Konstrukcija i analiza algoritama
vežbe 11

Nina Radojičić

22. decembar 2016

1 NP kompletnost

Za algoritam kažemo da je *efikasan* ako je njegova vremenska složenost $O(P(n))$, gde je $P(n)$ je polinom od veličine problema n, a za problem kažemo da je *rešiv*. Bitno je naglasiti da veličinu ulaza definišemo kao broj bitova potrebnih za predstavljanje ulaza. Klasu svih problema koji se mogu rešiti efikasnim algoritmom označavaćemo sa $\mathcal{P}$.

Postoje problemi za koje se danas ne zna ni jedan algoritam polinomijalne složenosti, moguće je da će oni biti rešeni efikasnim algoritmom jednog dana, ali je sasvim moguće i da postoje problemi koji se ne mogu rešiti efikasno. Razmatraćemo probleme za koje se ne zna da li su u klasi $\mathcal{P}$, sa akcentom na jednoj potklasi tzv. $\mathcal{NP}$-kompletnih problema. Važi da za neki $\mathcal{NP}$-kompletan problem postoji efikasan algoritam AKKO za svaki $\mathcal{NP}$-kompletan problem postoji efikasan algoritam. Vlada uverenje da ne postoji efikasan algoritam ni za jedan $\mathcal{NP}$-kompletan problem, iako ovo nije dokazano.

Ograničavamo se na *probleme odlučivanja* (problemi koji nakon izvršenja algoritma odgovaraju sa "DA"ili "NE"). Obično ako umemo da rečimo problem odlučivanja, umemo i da rešimo i polazni problem (npr. binarnom pretragom). Problem odlučivanja možemo da posmatramo i kao problem prepoznavanja jezika, tako što definišemo jezike L kao podskup skupa U svih mogućih ulaza za koje je rešenje "DA", gde kažemo da jezik odgovara problemu. Sada se naš zadatak svodi na utvrđivanje da li zadati ulaz pripada jeziku L.

Ako su L_1 i L_2 dva jezika koji su podskupovi skupova ulaza U_1 i U_2, kažemo da je L_1 *polinomijalno svodljiv* na L_2 ako postoji algoritam polinomijalne vremenske složenosti (u osnovu na veličinu ulaza $u_1 \in U_1$) koji za dati (proizvoljni) ulaz $u_1 \in U_1$ prevodi u ulaz $u_2 \in U_2$ tako da:

$$u_1 \in L_1 \Longleftrightarrow u_2 \in L_2$$

Ako znamo algoritam AL_2 za prepoznavanje jezika L_2 i neka je algoritam redukcije označen sa AR, tada proizvoljan ulaz $u_1 \in U_1$ možemo algoritmom redukcije AR transformisati u ulaz $u_2 \in U_2$ i primenom AL_2 ustanoviti da li $u_2 \in U_2$ i primenom AL_2 ustanoviti da li $u_2 \in L_2$, a time i da li $u_1 \in L_1$.

Relacija polinomijalne svodljivosti **nije simetrična**, tj. polinomijalna svodljivost jezika L_1 na L_2, ne povlači uvek polinomijalna svodljivost jezika L_2 na L_1. Može se desiti da se proizvoljan ulaz za L_1 može transformisati u ulaz za L_2, ali ne i obrnuto. Moguće je da ulazi za L_2 ovako dobijeni predstavljaju samo mali deo svih ulaza za L_2.

Ako je svaki od jezika L_1 i L_2 polinomijalno svodljiv na drugi, onda za jezike L_1 i L_2 kažemo da su *polinomijalno ekvivalntni*.

Relacija polinomijalne svodljivosti jeste **tranzitivna** tj. ako je L_1 polinomijalno svodljiv na L_2 i L_2 polinomijalno svodljiv na L_3, onda je L_1 polinomijalno svodljiv na L_3.

