

SEMINARSKI RAD

Predmet: Teorija Izračunljivosti

Profesor: Zoran Ognjanović

Student: Stefan Banović 177/06

Teorija složenosti izračunavanja

- Teorija algoritama
 - Odlučivi problemi
 - Neodlučivi problemi
- Posmatrajući Čerčovu tezu, dolazimo do zaključka da postoji razlika između:
 - Praktično izračunljivih funkcija
 - Funkcija koje se mogu izračunati u principu

➤ Tjuringova mašina

- Koristi se kao osnovna apstraktna mašina čiji se resursi procenjuju tokom rada programa
- Jedina oblast teorije složenosti gde se ne koriste Tjuringove mašine se odnosi na paralelno izračunavanje

➤ Prednosti Tjuringove mašine

- Relativna jednostavnost
- Mogućnosti
- Efikasnost (osnovne klase složenosti se ne menjaju)

➤ U nastavku ćemo govoriti o sledećim stvarima:

- Složenost problema
- Složenost skupa
- Složenost izvršavanja programa (izračunavanja f -ja)

➤ Praktično izračunljivi problemi

- Problemi kod kojih je dužina rada odgovarajućeg programa limitirana nekim stepenom dužine ulaznih podataka.

➤ Praktično neizračunljivi problemi

- Preostali odlučivi problemi, tj. Izračunljivi u principu
- Ne konstruišemo opšte algoritme, već algoritme za pojedine slučajeve

Opis problema

- Formalni model izračunavanja koji ćemo koristiti u analizi složenosti su Tjuringove mašine sa više traka:
 - Nedetrministička Tjuringova mašina
 - Deterministička Tjuringova mašina
 - Tjuringova mašina sa ulazom i izlazom
- Pokazano je da mašina sa jednom trakom može simulirati gore navedene tipove Tjuringovih mašina. Njihovim korišćenjem se ne povećava izražajnost u odnosu na osnovni model.

➤ Nederministička Tjuringova mašina

- Konačnim brojem $k \geq 1$ traka koje su ograničene sa leve strane i od kojih prva traka sadrži ulazne podatke, a poslednja rezultat
- Konačnim skupom S stanja, od kojih je q_0 početno, a $\{q_z, q_{da}, q_{ne}\}$ je skup završnih stanja koja redom označavaju završetak rada, pozitivan odgovor na pitanje i negativan odgovor na pitanje
- Alfabetom A koji sadrži znake koji se upisuju u ćelije mašine među kojima su marker levog kraja trake $\blacktriangleright$ i blanko znak i
- Programom u kome se svakoj kombinaciji tekućeg stanja i sadržaja ćelija nad kojima se nalaze glave mašine pridružuju jedna, ili više akcija; svaka akcija opisuje da li se ostaje u istom ili se prelazi u novo stanje, da li se upisuje novi sadržaj u ćelije iznad kojih se nalazi glava i da li se glava pomera Levo, Desno ili ostaje tu gde je.

➤ Deterministička Tjuringova mašina

- Od nederminističke se razlikuje samo po tome što se svakoj kombinaciji tekućeg stanja i sadržaja ćelija nad kojima se nalaze glave mašine pridružuje najviše jedna akcija

➤ Tjuringova mašina sa ulazom i izlazom

- Deterministička
- Nedeterministička
- Ima bar dve trake – prva traka služi samo za čitanje, a u poslednju se smesta rezultat (samo za pisanje). Kod poslednje trake glava se ne sme kretati u levo.

- Završno stanje q_z se koristi pri analizi složenosti izračunavanja funkcija. Prelazak u to stanje označava završetak rada programa.
- Ranije smo diskutovali o efikasnosti simulacije i zaključke možemo sumirati u naredne dve teoreme:
 - Teorema 1: Za datu Tjuringovu mašinu M sa k -trakom i vremenskom granicom složenosti $f(n)$ može se konstruisati deterministička Tjuringova mašina M' sa jednom trakom koja simulira rad mašine M i ima vremensku granicu složenosti $O(f(n)^2)$.
 - Teorema 2: Za datu nedeterminističku Tjuringovu mašinu M sa k -trakom i vremenskom granicom složenosti $f(n)$ može se konstruisati deterministička Tjuringova M' sa jednom trakom koja simulira rad mašine M i ima vremensku složenost $O(c^{f(n)})$, za $c > 1$ koje zavisi od mašine M .

