

Obliczalność i złożoność

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

sem. zimowy 2025/26

inf.ug.edu.pl/~amb/oblicz

Andrzej M. Borzyszkowski (Instytut Informatyki) Obliczalność i złożoność sem. zimowy 2025/26 1 / 19

Wykład 00 Maszyna Turinga

Alan Turing

- Alan Turing (1912-1954) – angielski matematyk, logik, informatyk
- kierował kryptoanalizą niemieckiej maszyny szyfrującej Enigma
- zaprojektował architekturę komputerów, t.j. maszyn wykonywujących dowolny algorytm – maszyny Turinga
- pojęcie *obliczalności*
- test Turinga – odróżnienie człowieka od maszyny
- CAPTCHA – *Completely Automated Public Turing test to tell Computers and Humans Apart*

Andrzej M. Borzyszkowski (Instytut Informatyki) Obliczalność i złożoność sem. zimowy 2025/26 3 / 19

Maszyny Turinga

Andrzej M. Borzyszkowski (Instytut Informatyki) Obliczalność i złożoność sem. zimowy 2025/26 2 / 19

Wykład 00 Maszyna Turinga

Maszyna Turinga

- automat z nieskończoną taśmą wejściową
- głowica automatu może poruszać się w obu kierunkach, nie tylko czytać po kolei
- automat może zapisywać litery na taśmie, zarówno nadpisywać istniejące jak i zapisywać miejsca dotychczas puste

Andrzej M. Borzyszkowski (Instytut Informatyki) Obliczalność i złożoność sem. zimowy 2025/26 4 / 19

Maszyna Turinga (deterministyczna)

- $\mathcal{A} = \langle Q, A, \Gamma, B, \delta, q_0, F \rangle$ gdzie:
 - Q – skończony zbiór stanów,
 - $q_0 \in Q$ – stan początkowy,
 - $F \subseteq Q$ – zbiór stanów końcowych (akceptujących),
 - A – alfabet wejściowy
 - $\Gamma \setminus A$ – alfabet roboczy
 - $B \in \Gamma \setminus A$ – symbol *blank* pustego miejsca na taśmie
 - $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$ funkcja częściowa określająca przejście stanów, nadpisywanie liter oraz ruch głowicy czytającej taśmę

Język maszyny Turinga

- konfiguracja jest *akceptująca* jeśli jej stan jest akceptujący
- obliczenie jest akceptujące jeśli jest skończone i ostatnia konfiguracja jest akceptująca
- $L(\mathcal{A}) = \{w \in A^* \mid \mathcal{A} \text{ akceptuje } w\}$
- język L jest *rekurencyjnie przeliczalny* (*recursively enumerable*) jeśli $L = L(\mathcal{A})$ dla pewnej maszyny Turinga $\mathcal{A}$
- $w \in L(\mathcal{A})$ oznacza, że maszyna Turinga zatrzyma swoje obliczenia i ujawni pozytywny fakt
- $w \notin L(\mathcal{A})$
 - może być zauważone po zatrzymaniu się maszyny Turinga w stanie nieakceptującym
 - ale może się okazać, że obliczenia maszyny Turinga ze słowem wejściowym w są nieskończone

Ruchy maszyny Turinga

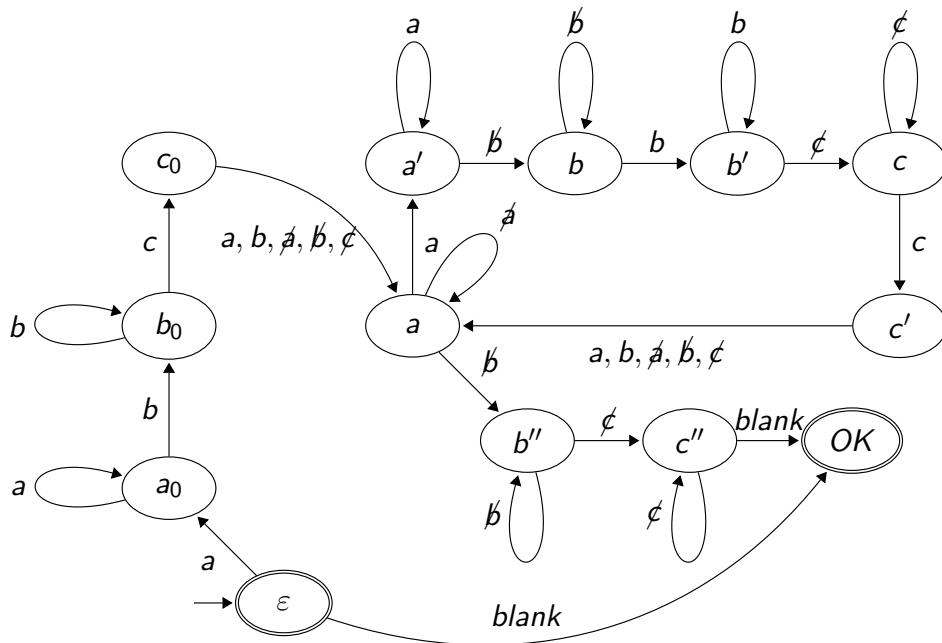
- jeśli $\delta(q, \sigma) = (p, \tau, S)$ to mówimy, że (q, σ, p, τ, S) opisuje jeden ruch automatu
 - w stanie q przeczytał literę σ i przeszedł w stan p , wpisał literę τ w miejscu gdzie stała σ oraz przesunął głowicę czytającą w prawo, lewo lub wcale (S)
- konfiguracja to trójka $uqvw$, $q \in Q$, u jest słowem przed głowicą czytającą, pierwsza litera w jest wskazywana przez głowicę
- ruch (q, σ, p, τ, N) oznacza przejście $uq\sigma v \rightarrow up\tau v$
- ruch (q, σ, p, τ, R) oznacza przejście $uq\sigma v \rightarrow u\tau p v$
- ruch (q, σ, p, τ, L) oznacza przejście $u\mu q\sigma v \rightarrow up\mu\tau v$
- obliczenie – skończony lub nieskończony ciąg konfiguracji
- konfiguracja początkowa – $q_0 w$, $w \in A^*$ (na początku wpisane jest słowo z liter alfabetu wejściowego, potem mogą pojawiać się litery z alfabetu roboczego)