Osnovni model izračunavanja je ***Tjuringova mašina***. Deterministička Tjuringova mašina (DTM) se sastoji od upravljačkog bloka sa konačnim brojem stanja, glave za čitanje/pisanje i dvostrano neograničene trake.

Vreme posmatramo kao broj koraka DTM do zaustavljanja. Ako se program M zaustavlja za sve ulaze $x \in \Sigma^*$, onda se vremenska složenost definiše kao:

$$T_M(n) = max\{m| \text{ postoji } x \in \Sigma^*, |x| = n,$$
$$\text{tako da je vreme izvršavanja programa } M \text{ za ulaz } x \text{ jednako } m\}$$

M je polinomijalni program za DTM ako postoji polinom p tako da je $\forall n \in Z^+ : T_M(n) \leq p(n)$.

Klasu $\mathcal{P}$ definišemo kao:

$\mathcal{P} = \{L| \text{ postoji polinomijalni program } M\text{za DTM tako da je } L_M = L\}$

Polinomijalna proverljivost je pojam koji karakteriše klasu $\mathcal{NP}$. Bitno je uočiti da polinomijalna proverljivost ne povlači nužno i rešavanje za polinomijalno vreme! Polinomijalna proverljivost nam daje da za konkretan ulaz mi za polinomijalno vreme možemo da utvrdimo da li je za taj ulaz odgovor "DA", ali se ovde ne uzima u obzir vreme potrebno za nalaženje rešenja.

Klasu $\mathcal{NP}$ definišemo pomoću pojma nedeterminističkog algoritma - on se sastoji iz dve faze: faze pogađanja i faze provere. Za dati ulaz u pogađa se neka struktura S i zatim se u i S zajedno predaju fazi provere koja se izvodi na deterministički način i završava se odgovorom "DA"ili "NE", ili se izvršava beskonačno dugo.

Nedeterministički algoritam rešava problem odlučivanja P ako su za proizvoljan ulaz $u \in U_P$ za ovaj problem ispunjena sledeća dva uslova:

- Ako je $u \in L_P$ onda postoji struktura S čije bi pogađanje za ulaz u dovelo do toga da faza provere sa ulazom (u, S) završi odgovorom "DA"
- Ako je $u \notin L_P$ onda ne postoji takva struktura S čije bi pogađanje za ulaz u obezbedilo završavanje faze provere sa ulazom (u, S) završi odgovorom "DA"

Kažemo da nedeterministički algoritam koji rešava problem odlučivanja radi za polinomijalno vreme ako postoji polinom p tako da za svaki ulaz $u \in U_P$ postoji pogađanje S tako da se faza provere sa ulazom (u, S) završava sa odgovorom "DAža vreme $p(|u|)$.

Klasa $\mathcal{NP}$ bi neformalno bila klasa svih problema odlučivanja koji mogu biti rešeni nedeterminističkim algoritmom za polinomijalno vreme. Naglasimo još jednom da je osnovni smisao polinomijalnog nedeterminističkog algoritma njegova proverljivost za polinomijalno vreme.

Izgleda razumno da su nedeterministički algoritmi moćniji od determinističkih, ali ovo nije pokazano.

- Da bi važilo $\mathcal{P} \neq \mathcal{NP}$, dovoljno bi bilo pronaći neki $\mathcal{NP}$ problem da nije u $\mathcal{P}$.
- Da bi važilo $\mathcal{P} = \mathcal{NP}$, trebalo bi pokazati da svaki problem iz klase $\mathcal{NP}$ može biti rešen determinističkim algoritmom polinomijalne složenosti.

Definicija Kažemo da je problem X $\mathcal{NP}$-težak ako je svaki problem iz klase $\mathcal{NP}$ polinomijalno svodljiv na X.

