

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 / 20

Wykład 04 Niedeterministyczne maszyny Turinga

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
 - dowód będzie pod koniec wykładu
- nie są znane żadne metody fizycznej realizacji maszyn niedeterministycznych
- w szczególności ew. powstanie komputerów kwantowych nie zrealizuje dowolnych maszyn niedeterministycznych

Andrzej M. Borzyszkowski (Instytut Informatyki) Obliczalność i złożoność sem. zimowy 2025/26 3 / 20

Maszyny Turinga c.d.

Andrzej M. Borzyszkowski (Instytut Informatyki) Obliczalność i złożoność sem. zimowy 2025/26 2 / 20

Wykład 04 Niedeterministyczne maszyny Turinga

Przykład maszyny niedeterministycznej

- wejście: ciąg zer długości n
- problem: pytanie, czy n jest liczbą złożoną
- maszyna niedeterministyczna:
 - wpisuje k zer na taśmie pomocniczą, $1 < k < n$
 - przesuwa głowicę na obu taśmach jednocześnie
 - gdy skończą się zera na drugiej taśmie wraca do jej początku
 - kończy, gdy skończą się zera na pierwszej taśmie
 - rozstrzygnięcie: czy druga taśma też została przeczytana do końca
- to jest oczywiście naiwna implementacja dzielenia z resztą
- ta maszyna ma własność stopu, po zatrzymaniu się wiemy, czy wejście zostało zaakceptowane, czy nie
- maszyna na pewno zatrzyma się po n krokach (plus narzut na początek)

Andrzej M. Borzyszkowski (Instytut Informatyki) Obliczalność i złożoność sem. zimowy 2025/26 4 / 20

Przykład maszyny deterministycznej

- ten sam problem może być rozwiązany przez maszynę deterministyczną poprzez wypróbowanie kolejnych możliwości
 - tzn. zaczynamy od testowania podzielności n przez 2 – dwa zera na drugiej taśmie
 - po zatrzymaniu się testu z negatywnym wynikiem dopisujemy zero na drugiej taśmie
 - i powtarzamy test
 - trzeba się zabezpieczyć przed przypadkiem $k = n$, tzn. gdy do tego dojdzie to maszyna kończy działanie
- ta maszyna deterministyczna skończy działanie najdalej po n^2 krokach

Kodowanie grafów

- wierzchołki grafu można ponumerować, tzn. przyjmujemy $V = \{0, \dots, n-1\}$ dla grafu o n wierzchołkach
- dla wierzchołka $k \in V$ mamy listę jego sąsiadów, zapisywaną $L_k = k : i_1, \dots, i_{j_k}$, gdzie $(k, i_\ell) \in A$
- cały graf zapisujemy jako lista list sąsiedztw tj. $L_0; \dots; L_{n-1}$
- czyli na taśmie wejściowej mamy zapisane ciągi liczb oraz przecinki, dwukropki i średniki, wszystkie te elementy należą do naszego alfabetu
- można sprawdzić czy taki zapis jest rzeczywiście zapisem grafu
- w konkretnych zastosowaniach można stosować bogatsze dane, np. długości krawędzi, czy inne etykiety

Teoria grafów

- graf skierowany $G = (V, A)$ gdzie $A \subseteq \{(v, w) \in V \times V \mid v \neq w\}$
- krawędzie (łuki) mają kierunek „od v do w ” – początek i koniec
- krawędź (v, w) to co innego niż (w, v) , mogą istnieć obie niezależnie
- ścieżką jest ciąg różnych wierzchołków $v_0, \dots, v_n$ takich, że $(v_i, v_{i+1}) \in A$, $i = 0, \dots, n-1$
- cyklem jest ciąg $v_0, \dots, v_n, v_0$ taki, że $v_0, v_1, \dots, v_n$ jest ścieżką i $(v_n, v_0) \in A$
- cykl może być bardzo krótki, v, w, v jeśli $(v, w) \in A$ i $(w, v) \in A$, zaangażowane są tylko dwa wierzchołki
- graf jest silnie spójny jeśli każde dwa wierzchołki łączy ścieżka
- graf jest acykliczny jeśli nie ma cykli
- graf silnie spójny (nie punkt) ma cykle (od v do w i z powrotem)

Inne kodowanie grafów

- graf o n wierzchołkach zapisujemy w macierzy G wymiaru $n \times n$ kodując krawędź $(i, j) \in V \times V$ jako jedynkę w $G[i, j]$
- jeśli zachodzi konieczność etykietowania krawędzi, w komórkach macierzy wpisywane są właśnie etykiety
- grafy nieskierowane możemy traktować jako szczególne grafy skierowane – mają krawędzie pogrupowane w pary, zwrot w jedną i w drugą stronę

