Teorija izračunljivosti

Ivan Prokić
kabinet 117, F blok
prokic@uns.ac.rs
http://imft.ftn.uns.ac.rs/∼iprokic/

Novi Sad

Tema 1

Nedeterministička Tjuringova mašina

Nedeterministička Tjuringova mašina

- Tjuringove mašine koje smo do sada spominjali su u svakoj konfiguraciji imale deterministički (jednoznačno) određen sledeći korak;
- Sada ćemo definisati **nedeterminističke Tjuringove mašine**: iz trenutne konfiguracije mašina može imati više mogućih koraka, i pri tome mi ne utičemo na izbor sledećeg koraka;
- Ovakve mašine su samo koncept, tj. njihova implementacija nije realna - ali koncept nedeterminizma se svakako pojavljuje u realnim računarskim sistemima, npr. u konkurentnim i distribuiranim.

Nedeterministička Tjuringova mašina

Nedeterministička Tjuringova mašina može se smatrati upoštenjem determinističke Tjuringove mašine - jedina razlika je u tome što se funkcija prelaza definiše sa

$$\delta : Q \times \Gamma \to P\big(Q \times \Gamma \times \{L, R\}\big),$$

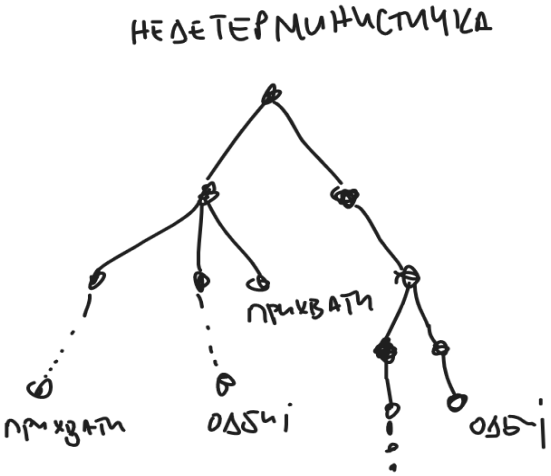
tj. za dato stanje q i slovo a funkcija prelaza može definisati više mogućih sledećih koraka

$$\delta(q, a) = \begin{cases} (q_1, b_1, S_1) \\ (q_2, b_2, S_2) \\ \dots \\ (q_n, b_n, S_n) \end{cases}$$

gde $S_i \in \{L, R\}$.

Nedeterministička Tjuringova mašina: drvo izračunavanja

Izračunavanje nedeterminističke Tjuringove mašine možemo prikazati kao drvo, pri čemu su čvorovi konfiguracije, a grane predstavljaju prelaze među konfuguracijama. Isto možemo uraditi i za determinističku Tjuringovu mašinu - takvo drvo ima samo jednu granu.



Jezik nedeterminističke Tjuringove mašine

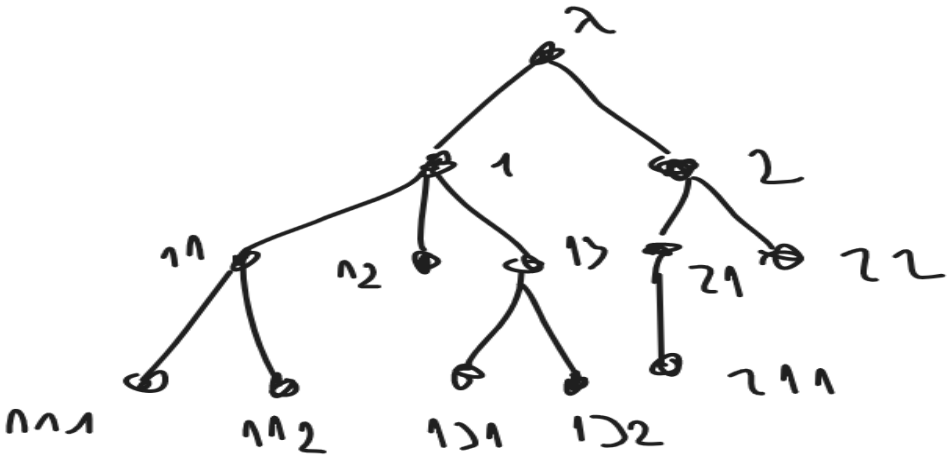
- Kažemo da nedeterministička Tjuringiva mašina N prihvata reč w ako bar jedna grana drveta izračunavanja vodi ka prihvatanju.
- Jasno da je klasa svih jezika determinističkih sadržan u klasi jezika svih nedeterminističkih Tjuringovih mašina (jer svaku determinističku mašinu možemo smatrati nedeterminističkom koja u svakoj konfiguraciji ima samo jedan mogući sledeći korak).
- sledeća teorema pokazuje da su ove dve klase zapravo ekvivalentne, odnosno da je jezik svake nedeterminističke mašine $\mathbf{RE}$.