Definicija Kažemo da je problem X $\mathcal{NP}$-kompletan ako važi:

- X pripada klasi $\mathcal{NP}$
- X je $\mathcal{NP}$-težak

Posledica 1.1 *Ako za bilo koji $\mathcal{NP}$-težak problem dokazemo da pripada klasi $\mathcal{P}$, onda je $\mathcal{P} = \mathcal{NP}$.*

Teorema 1.2 *Problem X $\mathcal{NP}$-kompletan ako:*

- *X pripada klasi $\mathcal{NP}$*
- *postoji $\mathcal{NP}$-kompletan problem Y koji je polinomijalno svodljiv na X*

Teorema 1.3 (Kukova teorema) *SAT problem $\mathcal{NP}$-kompletan.*

1.1 Zadaci

1. Pokazati da su svaka dva netrivijalna problema iz klase $\mathcal{P}$ polinomijalno ekvivalentna.

   Rešenje: Neka su B i C dva netrivijalna problema iz klase $\mathcal{P}$. Netrivijalna znači da za oba problema postoje ulazi sa odgovorom "DA"i ulazi sa odgovorom "NE".

   Neka je u_0 ulaz za problem C tako da je za u_0 rešenje "NE", a neka je u_1 ulaz za rešenje "DA"tj. $u_0 \notin L_C$ i $u_1 \in L_C$.

   Da bismo sveli B na C možemo iskoristiti sledeći polinomijalni algoritam:

   - Ulazu v za problem B pridružujemo $\Phi(v) = u_0$, ako $v \notin L_B$
   - Ulazu v za problem B pridružujemo $\Phi(v) = u_1$, ako $v \in L_B$

   Tada važi:
   $$v \in L_B \Longleftrightarrow \Phi(v) \in L_C$$
   tj. jezik B je polinomijalno svodljiv na C.

   Analogno se pokazuje da je i C polinomijalno svodljiv na B tj. problemi B i C su polinomijalno ekvivalentni.

2. Problem 3SAT je $\mathcal{NP}$-kompletan.

   3SAT problem: Zadat je Bulovski izraz u KNF tako da svaka klauza sadrži tačno tri literala. Treba utvrditi da li je izraz zadovoljiv.

   Rešenje:

- Jeste u klasi $\mathcal{NP}$ jer se za izabrane vrednosti promenljivih za polinomijalno vreme može utvrditi da li je izraz tačan.
- Proizvoljan ulaz za SAT svodimo na ulaz za 3SAT tako da je rešenje problema SAT "DA"AKKO je "DA"rešenje problema 3SAT.

  Neka je $c = (x_1 + x_2 + \cdots + x_k)$, $k \geq 4$ proizvoljna klauza. Uvodimo nove promenljive $y_1, \ldots, y_{k-3}$:

  $$c' = (x_1 + x_2 + y_1)(x_3 + \bar{y}_1 + y_2)(x_4 + \bar{y}_2 + y_3)\ldots$$
  $$(x_{k-2} + \bar{y}_{k-4} + y_{k-3})(x_{k-1} + x_k + \bar{y}_{k-3})$$

  Izraz dobijen od izraza E zamenom c sa c' je zadovoljiv AKKO je zadovoljiv izraz E.

  Klauzu $c = (x_1 + x_2)$ menjamo sa $c' = (x_1 + x_2 + z)(x_1 + x_2 + \bar{z})$, a klauzu $c = x_1$ sa $c' = (x_1 + y + z)(x_1 + \bar{y} + z)(x_1 + y + \bar{z})(x_1 + \bar{y} + \bar{z})$.

3. Neka je E izraz u KNF, u kome se svaka promenljiva x pojavljuje najviše jednom i njena negacija $\bar{x}$ se pojavljuje najviše jednom. Konstruisati algoritam polinomijalne vremenske složenosti za utvrđivanje da li je izraz E zadovoljiv, ili dokazati da je ovaj problem $\mathcal{NP}$-kompletan.

   Rešenje: Ukupan broj literala u formuli je najviše $2n$, gde je n broj promenljivih. U zavisnosti od toga da li se i kako javljaju x_n i $\bar{x}_n$, izraz E uzima jedan od oblika:

   1) $(x_n + A)(\bar{x}_n + B)E'$
   2) $(x_n + A)E'$
   3) $(\bar{x}_n + B)E'$

   gde se y A, B i E' ne javlja ni x_n ni $\bar{x}_n$.

