kordecki

Maszyna Turinga

Materiały dodatkowe do przedmiotu
Architektura komputerów



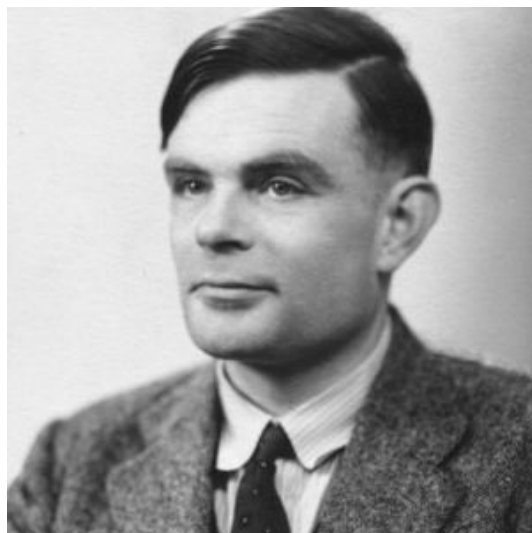
Collegium Witelona Uczelnia Państwowa
Legnica 2024

Na prawach rękopisu

Skład komputerowy w systemie L^AT_EX wykonał autor.

Rysunek na stronie tytułowej: Wikipedia

https://upload.wikimedia.org/wikipedia/commons/b/b7/Turing_Machine.png



Alan Mathison Turing (ur. 23 czerwca 1912 w Londynie, zm. 7 czerwca 1954 w Wilmslow) – angielski matematyk, kryptolog, twórca pojęcia maszyny Turinga i jeden z twórców informatyki.

Spis treści

Wstęp	4
1 Jednotaśmowa maszyna Turinga	4
1.1 Schemat maszyny Turinga	4
1.2 Definicja Maszyny Turinga	5
2 Algorytmy	7

Wstęp

Maszyna Turinga jest abstrakcyjnym urządzeniem wymyślonym przez Alana Turinga w 1936 roku. Pomimo swojej prostoty, maszyna może symulować dowolny komputer, niezależnie od tego, jak jest skomplikowany!

Turing, A.M., 1936-7, “On Computable Numbers, With an Application to the Entscheidungsproblem,” *Proceedings of the London Mathematical Society*, s2-42: 230–265; correction *ibid.*, s2-43 (1936): 544–546 (1937).

Papadimiriou w książce [3] pisze:

Niesamowite, jak niewiele trzeba, by mieć wszystko! Jako język programowania maszyna Turinga dysponuje jedną, dość prymitywną strukturą danych: ciągiem symboli. Dostępne są operacje umożliwiające programowi przesuwanie kursora po ciągu w prawo i w lewo w zależności od wartości bieżącego symbolu. Tak czy inaczej, jest to wyjątkowo ubogi i prymitywny język. Jednak (...) można w nim wyrazić dowolny algorytm i symulować dowolny język programowania.

1 Jednotaśmowa maszyna Turinga

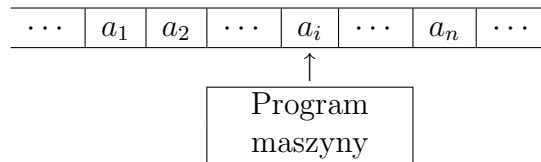
1.1 Schemat maszyny Turinga

Maszyna Turinga składa się z następujących elementów:

- skończonego alfabetu symboli;
- skończonego zbioru stanów;
- nieskończonej taśmy z zaznaczonymi kwadratami, z których każdy może zawierać pojedynczy symbol;
- ruchomej głowicy odczytująco–zapisującej, która może przesuwać się wzdłuż taśmy o jeden kwadrat na raz.
- diagramu przejść między stanami, zawierającego instrukcje, które powodują, że zmiany następują przy każdym zatrzymaniu się.

W zależności od zaobserwowanego przez głowicę symbolu taśmy oraz stanu sterowania skończonego, maszyna Turinga w pojedynczym ruchu:

- zmienia stan,
- wpisuje symbol w obserwowanej komórce taśmy, zastępując symbol tam wpisany,
- przesuwa głowicę \uparrow o jedną komórkę w prawo lub w lewo.