Problem teorii grafów

- przykładowy problem grafowy: dany graf (skierowany) G , wierzchołki $v, w \in V$, czy istnieje ścieżka od v do w ?
- problem jest rozstrzygalny jeśli dla każdej instancji problemu maszyna Turinga zatrzyma się i da prawidłową odpowiedź

Niedeterministyczna maszyna Turinga akceptująca problem osiągalności w grafie

- są tylko dwie taśmy, pierwsza z grafem i wierzchołkami u i v
- na drugiej taśmie zaczynamy od wierzchołka u i wpisujemy po kolei (dowolnego) sąsiada ostatniego wierzchołka
- jeśli wpisujemy v , to istnieje ścieżka od u do v
- jeśli istnieje ścieżka od u do v , to istnieje obliczenie akceptujące tę instancję problemu
- jeśli takiej ścieżki nie ma, to żadne obliczenie nie zaakceptuje tej instancji
- jeśli w grafie jest cykl, to istnieje obliczenie nieskończone (w tej wersji nie dbaliśmy o zaznaczanie odwiedzonych wierzchołków)
- od maszyny niedeterministycznej nie oczekujemy własności stopu

Deterministyczna maszyna Turinga rozstrzygająca problem osiągalności w grafie

- maszyna ma trzy taśmy
- taśma pierwsza: zakodowany graf G oraz wierzchołki u i v
- taśma druga: wpisywane będą wierzchołki osiągalne z u , na początku wpisany jest u , będą one sukcesywnie zaznaczane (jako odwiedzone)
- maszyna powtarza cykl, w którym
 - zaznacza nowy wierzchołek w na drugiej taśmie
 - na trzecią taśmę wpisuje wierzchołki połączone z w
 - wierzchołki z trzeciej taśmy wpisuje na drugą, jeśli ich nie było
 - czyści trzecią taśmę
- kończy pracę gdy
 - v znajdzie się na trzeciej taśmie – jest ścieżka
 - lub wszystkie wierzchołki na drugiej taśmie są zaznaczone – nie ma ścieżki

Inne problemy grafowe

- silna spójność: istnieje ścieżka od u do v dla każdych wierzchołków u, v grafu
- istnienie ścieżki/cyklad Hamiltona: tj. ścieżki/cykladu, który odwiedza wszystkie wierzchołki bez powtórzeń
- kolorowanie grafu: ustalona liczba $k > 0$, pytanie, czy można wierzchołki podzielić na k kolorów tak by krawędzie zawsze miały końce o różnych kolorach

Problemy logiki zdaniowej

- dana formuła logiki zdaniowej, tj. złożona ze zdań atomowych połączonych spójnikami logicznymi, np. negacji, implikacji, koniunkcji i dyzjunkcji
- problem: czy istnieje wartościowanie zdań atomowych takie, że formuła ma wartość *True*
- albo czy jest tautologią, tj. ma wartość *True* dla każdego wartościowania
- maszyna niedeterministyczna rozstrzygająca spełnialność formuły
 - na drugą taśmę maszyna wpisuje wszystkie zdania atomowe występujące w formule
 - na trzecią taśmę maszyna wpisuje wartości *True* i *False*
 - następnie cofa się do początku pierwszej taśmy i zamienia zdania atomowe na *True* i *False* zgodnie z taśmami 2 i 3
 - jeszcze raz się cofa i oblicza wartość formuły zapisanej na pierwszej taśmie

Determinizacja c.d.

- istnieje stała d ograniczająca liczbę możliwości wyboru w maszynie niedeterministycznej, $d = \max(|\delta(q, a)|, q \in Q, a \in A)$
- na drugiej taśmie generuje się coraz dłuższe ciągi adresujące drzewo obliczeń M o stopniu rozgałęzienia $\leq d$
- na trzeciej taśmie wykonuje się obliczenia odpowiadające ciągowi zapisanemu na drugiej taśmie
- jeśli maszyna M akceptuje dane wejściowe, to istnieje obliczenie, po którym się zatrzyma i ogłosi akceptację
- czyli maszyna deterministyczna dojdzie do momentu, gdy znajdzie to obliczenie i zaakceptuje dane wejściowe
- przykład ze spełnialnością formuły logiki zdaniowej:
 - maszyna deterministyczna przejrzy 2^n możliwych wartościowań n zdań atomowych zawartych w danej formule