   Izrazi 2) i 3) su zadovoljivi AKKO je zadovoljivo E'.

   Izraz 1) je zadovoljiv AKKO je zadovoljiv izraz CE', gde je C' klauza koja se dobija od $A+B$ kada se ponovljeni literali, a ako se u $A+B$ javlja promenljiva i njena negacija, onda je $C' = 1$.

   U sva tri slučaja dobijeni izraz zvisi samo od promenljivih $x_1, x_2, \ldots, x_{n-1}$ i svaki literal se javlja najviše jednom. Znači ovim postupkom smo eliminisali jednu promenljivu za polinomijalno vreme $O(n)$, te je odgovarajući algoritam polinomijalne složenosti.

4. Napisati 3SAT izraz koji se dobija primenom postupka svođenja SAT problema na 3SAT problem za izraz:

   $$(x + y + \bar{z} + w + u + \bar{v})(\bar{x} + \bar{y} + z + \bar{w} + u + v)(x + \bar{y} + \bar{z} + w + u + \bar{v})(x + \bar{y})$$

   Rešenje:

   $$(x + y + a_1)(\bar{z} + \bar{a}_1 + a_2)(w + \bar{a}_2 + a_3)(u + \bar{v} + \bar{a}_3)\cdot$$
   $$(\bar{x} + \bar{y} + a_4)(z + \bar{a}_4 + a_5)(\bar{w} + \bar{a}_5 + a_6)(u + v + \bar{a}_6)\cdot$$
   $$(x + \bar{y} + a_7)(\bar{z} + \bar{a}_7 + a_8)(w + \bar{a}_8 + a_9)(u + \bar{v} + \bar{a}_9)\cdot$$
   $$(x + \bar{y} + a_{10})(x + \bar{y} + \bar{a}_{10})$$

5. kSAT je problem određivanja da li je zadovoljiv dati logički izraz u KNF-u, tako da svaka klauza sadrži tačno k literala.

   Poznato je da je 2SAT rešiv u polinomijalnom vremenu. Svesti 2SAT problem na 4SAT problem. Da li opisana redukcija problema znači da je i 4SAT problem rešiv u polinomijalnom vremenu?

   Rešenje: Neka je $E = E_1 \cdot E_2 \cdots E_n$ proizvoljna instanca 2SAT problema. Ideja je da svaku klauzu zamenimo određenim brojem klauza od po 4 literala.

   $$E_i = x + y$$

   menjamo sa:

   $$E_i' = (x + y + a + b) \cdot (x + y + \bar{a} + b) \cdot (x + y + a + \bar{b}) \cdot (x + y + \bar{a} + \bar{b})$$

   Važi E_i je zadovoljiv $\Longleftrightarrow$ E_i' je zadovoljiv.

   ($\Longrightarrow$) Bar jedan od literala x i y ima vrednost 1 $\Longrightarrow$ E_i' ima vrednost 1.

   ($\Longleftarrow$) Dokažimo da bar jedan od literala x i y ima vrednost 1. Pretpostavimo suprotno: $x = y = 0$, onda je $E_i' = (a + b) \cdot (\bar{a} + b) \cdot (a + \bar{b}) \cdot (\bar{a} + \bar{b})$ uvek netačan. Kontradikcija.

Prethodni postupak se primenjuje za svaku klauzu $E_i, i = 1, \ldots, n$.

Opisana polinomijalna redukcija 2SAT problema na 4SAT ne znači da je i 4SAT rešiv u polinomijalnom vremenu. Štaviše, poznato je da je 4SAT problem $\mathcal{NP}$-kompletan.

6. (a) Da li važi 1SAT $\in \mathcal{P}$?

   (b) Da li važi 1SAT $\in \mathcal{NP}$?