1.2 Definicja Maszyny Turinga

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, B, F \rangle$$

gdzie:

- Q – skończony zbiór stanów,
- $q_0 \in Q$ – stan początkowy,
- F – zbiór stanów końcowych,
- Γ – skończony zbiór dopuszczalnych symboli taśmowych,
- B – symbol pusty należący do Σ ,
- Σ – podzbiór Γ nie zawierający B , zwany zbiorem symboli wejściowych,
- δ – funkcja następnego ruchu, odwzorowanie:

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, -\}.$$

Ruchy głowicy:

- L oznacza ruch w lewo

- R oznacza ruch w prawo,
- $-$ oznacza brak ruchu

Każda z komórek (pól) może znajdować się w jednym ze stanów. Maszyna zawsze jest ustawiona nad jednym polem i znajduje się w jednym ze stanów. Zależnie od kombinacji stanu maszyny i pola, maszyna zapisuje nowy symbol w polu, zmienia stan, a następnie może przesunąć się o jedno pole w prawo lub w lewo. Taka operacja nazywana jest rozkazem. Maszyna Turinga jest sterowana listą zawierającą dowolną liczbę takich rozkazów. Lista rozkazów maszyny Turinga może być traktowana jako jej program.

Stany maszyny można przedstawić jako wierzchołki grafu. Od wierzchołka do wierzchołka prowadzą łuki grafu, którym przypisane są etykiety postaci $(a/b, D)$.

Kiedy maszyna Turinga znajdzie się w stanie q i odczytywanym symbolem będzie $a \in \Gamma$ to nastąpi wpisanie w to miejsce $b \in \Gamma$ i przesunięcie o jedno pole w kierunku D , gdzie $D = L$ lub $D = P$. Stan do którego nie prowadzą żadne łuki, to *stan początkowy*, a stan z którego nie wychodzą żadne łuki, to *stan końcowy*. Jednego węzła może wychodzić tylko jeden łuk z tym samym symbolem a .

Każda komórka tabeli zawiera w sobie instrukcje dla maszyny Turinga postaci

$$\left[\begin{array}{c} q \\ s, D \end{array} \right],$$

gdzie $q \in Q$, $s \in \Gamma$, $D \in \{L, R, -\}$.

Przykład.

	q_0	q_1	q_2	q_3	\dots
Φ	q_0 Φ, R	q_0 a, R	q_3 c, R	\dots	
a	q_1 b, L	q_4 a, L	\dots		
b	q_5 Φ, R	q_0 c, L	\dots		
c	q_7 a, R	q_2 b, L	\dots		

Dalsza studia – Klasyczna książka Aho, Hopcroft, Ullman [1] oraz Papadimitriou [3].

2 Algorytmy

Przykład. Palindrom – ciąg symboli, (napis) (a_1, a_2, \dots, a_n) taki, że $a_k = a_{n-k+1}$.
czyli

$$(a_1, a_2, \dots, a_{n-1}, a_n) = (a_n, a_{n-1}, a_{n-1}, \dots, a_1).$$

Madam, I'm Adam – palindrom, po pominięciu spacji i znaków przestankowych.

Zakładamy, że napis składa się tylko z liter a i b . Sprawdzamy, czy aba jest palindromem. Stanami końcowymi będą N – słowo nie jest palindromem oraz Y – słowo jest palindromem.

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$
- $q_0 = q_0$
- $F = \{q_6, q_7\}$
- $\Gamma = \{\Phi, a, b\}$
- $B = \Phi$
- $\Sigma = \{a, b\}$

Taśma:

Φ	a	b	a	Φ
--------	-----	-----	-----	--------

Tabela 1: Tabela stanów

	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7
Φ	$\begin{matrix} q_6 \\ -, - \end{matrix}$	$\begin{matrix} q_3 \\ \Phi, L \end{matrix}$	$\begin{matrix} q_4 \\ \Phi, L \end{matrix}$	$\begin{matrix} q_6 \\ -, - \end{matrix}$	$\begin{matrix} q_6 \\ -, - \end{matrix}$	$\begin{matrix} q_0 \\ \Phi, R \end{matrix}$	Y	N
a	$\begin{matrix} q_1 \\ \Phi, R \end{matrix}$	$\begin{matrix} q_1 \\ a, R \end{matrix}$	$\begin{matrix} q_2 \\ a, R \end{matrix}$	$\begin{matrix} q_5 \\ \Phi, L \end{matrix}$	$\begin{matrix} q_7 \\ -, - \end{matrix}$	$\begin{matrix} q_5 \\ a, L \end{matrix}$		
b	$\begin{matrix} q_2 \\ \Phi, R \end{matrix}$	$\begin{matrix} q_1 \\ b, R \end{matrix}$	$\begin{matrix} q_2 \\ b, R \end{matrix}$	$\begin{matrix} q_7 \\ -, - \end{matrix}$	$\begin{matrix} q_5 \\ \Phi, L \end{matrix}$	$\begin{matrix} q_5 \\ b, L \end{matrix}$		

W każdym kroku głowica odczytuje symbol z pola i wpisuje nowy, wykonuje ruch określony przez odczytany symbol i stan w dotychczasowy, po czym przechodzi do nowego stanu określonego przez tabelę stanów.