Determinizacja maszyny Turinga

- tw.: dla każdej niedeterministycznej maszyny Turinga M istnieje maszyna deterministyczna MD akceptująca ten sam język – $L(M) = L(MD)$
- dw.: zakładamy, że maszyna ma jedną taśmę, rozpatrujemy graf skierowany
 - wierzchołki: konfiguracje maszyny M , tj. trójki uqw , u, w słowa przed/po głowicy czytającej, $q \in Q_M$ stan maszyny
 - krawędzie: wykonanie jednego kroku obliczeń
- problem: czy istnieje ścieżka od konfiguracji początkowej do jakiejś akceptującej
- NIE jest to dokładnie problem osiągalności w grafie, bo powyższy graf może być nieskończony
- ale można zapisywać i rozważać coraz większe fragmenty grafu w miarę rozwijania się obliczeń

Złożoność obliczeniowa (czasowa)

Notacja asymptotyczna

- rozważamy ciągi liczb naturalnych, równoważnie, funkcji $b, c : \mathbb{N} \rightarrow \mathbb{N}$
- $b_n = O(c_n)$ (b jest „O duże od” c) jeśli
 $\exists C. \exists n_0. \forall n. n > n_0 \Rightarrow b_n < C \cdot c_n$
 „prawie wszystkie b są mniejsze niż c , stała jest nieważna”
- $b_n = o(c_n)$ (b jest „o małe od” c) jeśli
 $\forall C. \exists n_0. \forall n. n > n_0 \Rightarrow b_n < C \cdot c_n$
 „prawie wszystkie b są dowolnie mniejsze niż c ”
- $b_n = \Theta(c_n)$ (b jest „ Θ od” c) jeśli
 $\exists C_1. \exists C_2. \exists n_0. \forall n. n > n_0 \Rightarrow C_1 \cdot c_n < b_n < C_2 \cdot c_n$
 „prawie wszystkie b są mniej więcej równe c ”

Złożoność czasowa

- niech M będzie deterministyczną maszyną Turinga,
- zakładamy, że zatrzymuje się dla każdej konfiguracji początkowej
 - tj. mówimy o rozstrzygalności
- def: maszyna działa w czasie $f(n)$, $f : \mathbb{N} \rightarrow \mathbb{N}$, jeśli dla dowolnego słowa w długości n , zaczynając działanie w konfiguracji początkowej, kończy po najwyżej $f(n)$ krokach
- język L należy do klasy $TIME(f(n))$ jeśli jest akceptowany przez maszynę Turinga działającą w czasie $f(n)$
- uwaga: język L jest wówczas rekurencyjny, maszyna gwarantuje własność stopu i rozstrzygnięcie
- dodajemy literkę N do oznaczenia klasy, jeśli maszyna akceptująca jest niedeterministyczna

Notacja asymptotyczna c.d.

- jeśli b jest „o małe od” c , to jest również „O duże”
- z kolei c na pewno nie jest „O duże od” b
- „ Θ od” oznacza „O duże” w obie strony
- jeśli istnieje granica ilorazu $\frac{b_n}{c_n}$ skończona, niezerowa, to b jest Θ od c , ale bez tej granicy też może być
- relacja „O duże” jest zwrotna i przechodnia, ale brak słabej antysymetrii, relacja „Theta” jest relacją równoważności
- relacja „o małe” jest porządkiem (przeciwzwrotna i przechodnia)
- $\log n$ jest „o małe od” n , n^k jest „o małe od” n^ℓ , $n < \ell$
- n^ℓ jest „o małe od” 2^n dla dowolnej potęgi ℓ
- $100 \cdot n$ jest „o małe od” $n \cdot \log_2 n$, ale dla $n < 2^{100}$ $100 \cdot n > n \cdot \log_2 n$

Ważne klasy złożoności czasowej

Czas działania	deterministyczna	niedeterministyczna
logarytm $O(\log(n))$	L	NL
wielomian $O(p(n))$	P	NP
wykładniczy $2^{p(n)}$	EXPTIME	NEXPTIME

- dla każdej maszyny niedeterministycznej da się zbudować maszynę deterministyczną akceptującą ten sam język
- problem: czy maszynę niedeterministyczną działającą w czasie wielomianowym da się zamienić na deterministyczną działającą również w czasie wielomianowym?
- czyli: $P=NP$?
- główny problem teoretycznej informatyki
- w zasadzie nie wierzymy się da