

Katarzyna GAWRYŚ¹

¹Katedra Informatyki Przemysłowej, Politechnika Śląska, ul. Krasińskiego 8, 44-019 Katowice

Maszyna Turinga - podstawy działania i programowania

Streszczenie. Maszyna Turinga jest opracowanym przez Alana Turinga ideowym modelem programowania. Ten abstrakcyjny model urządzenia służył do zapisu i wykonania algorytmów. Niniejszy artykuł opisuje budowę i sposób działania maszyny Turinga oraz zasady zapisu algorytmów w postaci tabeli przejść. W artykule umieszczono przykład użycia symulatora maszyny Turinga do rozwiązania przykładowego zadania. Analiza zamieszczonego przykładu, pozwoli odbiorcy, na przyswojenie sposobu szukania rozwiązania problemu, dla ideowego modelu komputera, jakim jest maszyna Turinga.

Słowa kluczowe: algorytm, maszyna Turinga, tabela przejść, diagram stanów, rozkaz, zdarzenie.

1. Wstęp

Wiek XIX i XX to czasy pytań o to dla jakich problemów istnieją algorytmy i czy wszystkie twierdzenia można udowodnić lub obalić [2]. Teoria złożoności obliczeniowej, która zajmuje się analizą algorytmów, dzieli problemy na algorytmiczne i nie. Problemy algorytmiczne to takie, które mogą być rozwiązane za pomocą odpowiednich algorytmów komputerowych tzn. można napisać program komputerowy, który w skończonym czasie da poprawną odpowiedź dla dowolnych poprawnych danych wejściowych przy założeniu dostępu do nieograniczonych zasobów pamięciowych. W celu uniezależnienia się od typu komputera, służącego do realizowania obliczeń, należy posłużyć się ogólnym abstrakcyjnym modelem obliczeń.

Maszyna Turinga to stworzony w 1936 roku przez Alana Turinga abstrakcyjny model komputera - automat abstrakcyjny zdolny do realizacji zapisanego w pewien formalny sposób algorytmu [1]. Model ten pozwala na zapis algorytmów i służy (głównie logikom) do określenia, jakie problemy są możliwe do algorytmicznego rozwiązania [2]. W teorii złożoności obliczeniowej Maszyna Turinga jest używana do udowadniania nierozstrzygalności różnych problemów - problem jest rozwiązywalny na komputerze, jeśli da się zdefiniować rozwiązującą go maszynę Turinga. Wnioski teorii złożoności obliczeniowej, mówiące o obliczeniach na maszynach Turinga, pozostają w ścisłym związku z rzeczywistymi problemami informatyki i praktycznymi implementacjami algorytmów [4]. Maszynę Turinga przyjmuje się za precyzyjną definicję pojęcia algorytmu [2] - nie ma algorytmu dla takiego problemu, którego nie można rozwiązać za pomocą maszyny Turinga.

2. Zasada działania maszyny Turinga

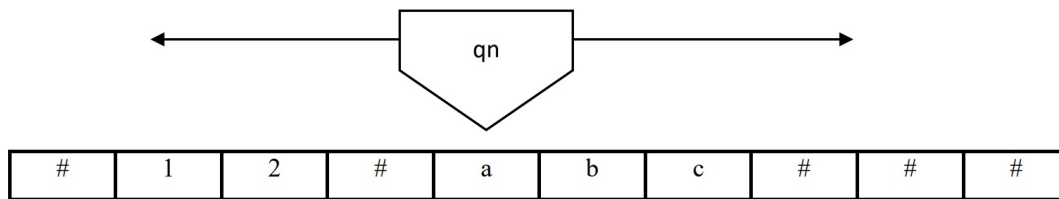
Maszyna Turinga będąca rodzajem automatu skończonego jest ideowym modelem programowania i służy do rozważań teoretycznych. Program zapisuje się w niej w postaci tabeli przejść, równoważnym zapisem jest diagram stanów.

2.1. Automat skończony

Automat skończony to abstrakcyjny, matematyczny, iteracyjny model zachowania systemu dynamicznego, oparty na tablicy dyskretnych przejść między jego kolejnymi stanami. Modelu używamy do zapisu rozwiązania dla problemu, który można podzielić na etapy, realizowane w kolejności sterowanej zdarzeniami lub zmianą stanu wejść.