Teorema

Za svaku nedeterminističku Tjuringovu mašinu N postoji (3-tračna) deterministička Tjuringova mašina M takva da je $L(N) = L(M)$.

Ideja za dokaz teoreme

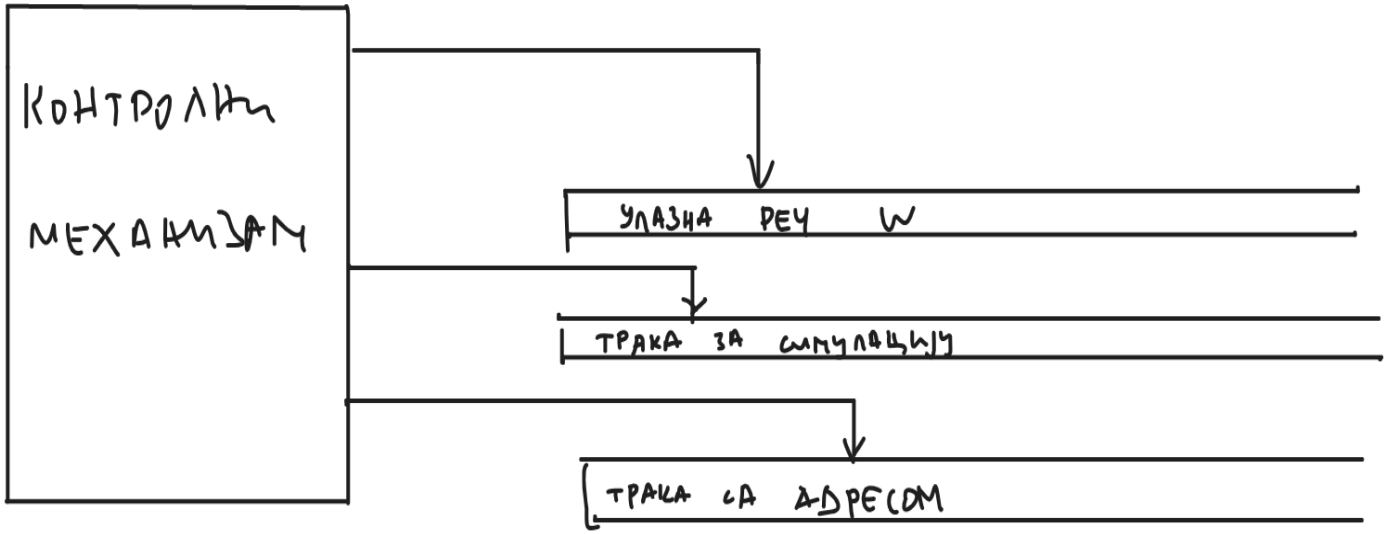
- Ideja dokaza teoreme je da konstruišemo determinističku Tjuringovu mašinu koja će da simulira sva moguća izračunavanja nedeterminističke mašine. Zapravo, radićemo pretragu drveta izračunavanja nedeterminističke mašine - i to po širini.
- Ideja je da čvorove drveta posećujemo po nivoima počevši od prvog (koji sadrži jedan čvor koji predstavlja početnu konfiguraciju).
- Da bi bili sigurni da ćemo posetiti sve čvorove, svakom čvoru dodelićemo jedinstven niz prirodnih brojeva: ako nedetermonistička mašina u svakoj konfiguraciji ima najviše b izbora, generisaćemo nizove elemanata skupa $\{1, \ldots, b\}$ u leksikografskom redosledu. Npr. u sledećem stablu je $b = 3$ pa imamo:



Dokaz teoreme $1/2$

Deterministička mašina M (koja simulira rad nedeterminističke N) ima tri trake.