Języki rekurencyjne

- język L jest rekurencyjny/ rozstrzygalny/ obliczalny, jeśli
 - $L = L(\mathcal{A})$ dla pewnej maszyny Turinga
 - obliczenia tej maszyny są skończone dla każdego słowa $w \in A^*$
- tzn. podając słowo w na taśmie wejściowej czekamy na zatrzymanie się obliczeń, by rozstrzygnąć czy $w \in L(\mathcal{A})$ czy też nie
 - do czasu zatrzymania się maszyny musimy czekać na rozstrzygnięcie

Przykład $\{a^n b^n c^n \mid n \geq 0\}$

- język $\{a^n b^n c^n \mid n \geq 0\}$ nie jest bezkontekstowy, tzn. nie jest rozpoznawalny przez automat ze stosem
- jest przekrojem dwóch języków bezkontekstowych, czyli jest rozpoznawalny przez automat z dwoma stosami
- jest rozpoznawalny przez maszynę Turinga
- w stanie początkowym można przeczytać a lub $blank$ (wówczas akceptuje i kończy)
- wpp zamienia a na a przekreślone i przechodzi w stan szukania b
- t.j. mija a do czasu znalezienia b , które przekreśla i szuka c
- j.w., po znalezieniu zamienia na c przekreślone i wraca do początku taśmy i do nowego stanu
- c.d. kolejny slajd

**Przykład** $\{a^n b^n c^n \mid n \geq 0\}$

- błędem będzie nie znalezienie właściwej litery do przekreślenia, maszyna zatrzyma się bez akceptacji
- teraz szuka a , tj. mija a przekreślone, znajduje a , przekreśla i szuka b
- t.j. mija ew. pozostałe a , potem b przekreślone, znajduje b , przekreśla i szuka c
- po znalezieniu wraca do początku i powtarza
- być może nie znajdzie nowego a , czyli wszystkie a są już przekreślone
- po pierwszym b przekreślonym czyta jedynie litery przekreślone, najpierw b , potem c
- jeśli kolejny znak jest $blank$, to akceptuje, każdy inny znak jest nie akceptowany
- już wcześniej mógł być błąd, gdy np. liter b było więcej niż a i zostały nieprzekreślone

Maszyna Turinga wielościeżkowa

- automat z dwoma stosami może rozpoznawać więcej języków niż automat z jednym stosem
- pytanie, czy maszyna Turinga z dwoma ścieżkami ma większą moc rozpoznawania niż z jedną ścieżką
- odpowiedź: **nie**, maszyna w wieloma ścieżkami jest równoważna maszynie z jedną ścieżką
- dw: jeśli przyjąć, że wszystkie ścieżki korzystają z tej samej głowicy czytającej, to w zasadzie możemy uznać, że taśma wejściowa jest jedna – każda komórka jest zestawem komórek poszczególnych taśm
- bardziej skomplikowany jest przypadek, gdy taśm naprawdę jest więcej i każda ma swoją pozycję głowicy

Maszyna Turinga wielościeżkowa c.d.