2.2. Budowa maszyny Turinga

Maszyna składa się z podzielonej na kratki taśmy i głowicy posiadającej możliwość przesuwania się wzdłuż tej taśmy oraz odczytywania i zapisywania symboli w kratkach, nad którymi się znajduje. Budowę maszyny Turinga przedstawiono na rys. 1.



Rysunek 1. Maszyna Turinga – budowa.

Z założenia taśma maszyny Turinga jest nieskończenie długa, co symbolizuje nieograniczone zasoby pamięciowe urządzenia. Na taśmie znajdują się wydzielone pola, zawierające symbole ze zbioru dopuszczalnych symboli lub symbol pusty reprezentowany przez znak #. Głowica maszyny wykonuje zdefiniowane działania – rozkazy. Głowica zawsze znajduje się w jakimś stanie. Będąc w danym stanie głowica sprawdza jaki symbol jest umieszczony bezpośrednio pod nią na taśmie. Na podstawie symbolu, który głowica „widzi” i stanu w jakim jest, podejmuje decyzję:

- jaki symbol umieścić na taśmie,
- w którą stronę się przesunąć (zawsze o jedno pole),
- do jakiego stanu przejść.

2.3. Tablica charakterystyczna

Zasady pisania programu dla maszyny Turinga w postaci tablicy przejść są następujące. Decyzja, którą podejmuje maszyna Turinga stanowi rozkaz, czyli instrukcję, którą należy zapisać w tablicy przejść - na

| Symbole /Stany | Stan {q0} | Stan {q1} | Stan {q2} | Stan {q3} | ... | Stan {qn} |
|-------------------|-----------|-----------|-----------|-----------|-----|-----------|
| # | | | | | | |
| {a} | | q5/b/L | | | | |
| {b} | | | | | | |
| {1} | | | | | | |
| ... | | | | | | |

Rysunek 2. Tablica charakterystyczna - przejść.

przecięciu odpowiedniego wiersza i kolumny. W niniejszym artykule określenia tablica charakterystyczna i tablica przejść są stosowane wymiennie.

Zapis na rys. 2. należy interpretować następująco: jeżeli będąc w stanie q_1 , pod głowicą znajduje się symbol „a” to wykonaj instrukcję: wpisz w miejsce symbolu „a” symbol „b”, przejdź do stanu głowicy q_5 i przesunij głowicę o jedno pole w lewo.

Założenia:

- symbol zapisany w kratce musi należeć do zbioru s_1, s_2, \dots, s_n zwanego alfabetem S maszyny,
- głowica musi znajdować się w jednym ze stanów q_1, q_2, \dots, q_m należącym do zbioru dopuszczalnych stanów głowicy Q ,
- stan głowicy q_j oraz symbol zapisany w kratce nad którą się znajduje, określają jednoznacznie stan maszyny,
- algorytm działania maszyny Turinga zapisany jest w tablicy charakterystycznej, w której dla każdego możliwego stanu maszyny Turinga określonego jako para (s_i, q_j) zdefiniowana jest instrukcja R_{ij} .
- program maszyny Turinga, opisuje funkcja przejścia, która przyporządkowuje uporządkowanej parze (symbol, stan) trójkę (nowy symbol, nowy stan, rodzaj ruchu).

2.4. Instrukcja Maszyny Turinga

Instrukcja maszyny Turinga określa:

1. jaki symbol zapisać w kratce, nad którą znajduje się głowica,
2. jaki ruch wykonać,
3. jak zmienić stan głowicy.

Każdą instrukcję dla Maszyny Turinga można zapisać następująco:

$R_{ij}(s_i, q_j) = (s_k, q_l, r)$, gdzie:

R_{ij} - rozkaz dla stanu maszyny określonego jednoznacznie przez przecięcie i -tego wiersza i j -tej kolumny,