- Na prvoj traci je ulazna reč w, i sadržaj ove trake se nikad ne menja;
- Druga i treća traka su na početku prazne. Druga će služiti za simulaciju rada N za jedan put u drvetu izračunavanja - i to do čvora čija adresa piše na trećoj traci.



Dokaz teoreme $2/2$

M simulira N na sledeći način:

- Kopiraj ulaz w sa trake 1 na traku 2;
- Koristi traku 2 da simuliraš rad mašine N za ulaz w. Pre svakog koraka u simulaciji proveri koji je sledeći simbol na traci 3, i na osnovu njega odaberi odgovarajući prelaz (od onih koje N može da napravi). Ako na traci 3 nema više simbola ili ako adresa koja je pročitana ne odgovara ni jednom putu u drvetu izračunavanja - pređi na sledeći korak. Takođe se prelazi na sledeći korak ako se uđe u konfiguraciju odbijanja. Ako se uđe u konfiguraciju prihvatanja - **prihvati** reč w;
- Zameni niz na traci 3 sa leksikografski sledećim nizom. Vrati se na prvi korak (gde se sada simulira sledeća ”grana” drveta izračunavanja mašine N).

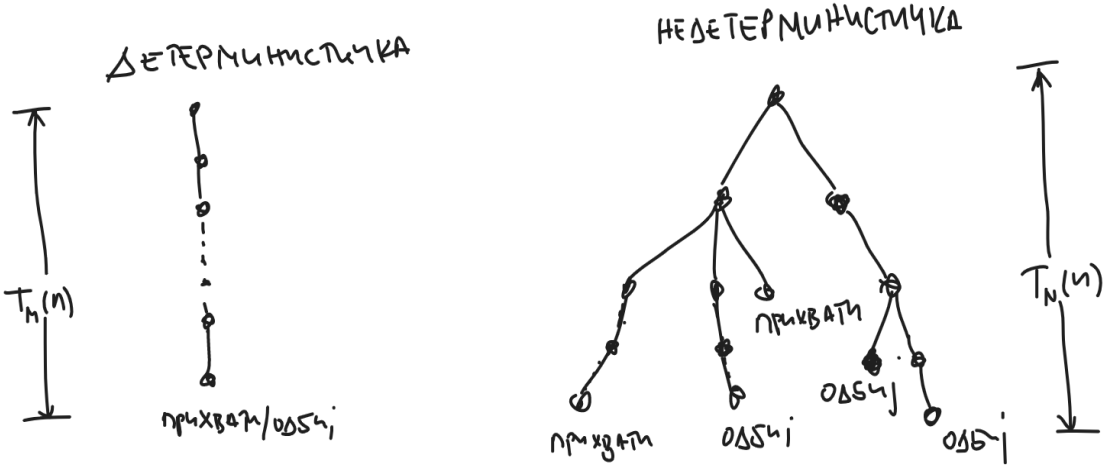
Deterministička Tjuringova mašina M vrši pretragu kompletnog stabla nedeterminističke mašine N za reč w - uz to, M prihvata reč w ako i samo ako N prihvata w.

Totalne nedeterminističke Tjuringove mašine i vremenska složenost

Nedeterministička Tjuringova mašina je **totalna** ako se zaustavlja za svaki ulaz u svim granama drveta izračunavanja.

Definicija

Neka je N totalna nedeterministička Tjuringova mašina. Funkcija $T_N : \mathbb{N} \to \mathbb{N}$ je **ocena vremenske složenosti** *za N ako za svaku ulaznu reč w, čija je dužina $|w| \leqslant n$, N se zaustavlja u najviše $T_N(n)$ koraka u svakoj od grana drveta izračunavanja (tj. drvo izračunavanja nije dublje od $T_N(n)$).*



Totalne nedeterminističke Tjuringove mašine i vremenska složenost

Teorema

Za svaku totalnu nedeterminističku Tjuringovu mašinu N složenosti $T_N(n)$ postoji totalna deterministička M složenosti $T_M(n) = O(2^{T_N(n)})$ takva da je $L(N) = L(M)$.