$q_0 : \Phi \underline{a} b a \Phi$
 $q_1 : \Phi \Phi \underline{b} a \Phi$
 $q_1 : \Phi \Phi \underline{b} a \Phi$
 $q_1 : \Phi \Phi \underline{b} a \Phi$
 $q_3 : \Phi \Phi \underline{b} a \Phi$
 $q_5 : \Phi \Phi \underline{b} \Phi \Phi$
 $q_5 : \Phi \underline{\Phi} b \Phi \Phi$
 $q_0 : \Phi \Phi \underline{b} \Phi \Phi$
 $q_2 : \Phi \Phi \Phi \underline{\Phi} \Phi$
 $q_4 : \Phi \Phi \Phi \underline{\Phi} \Phi$
 $q_6 : \Phi \Phi \Phi \Phi \quad Y$

Sprawdzamy teraz, czy abb jest palindromem. Tak jak poprzednio:

$q_0 : \Phi \underline{a} b b \Phi$
 $q_1 : \Phi \Phi \underline{b} b \Phi$
 $q_1 : \Phi \Phi \underline{b} b \Phi$
 $q_1 : \Phi \Phi \underline{b} b \Phi$
 $q_3 : \Phi \Phi \underline{b} b \Phi$
 $q_7 : \Phi \Phi \underline{b} b \Phi \quad N$

Przykład. Dodanie jedynki do liczby zapisanej na taśmie w postaci binarnej – symbolami są cyfry 0 i 1. Przykład jest zaczerpnięty z książki Brookshear, Brylow [2], rozdział 12.2, str. 663.

Stany: START, ADD, CARRY, OVERFLOW, RETURN, HALT.

Alfabet: 0,1.

Symbol pusty: Φ .

Stan początkowy: START.

Stan końcowy: HALT.

Stan początkowy taśmy:

$\Phi 101 \Phi$

Liczba 5, zwiększamy ją o 1.

Stan	Zaw.	Zapis	Kier.	Nowy
START	Φ	Φ	P	ADD
ADD	0	1	L	RETURN
ADD	1	0	P	CARRY
ADD	Φ	Φ	L	HALT
CARRY	0	1	L	RETURN
CARRY	1	0	P	CARRY
CARRY	Φ	1	P	OVERFLOW
OVERFLOW	(?)	Φ	L	RETURN
RETURN	0	0	L	RETURN
RETURN	1	1	L	RETURN
RETURN	Φ	Φ	-	HALT

Literatura

Źródła internetowe

https://commons.wikimedia.org/wiki/File:Turing_Machine.png?uselang=pl
<http://ais.informatik.uni-freiburg.de/tursi/>
<http://swizec.com/blog/a-turing-machine-in-133-bytes-of-javascript/swizec/3069>
<http://introc.s.princeton.edu/java/74turing/>
<http://mathworld.wolfram.com/TuringMachine.html>
<http://www.wolframalpha.com/input/?i=turing+machine>
<http://www.mind.ilstu.edu/curriculum/modOverview.php?modGUI=240>
<http://www.cs.nuim.ie/~tomn/turing/>
<http://plato.stanford.edu/entries/turing-machine/>
<http://aturingmachine.com/>
<http://www.britannica.com/topic/Turing-machine>
https://en.wikipedia.org/wiki/Turing_machine_examples
<https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/turing-machine/>
<http://www.montefiore.ulg.ac.be/~pw/cours/psfiles/calc-chap5.pdf>
<http://www.cs.ucc.ie/~dgb/courses/toc/handout39.pdf>
<http://web.cecs.pdx.edu/~sheard/course/CS311/Fall2012/ppt/>

TuringMachines.pdf

<http://mathoverflow.net/questions/11540/>

[what-are-the-most-attractive-turing-undecidable-problems-in-mathematics](http://mathoverflow.net/questions/11540/what-are-the-most-attractive-turing-undecidable-problems-in-mathematics)

[https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/](https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/turing-machine/one.html)

[turing-machine/one.html](https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/turing-machine/one.html)

Książki

- [1] A. V. Aho, J. E. Hopcroft, J. D. Ullman. *Projektowanie i analiza algorytmów komputerowych*. Helion, Gliwice, 2003.
- [2] J. G. Brookshear, D. Brylow. *Informatyka w ogólnym zarysie*. PWN, Warszawa, 2022.
- [3] C. H. Papadimitriou. *Złożoność obliczeniowa*. WNT, Warszawa, 2002.