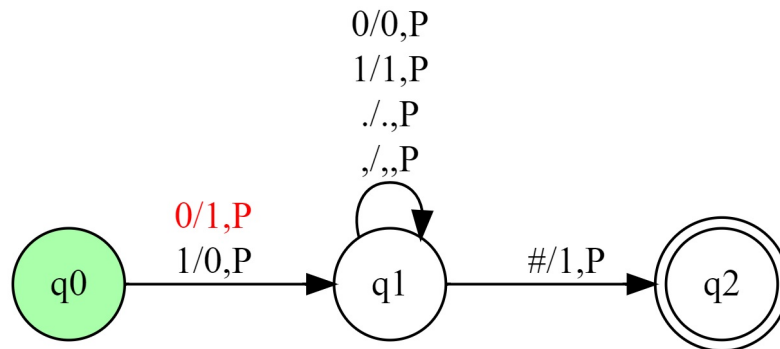
s - symbol należący do alfabetu maszyny Turinga (pierwsza kolumna tabeli),

q - stan głowicy maszyny Turinga (pierwszy wiersz tabeli),

r - ruch głowicy (w lewo - L, w prawo - P, pozostanie w miejscu - -).

2.5. Diagram stanów

Diagram stanów maszyny Turinga jest równoważnym dla tabeli przejść zapisem algorytmu. Ma on postać grafu w którym wierzchołki reprezentowane są przez okręgi z umieszczonym wewnątrz symbolem stanu. Krawędzie grafu łączą wierzchołki w przypadku, gdy w regule sterowania występuje przejście między stanami. Krawędź ma kierunek zgodny z cyklem pracy maszyny. Przejście między stanami jest zależne od aktualnego symbolu na taśmie, dlatego jest etykietowane nowym symbolem do wydrukowania w miejsce aktualnego oraz kierunkiem ruchu głowicy. Przykładowy diagram stanów przedstawia rys. 3.



Rysunek 3. Przykładowy diagram stanów maszyny Turinga.

3. Przykład zadania z rozwiązaniem

W rozdziale zostanie zaprezentowany proces projektowania tablicy charakterystycznej, rozwiązującej przykładowe zadanie przy użyciu opisanej notacji do zapisu algorytmów.

W procesie pisania programu dla maszyny Turinga - czyli zapisu instrukcji w tablicy przejść - czytelnik artykułu może skorzystać z dostępnego w internecie darmowego (freeware) symulatora zaprezentowanego na rys. 34., dostępnego na stronie:

<http://skinderowicz.pl/static/wdi/mt/mt.html>

Prosty symulator Maszyny Turinga

Niezabezpieczona skinderowicz.pl/static/wdi/mt.html

Symulator Maszyny Turinga

Dopuszczalne symbole wejściowe:

Liczba stanów:

| ~ | q0 etykieta | q1 etykieta | q2 etykieta |
|---|----------------|----------------|----------------|
| # | SB | q2/1/P | SK |
| 0 | q1/1/P | q1/0/P | SK |
| 1 | q1/0/P | q1/1/P | SK |
| . | SB | q1/.P | SK |
| , | SB | q1/,P | SK |

Dane wejściowe dla programu MT:

Głowica po prawej stronie

Podgląd taśmy

#####0.1011#####

Nast. krok Cofnij Od początku Nowa maszyna

Pobierz link do maszyny

Pokaż graf

Rysunek 4. Symulator maszyny Turinga.

3.1. Opis symulatora

Pracę nad zapisem algorytmu w postaci tablicy charakterystycznej, zaczynamy od zdefiniowania alfabetu czyli wpisania dopuszczalnych symboli w *pole tekstowe* opisane *etykietą*: **Dopuszczalne symbole wejściowe**.

Następnie wpisujemy, przewidywaną na etapie projektowania algorytmu, liczbę stanów głowicy w *pole tekstowe* pod *etykietą*: **Liczba stanów**. Poniżej znajduje się projektowana tabela przejść z komórkami do których należy wpisywać instrukcje.

Pod *etykietą*: **Dane wejściowe dla programu MT** znajduje się *pole tekstowe*, przeznaczone do wpisania przykładowej sekwencji wejściowej - to przykładowa sekwencja testująca poprawność przetwarzania, wpisanego w tabelę przejść algorytmu.

Pod tabelą znajduje się pole typu *checkbox* opisane *etykietą*: **Głowica po prawej stronie**. Zaznaczenie pola powoduje umieszczenia głowicy, na starcie pracy maszyny, z prawej strony ciągu wejściowego.