Ovde će nam poslužiti 3-tračna deterministička mašina M' iz prethodnog dokaza sa malim dodatkom: ako je $h = T_N(n)$ visina drveta izračunavanja mašine N za ulaz w (gde je $|w| = n$) mašina M' će na trećoj traci ispisivati sve nizove nad $\{1, \ldots, b\}$ ali ne duže od h. Ako na kraju ispisivanja svih nizova duzine $\leqslant h$ ne nađe konfiguraciju prihvatanja - M' prelazi u stanje odbijanja.
Sada ocenjujemo složenost: Ukupan broj čvorova stabla izračunavanja je manji ili jednak dvostrukom maksimalnom broju listova, pa ga ograničavamo sa $O(b^h)$. Vreme potrebno da se u simulaciji od korena stabla stigne do nekog čvora ograničavamo sa $O(h)$. Tako dobijamo $T_{M'}(n) = O(hb^h) = 2^{O(h)} = 2^{O(T_N(n))}$.
Dalje znamo da 3-tračnu (determinističku) mašinu M' možemo simulirati na mašini M sa jednom trakom sa najviše kvadratnim usporenjem. Zato, nedeterminističku N možemo simulirati na determinističkoj M (koja ima jednu traku) i važi $T_M(n) = (2^{O(T_N(n))})^2 = 2^{O(2T_N(n))} = 2^{O(T_N(n))}$.

Tema 2

Klasa složenosti $\mathbf{NP}$

Polinomna verifikacija

Definicija

Verifikator *za jezik* A *je algoritam* V*, takav da*

$$A = \{w \mid V \text{ prihvata } <w, c> \text{ za neku reč } c\}.$$

Polinomni verifikator *za jezik* A *je verifikator* V *polinomne složenosti (u odnosu na dužinu reči* w*). Reč* c *zovemo* **sertifikat** *(ili* **dokaz***) za* w*.*

Primer

U SAT problemu, za zadovoljivu Bulovu KNF formulu ϕ *sertifikat bi bio jedna valuacija* τ *u kojoj je formula tačna, tj.* $v_\tau(\phi) = \top$*. Polinomni verifikator za SAT problem bi bio algoritam koji u polinomnom vremenu proverava da li je Bulovu KNF formula* ϕ *u valuaciji* τ *tačna. Takav algoritam postoji - izračunavanje istinitosne vrednosti Bulove KNF formule u jednoj valuaciji može se uraditi u linearnom vremenu u odnosu na dužinu formule* ϕ*.*

Klasa složenosti $\mathbf{NP}$

Definicija

$\mathbf{NP}$ *je klasa svih jezika koji imaju polinomne verifikatore.*

Primer

Neki primeri $\mathbf{NP}$ *problema: SAT, problem Hamiltonovih kontura, problem klike,* 3*-obojivost grafa, problem trgovačkog putinka, itd.*

Ime $\mathbf{NP}$ dolazi od nedeterminističko polinomno vreme. Sledeća teorema pokazuje da smo klasu jezika $\mathbf{NP}$ mogli definisati baš kao jezike odlučive u polinomnom vremenu na nedeterminističkoj Tjuringovoj mašini.

Teorema

Jezik L *je u klasi* $\mathbf{NP}$ *ako i samo ako postoji totalna nedeterministička Tjuringova mašina* N *polinomne složenosti takva da je* $L(N) = L$.

Dokaz teoreme

($\Leftarrow$): Ideja je da iskoristimo polinomni verifikator da konstruišemo polinomnu nedeterminističku mašinu koja će da pogodi sertifikat.
Neka je $L \in \mathbf{NP}$ i neka je V polinomni verifikator (deterministička Tjuringova mašina) složenosti n^k. Konstruišemo totalnu nedeterminističku Tjuringovu mašinu N koja odlučuje L.

$N =$ "Za ulaz w dužine n:

1. Nedeterministički odaberi reč c dužine najviše n^k;
2. Pokreni V za ulaz $<w, c>$;
3. Ako V prihvata - prihvati, inače - odbij."

($\Rightarrow$): Ideja je da iskoristimo polinomnu nedeterminističku mašinu da konstruišemo verifikator, pri čemu će jedna grana u kojoj mašina prihvata poslužiti kao sertifikat.
Neka je N totalna nedeterministička Tjuringova mašina takva da je $L(N) = L$. Konstruišemo polinomni verifikator (determinističku Tjuringovu mašinu) V.