   Rešenje: Da bismo dokazali da je 1SAT $\in \mathcal{P}$, treba da dokažemo da postoji polinomijalni algoritam koji rešava problem 1SAT.

   Koristimo sledeće tvrđenje: Logički izraz $S = L_1 \cdot L_2 \cdots L_n$, gde su L_i literali, je zadovoljiv AKKO ne postoje $i, j \in \{1, \ldots, n\}$ takvi da je $L_i = p$, a $L_j = \bar{p}$, za neko iskazno slovo p.

   Algoritam bi se sastojao iz dva koraka:

   I) Smestimo sve literale izraza S u niz dimenzije n.

   II) Proverimo da li tom nizu postoje elementi takvi da je jedan negacija drugog. Ako postoje, algoritam daje odgovor "NE", a inače daje odgovor "DA".

   Složenost prvog koraka je $O(n)$, a drugog $O(n^2)$, pa je ovaj algoritam polinomijalan, tj. 1SAT $\in \mathcal{P} \subset \mathcal{NP}$.

7. Neka je X proizvoljan problem.

   (a) Da li iz X $\in \mathcal{P}$ i X $\in \mathcal{NP}$ sledi $\mathcal{P} = \mathcal{NP}$?

   (b) Da li iz X $\in \mathcal{P}$ i X $\notin \mathcal{NP}$ sledi $\mathcal{P} \neq \mathcal{NP}$?

   (c) Da li iz X $\in \mathcal{P}$ i X $\notin \mathcal{NP}$ sledi $\mathcal{P} = \mathcal{NP}$?

   (d) Da li iz X $\notin \mathcal{P}$ i X $\in \mathcal{NP}$ sledi $\mathcal{P} \neq \mathcal{NP}$?

   (e) Da li iz X $\notin \mathcal{P}$ i X $\notin \mathcal{NP}$ sledi $\mathcal{P} = \mathcal{NP}$?

   Rešenje:

   (a) Iz X $\in \mathcal{P}$ i X $\in \mathcal{NP}$ ne možemo zaključiti da važi $\forall Y \in \mathcal{NP}$ važi $Y \in \mathcal{P}$. Kada bi X bio $\mathcal{NP}$-težak problem, onda bismo mogli to da izvučemo.

   (b) Iskaz X $\in \mathcal{P}$ i X $\notin \mathcal{NP}$ je netačan za svako X, jer je $\mathcal{P} \subset \mathcal{NP}$, pa je izraz X $\in \mathcal{P} \wedge$ X $\notin \mathcal{NP} \Longrightarrow \mathcal{P} \neq \mathcal{NP}$ je formalno tačan, ali to još uvek ne znači da je $\mathcal{P} \neq \mathcal{NP}$ jer ne postoji X za koje je X $\in \mathcal{P}$ i X $\notin \mathcal{NP}$.

   (c) Kao pod (b), formalno tačan izraz.

   (d) Odgovor je "DA", jer ako postoji element skupa $\mathcal{NP}$ koji ne pripada skupu $\mathcal{P}$, onda se ti skupovi razlikuju (bar za taj element).

   (e) Iz X $\notin \mathcal{P}$ i X $\notin \mathcal{NP}$ ne možemo izvesti $\forall Y \in \mathcal{NP} \Longrightarrow Y \in \mathcal{P}$. Pa iz X $\notin \mathcal{P}$ i X $\notin \mathcal{NP}$ NE sledi $\mathcal{P} = \mathcal{NP}$

8. Dokazati da je sledeći problem $\mathcal{NP}$-kompletan. Dat je neusmereni graf G i prirodan broj k. Utvrditi da li G sadrži podksup od k čvorova čiji je indukovani podgraf aciklički.