Podgląd taśmy zawiera widok taśmy w danym momencie pracy maszyny, realizującej zapisany algorytm i pozwala na bieżąco śledzić poprawność działania algorytmu oraz wpisywać lub modyfikować instrukcje znajdujące się w kratkach tabeli przejść. *Czerwony trójkąt* w sekcji *Podgląd taśmy*, pokazuje aktualną pozycję głowicy. W tabeli przejść *zielonym kolorem* jest wyróżniona instrukcja, która w danym momencie pracy algorytmu jest wykonywana. Po wykonaniu programu sekcja *Podgląd taśmy* prezentuje stan taśmy, będący wynikiem działania algorytmu.

Projektowanie tablicy charakterystycznej, polega na wpisaniu do każdej komórki tablicy charakterystycznej, będącej przecięciem wiersza i kolumny, odpowiedniej instrukcji. Zasady zapisu instrukcji zostały dokładnie odpisane w rozdziale: 2.4 niniejszego artykułu.

Symulator posiada 6 *przycisków*, pierwsze cztery są przydatne w procesie zapisu algorytmu, ostatnie dwa w procesie jego testowania:

- **Nowa maszyna** - przywraca domyślne ustawienia maszyny, zaleca się, aby użyć go na początku pracy nad nowym zadaniem.

W procesie zapisu algorytmu zaleca się używanie trzech przycisków:

- **Następny krok** - przydatny zarówno podczas tworzenia/zapisu i edycji instrukcji programu oraz weryfikacji poprawności działania wpisanego już programu, w procesie testowania pracy algorytmu.
- **Cofnij** - cofa maszynę o jeden krok - pole na taśmie.
- **Od początku** - przywraca głowicę do ustawienia początkowego - w zależności od założeń zadania.

Po wypełnieniu tabeli zaleca się przetestowanie jej odpowiednimi danymi testowymi oraz równoczesne przeanalizowanie diagramu stanów, którego widok jest dostępny po kliknięciu *przycisku* „Pokaż graf”.

- **Pokaż graf** - wyświetla diagram przejść odpowiadający zapisanemu w tablicy przejść programowi. Jest bardzo pomocny w procesie analizy algorytmu. Pozwala zweryfikować poprawność zapisanego programu, jest zapisem równoważnym tablicy charakterystycznej.
- **Pobierz link do maszyny** - po zakończeniu pracy nad zadaniem, po poprawnym zapisie algorytmu i przetestowaniu go, zaleca się zapisanie rozwiązania poprzez użycie tego przycisku. Powoduje on wygenerowanie unikalnego linku, który zawiera rozwiązanie w postaci algorytmu zapisanego w tabeli przejść, pozwala na odtworzenie programu stanowiącego rozwiązanie zadania.

3.2. Proces rozwiązania przykładowego zadania

Treść zadania:

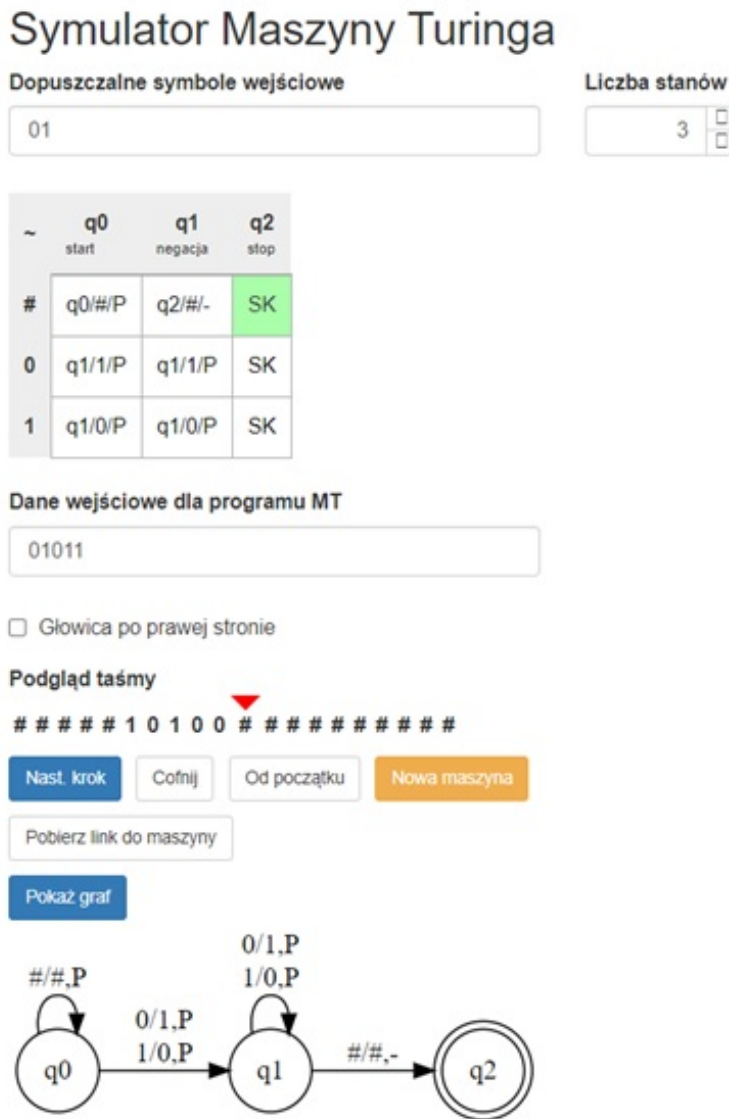
Zaprojektować maszynę Turinga negującą liczbę binarną.