$V =$ "Za ulaz $<w, c>$:

1. Simuliraj rad mašine N za ulaz w, koristeći simbole sertifikata c za odabir sledećeg koraka u funkciji prelaza od N (kao što koristili adrese u simulaciji nedeterminističke mašine na determinističkoj);
2. Ako ova grana u izračunavanju mašine N prihvata - prihvati, inače - odbij."

Druga definicija klase $\mathbf{NP}$

Klase složenosti nedeterminističkih Tjuringovuh mašina N definišemo analogno determinističkim M:

$$NTIME(f(n)) = \{L : \exists N \text{ takvo da je } L(N) = L \text{ i } T_N(n) = O(f(n))\}.$$

Posledica

$$\mathbf{NP} = \bigcup_{k \geqslant 0} NTIME(n^k).$$

Jasno da važi $\mathbf{P} \subseteq \mathbf{NP}$.

$\mathbf{P} \neq \mathbf{NP}$?

- Simulacija nedeterminističke mašine na determinističkoj pokazuje zašto imamo eksponencijalno usporenje: potrebno je ispitati sve moguće izbore metodom ”brute-force”;
- Kod problema koji su u klasi $\mathbf{P}$, npr. HORNSAT problem, mogli smo da iskoristimo poznavanje osobina problema pa da skratimo pretragu i da rezultujući algoritam bude polinomne složenosti;
- Kod problema koji su $\mathbf{NP}$ kompletni (pojam koji će uskoro biti objašnjen) nije poznato da li postoji način da se pretraga kroz stablo izračunavanja skrati pa da dobijemo polinomni algoritam;
- Pronaći takav algoritam - ili dokazati njegovo nepostojanje - jedan je od najvećih otvorenih problema matematike (uz to vredi i milion dolara!).

Tema 3

NP-kompletnost

$\mathbf{NP}$-kompletnost

- Važan proboj u formulaciji problema $\mathbf{P} \neq \mathbf{NP}$ napravili su 1970-tih godina Stiven Kuk i Leonid Levin;
- Oni su okrili da postoje problemi u klasi $\mathbf{NP}$ čija se složenost može dovesti u vezu složenosti čitave klase;
- Ti problemi zovu se $\mathbf{NP}$**-kompletni**.

Fenomen $\mathbf{NP}$-kompletnosti važan je i sa teorijskog i sa praktičnog stanovišta:

- Teorijski: pokazati da je $\mathbf{P} \neq \mathbf{NP}$ svodi se na pokazivanje da bilo koji $\mathbf{NP}$ problem zahteva više od polinomnog vremena; pokazati da je $\mathbf{P} = \mathbf{NP}$ svodi se na pronalaženje polinomnog rešenja za bilo koji $\mathbf{NP}$-kompletan problem.
- Praktičan: Ako se problem koji rešavamo pokaže kao $\mathbf{NP}$-kompletan, možemo ga smatrati ”teškim”, i odustati od traženja ”polinomno efikasnog” algoritma.

Polinomne redukcije

Polinomna redukcija je izračunljiva funkcija polinomne složenosti koja jedan problem konvertuje u drugi.

Definicija

Funkcija $f : \Sigma^* \to \Sigma^*$ je **polinomno izračunljiva** *ako postoji Tjuringova mašina polinomne složenosti koja se za svaki ulaz $w \in \Sigma^*$ zaustavlja sa $f(w)$ na traci (pri čemu glava pokazuje na prvo slovo reči $f(w)$).*

Sada definišemo polinomne redukcije između jezika.

Definicija

Jezik A se **polinomno redukuje** *na jezik B, u oznaci $A \leqslant_P B$, ako postoji polinomno izračunljiva funkcija $f : \Sigma^* \to \Sigma^*$ takva da za svako w važi:*

$$w \in A \quad \Leftrightarrow \quad f(w) \in B.$$

Funkcija f iz prethodne definicije se zove **polinomna redukcija** iz A u B.

Polinomne redukcije

Teorema

Ako je $A \leqslant_P B$ i $B \in \mathbf{P}$, tada je i $A \in \mathbf{P}$.