   Rešenje:

   - Jeste u klasi $\mathcal{NP}$ jer se za izabrani podskup od k čvorova za polinomijalno vreme može utvrditi da li je indukovani graf aciklički.
   - Svodimo problem 3SAT na dati problem. Neka je $E = c_1 \ldots c_n$ proizvoljni ulaz za 3SAT. Konstruišemo graf sa $4n + 1$ čvorova na sledeći način:
     - Za svaku klauzu c_i dodajemo 4 čvora - 1 odgovara klauzi, a tri njenim literalima.
     - Ta 4 čvora su povezana sa svih 6 grana.
     - Pored toga, svaka dva čvora koja odgovaraju promenljivoj i njenom komplementu povezana su granom i još dodajemo čvor r koji povezujemo sa svim čvorovima pridruženim literalima.

     G ima podskup od $2n + 1$ čvorova čiji je indukovani graf aciklički AKKO je izraz E zadovoljiv.

     Ako je E zadovoljiv onda postoji pridruživanje vrednosti promenljivim tako da je u svakoj klauzi tačan bar jedan literal. Podskup koji se sastoji od čvora r, n čvorova koji odgovaraju klauzama i n čvorova (po 1 iz svake klauze) koji odgovaraju tačnim literalima indukuje aciklički graf. Ovo će preciznije biti stablo sa korenom r koji je povezan sa čvorovima tačnih literala, a oni sa čvorovima klauza.

     Ako postoji podskup S od $2n + 1$ čvorova koji indukuje aciklički graf, onda S sadrži r i po tačno 2 čvora iz svake klauze (ako bi sadržao 3, bili bi povezani).

     Pošto S sadrži r, ne sme istovremeno da sadrži dva čvora koji pripadaju promenljivoj i njenom komplementu!

     Stoga S određuje takvo pridruživanje vrednosti promenljivim koje izraz E čine tačnim.

9. Problem klika je $\mathcal{NP}$-kompletan.

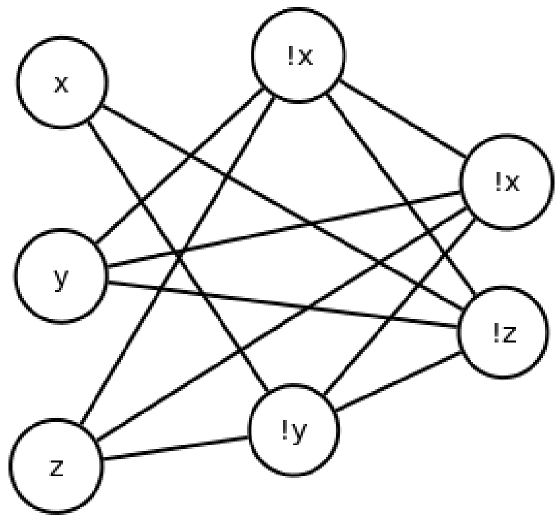
   Problem klika: Dat je neusmeren graf $G = (V, E)$ i prirodan broj k. Ustanoviti da li G sadrži kliku veličine bar k. Klika je takav podgraf grafa G u kome su svi čvorovi međusobno povezani granama iz G.

   Rešenje:

   - Problem klika je u klasi $\mathcal{NP}$, jer za svaki pretpostavljeni k-točlani podskup možemo za polinomijalno vreme da proverimo da li jeste klika.
   - Proizvoljan ulaz za SAT problem svodimo na ulaz za problem klika: Klauzi $(a_1 + a_2 + \ldots a_m)$ dodeljujemo m čvorova koji odgovaraju literalima - tj. graf G imaće po 1 čvor za svaku pojavu bilo koje promenljive. Čvorove iz iste klauze ne povezujemo granama, dok su čvorovi iz različitih klauza skoro uvek povezani, osim kada odgovaraju komplementarnim literalima. Tvrdimo da je izraz $E = E_1 \ldots E_n$ zadovoljiv AKKO graf G ima kliku veličine bar n. Dokaz slično kao prethodni primer.

10. Proveriti zadovoljivost iskazne formule $(x + y + z)(\bar{x} + \bar{y})(\bar{x} + \bar{z})$ svođenjem na problem klike.