Założenia:

- liczba może być dowolnej długości,
- w stanie początkowym głowica znajduje się z lewej strony taśmy,
- w stanie początkowym głowica poszukuje pierwszego symbolu do analizy, czyli jest ustawiona w pozycji nad symbolem pustym a nie pierwszym z przetwarzanego ciągu binarnego,
- po zakończeniu pracy głowica ustawia się nad pierwszym symbolem pustym, następującym po przetwarzanym ciągu symboli oraz przechodzi do stanu końcowego.

Rozwiązanie.

Algorytm zapisany w tablicy przejść i stanowiący program, musi rozwiązywać zadany problem dla każdej możliwej postaci ciągu wejściowego ze względu na ustawienia zer i jedynek oraz długość ciągu. Negacja liczb binarnych polega na zapisie na taśmie w miejsce symbolu „0” - symbolu „1” oraz w miejsce symbolu „1”- symbolu „0”. Na rys. 5. przedstawiono rozwiązanie zadania w postaci tabeli charakterystycznej oraz diagramu stanów.



Rysunek 5. Tabela przejść i diagram stanów dla rozważanego przykładowego zadania.

Przy założeniu, że głowica jest umieszczona z lewej strony taśmy a algorytm ma skończyć pracę, gdy głowica znajdzie się nad pierwszym symbolem pustym znajdującym się na taśmie po przetworzonym ciągu symboli, będzie się ona poruszała w prawo w ramach każdej instrukcji programu.

Pracę z symulatorem zaczynamy od zdefiniowania alfabetu. Dowolna liczba binarna o dowolnej długości i wartości liczbowej zawiera ciąg „zer” i „jedynek”. Dlatego alfabet składać się będzie z dwóch symboli: „0” i „1”.

Następnie próbujemy oszacować liczbę stanów głowicy wymaganą do rozwiązania problemu, aby wpisać ją do pola symulatora. Ponieważ z założenia algorytm musi poszukiwać pierwszego symbolu do analizy, potrzebujemy trzy stany głowicy: startowy, stan negacji oraz stan końcowy.

W stanie q_0 , jeżeli głowica widzi pod sobą pusty symbol to: pozostaje w tym stanie, pozostawia dany symbol i porusza się w prawo w celu poszukania symbolu „0” lub „1”.

Jeżeli w stanie q_0 głowica „zobaczy” pod sobą symbol inny niż pusty - „0” lub „1”, to przechodzi do stanu negacji czyli q_1 i dokonuje pierwszej negacji tzn. gdy widzi symbol „0” zapisuje w jego miejsce „1”, a gdy pod głowicą znajduje się symbol „1” to zapisuje „0”, przy czym wykonuje ruch w prawo o jedno pole.

Stan q_1 to stan negacji, w którym maszyna zostaje tak długo, dopóki widzi pod sobą symbol „0” lub „1”. Wykonuje negację symbolu, pozostaje w tym stanie i porusza się o 1 pole w prawo.

Ze stanu q_1 głowica przechodzi do stanu q_2 - końcowego, tylko gdy znajdzie się nad symbolem pustym, czyli po zakończeniu przetwarzania ciągu symboli.

Stan końcowy składa się z symboli SK, stan ten oznacza, że maszyna jest w stanie końcowym.

Weryfikacja poprawności.