Neka je M totalna Tjuringova mašina polinomne složenosti koja odlučuje B i neka je f polinomna redukcija iz A u B (tj. $w \in A \Leftrightarrow f(w) \in B$). Sada jezik A odlučuje totalna mašina N polinomne složenosti data sa:

$N =$ "Za ulaz w:

1. Izračunaj $f(w)$;
2. Pokreni mašinu M za ulaz $f(w)$. Ako M prihvata (tj. ako $f(w) \in B$) - prihvati, ako M odbija (tj. ako $f(w) \notin B$) - dobij."

Oba koraka troše polinomno vremene, pa je i N polinomne složenosti (jer je kompozicija dva polinoma ponovo polinom).

$\mathbf{NP}$-kompletnost

Definicija

Jezik A je $\mathbf{NP}$-kompletan ako važe sledeća dva uslova:

- $A \in \mathbf{NP}$ *i*
- *za svaki jezik B koji je u $\mathbf{NP}$ važi $B \leqslant_P A$.*

$\mathbf{NP}$-kompletni problemi su najteži predstavnici klase $\mathbf{NP}$.

Teorema

Ako je A $\mathbf{NP}$-kompletan, $A \leqslant_P B$ i $B \in \mathbf{NP}$, tada je i B $\mathbf{NP}$-kompletan.

Ako je A $\mathbf{NP}$-kompletan, tada se svaki jezik C koji je u $\mathbf{NP}$ važi $C \leqslant_P A$. Sada imamo $C \leqslant_P A \leqslant_P B$ za sve C iz $\mathbf{NP}$ klase, a kako je kompozicija polinomnih redukcija ponovo polinomna redukcija, imamo da je B $\mathbf{NP}$-kompletan.

Kuk-Levinova teorema

Teorema (Kuk-Levinova)

SAT problem je $\mathbf{NP}$-kompletan.

Ideja dokaza je da za jezik A koji je iz klase $\mathbf{NP}$ i reč w iz azbuke jezika A konstruišemo iskaznu formulu $\phi(N, w)$ (gde je N nedeterministička Tjuringova mašina za koju je $L(N) = A$) za koju će važiti

$$w \in L(N) \text{ ako i samo ako je } \phi(N, w) \text{ zadovoljiva.}$$

Ovde nećemo ići u detalje dokaza.

Primer polinomne redukcije

Problem KLIKE: Za dati neorijentisani graf $G = (V, E)$ i prirodan broj k utvrditi da li G ima kompletan podgraf (kliku) od k čvorova.

KLIKE je u klasi $\mathbf{NP}$: sertifikat je B podskup skupa čvorova V, a verifikacija je provera da li je podgraf sastavljen samo od čvorova iz B kompletan, odnosno da li su mu svaka dva različita čvora susedni (ako B ima k čvorova, imali bi $\frac{k(k-1)}{2}$ provera susednosti).

Teorema

Problem KLIKE je $\mathbf{NP}$-kompletan.

Dokaz teoreme

Zapravo ćemo pokazati: SAT $\leqslant_P$ KLIKE
Neka je ϕ Bulova formula u KNF obliku. Konstruišemo graf $G_\phi = (V_\phi, E_\phi)$ na sledeći način:

- Za svaki literal u svakom konjuktu dodamo po jedan čvor u graf G_ϕ. Na primer, za

  $$\phi = (x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee y \vee \neg z),$$