- dane: maszyna z np. dwoma taśmami i niezależnymi głowicami
- dodajemy dwie nowe taśmy, które mają tylko jeden symbol, mianowicie pozycję głowicy 1. lub 2.
- otrzymujemy maszynę z czterema taśmami, przyjmujemy, że jedyna głowica domyślnie jest na początku
- jeden ruch maszyny z dwoma taśmami:
 - odszukanie pozycji taśmy 1. na dodatkowej taśmie, odczyt symbolu z tej taśmy, powrót
 - to samo dla taśmy 2.
 - decyzja co wpisać na obu taśmach, jak zmienić pozycje obu głowic i jak zmienić stan
 - odszukanie pozycji taśmy 1., wpis nowego symbolu, powrót
 - to samo dla taśmy 2.
 - a potem, na obu taśmach, odszukanie pozycji i ew. ich zmiana

Maszyny Turinga a automaty ze stosem

- automat ze stosem można zaimplementować jako maszynę Turinga
 - na taśmie wejściowej niczego nie zapisujemy, ruch tylko w prawo
 - na drugiej taśmie zapisujemy zawartość stosu, przyda się symbol dna stosu
- maszynę Turinga można zaimplementować jako automat z dwoma stosami
 - na jednym stosie leży zawartość taśmy na prawo od głowicy, na drugim na lewo
 - w ogóle nie korzystamy z taśmy wejściowej dla automatu
 - ruch głowicy czytającej oznacza przenoszenie jednej komórki pomiędzy stosami

Przykłady maszyn wielościeżkowych

- język $\{wcw \mid w \in (a+b)^*\}$
maszyna przepisuje słowo w na drugą taśmę, gdy znajdzie separator c cofa głowicę drugiej taśmy i dalej porównuje oba napisy
- język $\{a^n \mid n \text{ jest potęgą } 2\}$
maszyna czyta kolejne a na pierwszej taśmie, a na drugiej ma zapis binarny n (w kolejności od najmniej znaczących bitów) i dodaje binarnie 1.
na końcu sprawdza, czy n jest potęgą 2
- maszyna Turinga z taśmą obustronnie nieskończoną
drugą taśmą jest lewa część oryginalnej taśmy z odwróconym kierunkiem
przejście głowicy czytającej na lewą stronę oznacza, że przechodzimy do drugiej taśmy

Maszyna Turinga obliczająca

- dotychczasowa definicja maszyny Turinga mówiła o problemie decyzyjnym, czy $w \in L(\mathcal{A})$?
- def.: funkcja $f : A^* \rightarrow \Sigma^*$ jest *obliczalna* jeśli istnieje (deterministyczna) maszyna Turinga $\mathcal{A}$ z dwiema taśmami, wejściową i wyjściową, taka, że
 - jeśli na wejściu znajduje się $w \in A^*$, to maszyna zatrzyma swoje działanie
 - i na wyjściu znajdzie się słowo $f(w)$

Obliczenie $n \cdot m$

- wejście: ciąg bitów przedstawiający n i m
- problem: jak odróżnić obie liczby
 - np. dodatkowa litera w alfabecie wejściowym (w teorii nie ma ograniczeń na liczbę liter alfabetu)
 - albo od razu dopuścić by były dwie odrębne taśmy z danymi wejściowymi
- wyjście: ciąg bitów przedstawiający iloczyn $n \cdot m$
- konstrukcja maszyny
 - przepisujemy m na taśmę pomocniczą
 - w każdym ruchu odejmujemy jedynkę od n i dodajemy m do zawartości taśmy wyjściowej
 - tak długo aż $n == 0$
- dodawanie zawartości dwóch liczb trzeba dokładniej opisać
 - lub posłużyć się inną maszyną, która to realizuje

Rozstrzygalność

- problem logiczny można sprowadzić do pytania, czy pewne zdanie jest prawdziwe
- równoważnie, mamy zapisy zdań prawdziwych i pytamy, czy nasze zdanie jest prawdziwe
- równoważnie, mamy język (zbiór zdań), słowo i pytamy czy $w \in L$
- rozstrzygalność: maszyna zatrzyma się na każdym słowie i da odpowiedź tak/nie
- akceptowanie: maszyn zatrzyma się na słowach, które zaakceptuje, ma prawo nie kończyć obliczeń dla innych
- dla języków rozróżnialiśmy klasy języków rekurencyjnych i jedynie rekurencyjnie przeliczalnych
 - tw. jeśli język i jego dopełnienie są rekurencyjnie przeliczalne, to oba są rekurencyjne

Niedeterministyczne maszyny Turinga

- funkcja przejścia $\delta : Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R, N\})$ jest niedeterministyczna, t.j. definiuje zbiór możliwych wyników
- konfiguracje układają się nie w ciąg obliczeń a w drzewo
- słowo jest zaakceptowane jeśli istnieje ścieżka do liścia z konfiguracją akceptującą
- zakładamy, że stan akceptujący nie dopuszcza dalszych obliczeń
- tw.: niedeterministyczne maszyny Turinga rozpoznają tę samą klasę języków co maszyny deterministyczne
- nie są znane żadne metody fizycznej realizacji maszyn niedeterministycznych
- w szczególności ew. powstanie komputerów kwantowych nie zrealizuje dowolnych maszyn niedeterministycznych