Poprawność rozwiązania można wykazać wybierając reprezentatywne dla rozwiązywanego problemu przypadki testowe.

Przypadek testowy 1:

| 1 | 0 | 1 | 1 | 0 | 0 |

Oczekiwany ciąg wyjściowy:

| 0 | 1 | 0 | 0 | 1 | 1 |

Przypadek testowy 2:

| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

Oczekiwany ciąg wyjściowy:

| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |

Dodatkowo można sprawdzić, że dla ciągu złożonego z samych zer otrzymamy, zgodnie z oczekiwaniami, sekwencje o tej samej długości złożoną z samych jedynek i odwrotnie - dla ciągu złożonego z samych jedynek otrzymamy same zera.

Analiza diagramu stanów.

Po naciśnięciu przycisku „pokaż graf” na stronie pod tabelą przejść, zostanie wyświetlony odpowiadający jej diagram stanów. Jest to inna - równoważna forma zapisu tego samego algorytmu, zapisanego w tabeli przejść.

Stan głowicy jest reprezentowany przez okrąg z umieszczoną wewnątrz nazwą. Stan końcowy reprezentowany jest przed okrąg w podwójnej obwódce. Stany – okręgi są połączone strzałkami i określają możliwe przejścia między stanami, które zachodzą, gdy wystąpi określone zdarzenie. Strzałki symbolizujące przejścia między stanami są opisane trzema symbolami:

„symbol1”/„symbol2”, „ruch”.

Zapis należy odczytać: przejście między stanami połączonymi strzałką z danym opisem, może mieć miejsce, w wyniku zajścia zdarzenia: „symbol1”, tzn. gdy głowica znajdzie się nad symbolem z alfabetu: „symbol1”. Wtedy w miejsce tego symbolu na taśmie, zostaje zapisany „symbol2”, maszyna wykonuje opisany „ruch” oraz przechodzi do stanu, na który wskazuje grot strzałki.

Strzałka - przejście, która wychodzi z danego stanu i wchodzi również do tego samego stanu oznacza sytuację, w której głowica pozostaje w danym stanie tak długo, dopóki znajduje się nad symbolem: „symbol1”.

Diagram stanów pokazuje najważniejsze cechy zapisanego algorytmu pod kątem zdarzeń powodujących przejścia między stanami i służy weryfikacji jego poprawności.

Stan q_0 to stan początkowy - startowy, w którym zostajemy tak długo, dopóki na taśmie jest symbol pusty. Ze stanu q_0 do q_1 maszyna przechodzi, pod wpływem zajścia zdarzenia, polegającego na napotkaniu symboli „0” lub „1”. Pod wpływem zajścia dowolnego z tych dwóch zdarzeń zapisujemy symbol będący jego negacją i przechodzimy do stanu q_1 . Stan q_1 to stan negacji w którym wykonujemy charakterystyczne dla niego działania, czyli negację wszystkich symboli i przechodzimy, po napotkaniu symbolu pustego, do stanu końcowego.

4. Podsumowanie

Artykuł ten może być materiałem pomocniczym dla studentów pierwszego roku na kierunku Informatyka Przemysłowa lub Informatyka, realizujących przedmiot Podstawy informatyki. Prezentuje związek między algorytmiką a przykładem automatu skończonego jakim jest maszyna Turinga. Przedstawia notację do zapisu algorytmów w postaci tabeli przejść oraz diagramu stanów. Uczy myślenia w kategorii stanów w jakich może znajdować się automat oraz zdarzeń powodujących przejścia między nimi w celu rozwiązania problemu.

Podziękowania

Autor artykułu serdecznie dziękuje recenzentom za trud włożony w recenzję.

Literatura

1. S. Kozielski. *Zbiór zadań z podstaw informatyki*, Wydawnictwo Politechniki Śląskiej, Gliwice 1998.
2. M. Kubale, *Łagodne wprowadzenie do analizy algorytmów*, Wydawnictwo Politechniki Gdańskiej, Gdańsk 2002.
3. S. Węgrzyn, *Wykłady z podstaw informatyki*, Wydawnictwo Politechniki Śląskiej. Gliwice 2003.
4. https://wazniak.mimuw.edu.pl/index.php?title=Złożoność_Obliczeniowa/Wykład_1:_Obliczenia_w_modelu_maszyzny_Turinga