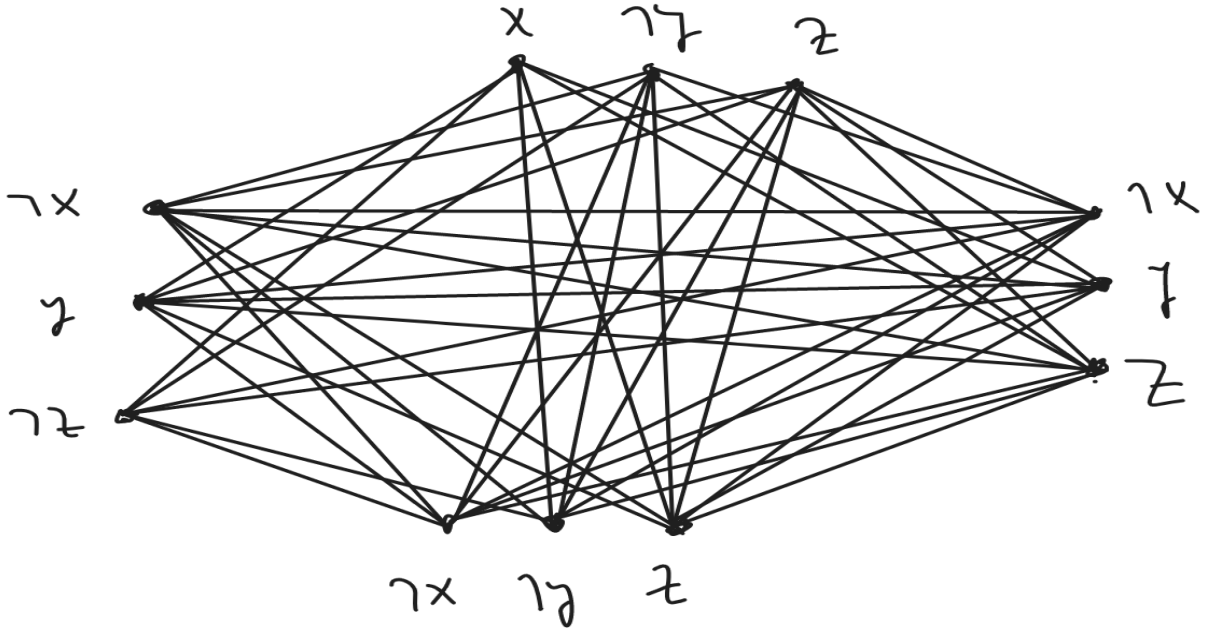
  skup čvorova bi bio $V_\phi = \{x, \neg y, z, \neg x, y, z, \neg x, \neg y, z, \neg x, y, \neg z\}$;
- granama spajamo svaka dva čvora, sem:
  1. ako su to dva čvora koja dolaze iz istog konjukta i
  2. ako su to dva čvora koja odgovaraju komplementarnim literalima (npr. x i $\neg x$).

Dokaz teoreme: primer

Na primer, za

$$\phi = (x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee y \vee \neg z),$$

odgovarajući graf G_ϕ je



Dokaz teoreme

Sada pokazujemo da je ϕ zadovoljiva ako i samo ako graf G_ϕ ima m-kliku, gde je m broj konjukata u ϕ.

($\Rightarrow$): Neka je ϕ zadovoljiva. Neka je τ valuacija za koju važi $v_\tau(\phi) = \top$. To znači da je u svakom konjuktu postojao bar jedan literal koji je bio tačan u τ. Neka je $\{a_1, \ldots, a_m\}$ skup tih literala (iz svakog konjukta po jedan). Tada je podgraf od $\{a_1, \ldots, a_m\}$ čvorova kompletan: pošto su a_i i a_j (za $i \neq j$) iz različitih konjukata i oba su tačna u istoj valuaciji - pa ne mogu biti komplementarni - to znači da postoji grana između ova dva čvora.

($\Leftarrow$): Neka G_ϕ ima m-kliku $\{a_1, \ldots, a_m\}$. Posmatrajmo valuaciju $\tau : X_\phi \to \{\top, \bot\}$ datu sa

$$\tau(a) = \begin{cases} \top & a \in \{a_1, \ldots, a_m\} \\ \bot & \text{inače.} \end{cases}$$

Primetimo da a_i može biti iskazno slovo ili njegova negacija, dok je a iskazno slovo. Tada je $v_\tau(\phi) = \top$ jer je u i-tom konjuktu literal a_i tačan i svaka dva literala iz $\{a_1, \ldots, a_m\}$ nisu komplementarni (na osnovu konstrukcije grafa).

Još treba primetiti da je redukciona funkcija izračunljiva u polinomnom vremenu u odnosu na n dužinu formule ϕ.

Graf možemo konstruisati za $O(n^3)$ vremena: broj čvorova grafa je $O(n)$, provera susednosti dva čvora (kojih ima $O(n^2)$) je $O(n)$, provera komplementarnosti $O(1)$.

Dokaz teoreme: primer

$$\phi = (x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee y \vee \neg z),$$

je tačna u valuaciji $\tau(x) = \tau(y) = \tau(z) = \top$, a jedna odgovarajuća 4-klika je