- Pored nedeterminističkih i determinističkih mašina u opisima klasa složenosti se koriste i druge varijante Tjuringovih mašina
- Problemi koji se analiziraju u teoriji složenosti izračunavanja karakterišu se pitanjima koja obično imaju odgovor “da” ili “ne”. Npr. Da li je određeni graf povezan? Svaki konkretan graf za koji se postavi ovo pitanje je primerak problema. U nekim situacijama, kod optimizacije recimo, rešenje problema je neki numerički izraz.
- Predstavljanje problema se vrši u nekom formalnom jeziku na alfabetu neke Tjuringove mašine, što se formalno dokazuje sledećom definicijom:

- Definicija : *Problem* L za koji se ispituje soženost je podskup skupa svih reči nekog alfabeta. *Komplement problema* L , u oznaci $\bar{L}$ na nekom alfabetu je skup svih reči na tom alfabetu koje nisu u L .
- Dokaz: Neka je dat alfabet A i neka je L problem. Tjuringova mašina *prihvata* ulazni podatak, odnosno reč x , ako postoji izračunavanje u kome se, polazeći od reči x upisane na ulaznoj traci u početnom stanju q_0 , dolazi do završnog stanja q_{da} , a *odbacuje* reč x ako uvek dolazi do završnog stanja q_{ne} . Ako Tjuringova mašina M prihvata sve reči x jezika koje pripadaju problemu L , a odbacuje svaku reč koja nije u problemu L , kaže se da M *odlučuje* problem L .

Ako Tjuringova mašina M za ulazni podatak x u izračunavanju dolazi do stanja q_z , onda je sadržaj poslednje trake rezultat rada mašine u oznaci $M(x)$. Sa $L(x)$ ćemo označavati primerak problema L za ulazni podatak x , odnosno pitanje da li x pripada problemu L .

O notacija

- U teoriji složenosti izračunavanja često se razmatra brzina rasta funkcije. Brzina rasta se analizira asimptotski, pri čemu se često koriste različite aproksimacije koje opisujemo narednim definicijama i tvrđenjima:
- Definicija 1: Neka su f i g aritmetičke funkcije. Tada je funkcija f u velikom O od g (u oznaci $f(x) = O(g(x))$) ako postoje brojevi c i n takvi da za svaki $x > n$ važi $f(x) \leq c \cdot g(x)$. Ako takvi brojevi ne postoje onda $f(x) \neq O(g(x))$.
Umesto funkcija f je u velikom O od g kaže se da je *funkcija f reda funkcije g* ili *funkcija g je asimptotska gornja granica funkcije f* .

➤ Definicija 2: Funkcija f raste brže od funkcije g ako $f(x) \neq O(g(x))$. Funkcije rastu istom brzinom, u oznaci $f(x) = \Theta(g(x))$, ako važi $f(x) = O(g(x))$ i $g(x) = O(f(x))$.

➤ Teoreme 1: Neka su f i g aritmetičke funkcije i neka je $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \beta$.

Ako je β pozitivan realan broj, onda funkcije f i g rastu istom brzinom. Ako je $\beta = \infty$, onda važi $g(x) = O(f(x))$ i $f(x) \neq O(g(x))$, odnosno funkcija f raste brže od g .

➤ Teorema 2: Neka je $P(n) = a_0 + a_1 \cdot n^1 + \dots + a_r \cdot n^r$, $a_r \neq 0$, polinom stepena r sa celobrojnim koeficijentima. Tada za $P(n)$ i n^m važi:

- ako je $m = r$, $P(n)$ i n^m rastu istom brzinom
- ako je $m < r$, $P(n)$ raste brže od n^m
- ako je $m > r$, n^m raste brže od $P(n)$

➤ Teorema 3: Neka je $k > 1$. Funkcije k^n raste brže od bilo kog polinoma sa celobrojnim vrednostima. Svaki polinom sa celobrojnim koeficijentima raste brže od bilo koje logaritamske funkcije.

Kako je $\log_c x = \log_c d \cdot \log_d x$, direktno sledi sledeća teorema.

- Teorema 4: Za svake dve realne konstante $c, d > 1$ važi
- $$\log_c(x) = \Theta(\log_d(x))$$

Funkcije koje se javljaju u O-notaciji prilikom analize složenosti su: logaritamska funkcija $\log_2 n$, linearna funkcija $k \cdot n$, njihov proizvod $n \log_2 n$, stepen funkcije n^k , eksponencijalna funkcija k^n itd. Svima njima je zajedničko:

- $\lim_{n \rightarrow \infty} f(n) = \infty$, što ima razumljivo intuitivno opravdanje: što je problem većih dimenzija složenost izračunavanja je veća
 - funkcije su neopadajuće
 - postoje Turingove mašine koje ih izračunavaju u prostoru i vremenu koji su proporcionalni vrednostima funkcija.
- Ovakve funkcije se nazivaju *prave funkcije složenosti* i upotrebljavaju se u analizi složenosti izračunavanja.