    Rešenje: Problem se svodi na pronalaženje klike veličine 3 u sledećem grafu ($\bar{x}$ na slici označen sa !x, isto važi za y i z):



    Jedna klika veličine 3 je {!x, !x, z}, što znači da je početna formula zadovoljiva, a primer valuacije je x = netacno, z = tacno, y je proizvoljno.

11. Razmotrimo sledeći algoritam za utvrđivanje da li graf sa n čvorova ima kliku veličine k. Najpre se generišu skupovi od po tačno k čvorova (ima ih $O(n^k)$), a zatim se za svaki podgraf indukovan tim podskupom proverava da li je kompletan. Zašto ovo nije algoritam polinomijalne složenosti?

    Rešenje: Funkcija n^k jeste polinom po n, ali zavisi eksponencijalno od k. Pošto je k deo predstave problema (i može da dostigne i n), vremenska složenost ovog algoritma nije polinomijalna.

12. Dokazati $\mathcal{NP}$-kompletnost sledećeg problema. Za dati neusmereni graf $G = (V, E)$ i prirodan broj k ustanoviti da li G sadrži kliku veličine k i nezavisni skup veličine k.

    Rešenje:

    - Jeste u klasi $\mathcal{NP}$, jer za svaki od mogućih $\binom{n}{k}\binom{n}{k}$ izbora dva podskupa možemo za polinomijalno vreme da proverimo da li prvi podskup jeste klika, a drugi nezavisan skup.
    - Svodimo problem klika na ovaj problem. Za dati graf $G = (V, E)$ i prirodan broj k (proizvoljan ulaz za problem klika) formiramo graf $G' = (V', E')$ sa skupom čvorova V' koji se dobija od skupa V dodavanjem k izolovanih čvorova. Rešavanjem problem nalaženja k-klike i k-nezavisnog skupa za graf G' pronalazi se klika veličine k u G', koja je istovremeno klika veličine k u G .

    Dakle, ovim je dokazano da je polazni problem $\mathcal{NP}$-kompletan.

13. Zadat je regularni graf G (graf je regularan ako su mu svi čvorovi istog stepena) i prirodan broj k. Dokazati da je problem utvrđivanja da li regularni graf G sadrži kliku veličine k $\mathcal{NP}$-kompletan.

    Rešenje:

- Jeste u klasi $\mathcal{NP}$, jer za svaki k-točlani podskup može u polinomijalnom vremu proverimo da li on čini kliku.
- Svodimo problem klika na ovaj problem. Dati graf $G = (V, E)$ i prirodan broj k (proizvoljan ulaz za problem klika) transformišemo u graf R tako da je graf R još regularan i još treba da pazimo da ne formiramo nove klike. Neka je d' maksimalni od svih stepena čvorova iz G i neka je:

$$d = \begin{cases} d' & d' - parno \\ d' + 1 & d' - neparno \end{cases}$$

  Sad za svaki čvor $v \in V$ koji je stepena $d(v) < d$ dodajemo $d - d(v)$ novih čvorova i sve ih granom povezujemo sa v. Ako je $|V| = n$, ukupan broj novododatih čvorova je

$$dn - \sum_{i=1}^{n} d(v_i) = d_n - 2|E|$$

  a ovo je paran broj, jer smo d konstruisali tako da bude parno.

  Ovim smo dobili da svi stari čvorovi imaju isti stepen jednak d, a nove klike se nisu javile (svaki od novih čvorova je povezan samo sa jednim čvorom). Ali svi novododati čvorovi imaju stepen 1.

  Njihovi stepeni se mogu povećati na d, a da ne dodamo nove klike: Podelimo skup novih čvorova na dva jednaka podskupa (možemo jer ih ima paran broj), a onda svaki novi čvor dodamo iz jednog podskupa povezujemo sa tačno $d - 1$ čvorova iz drugog podskupa (vodeći računa da stepeni novih čvorova ne pređu d). Novi čvorovi indukuju bipartitni graf - a on ne može da sadrži kliku veću od 2.

14. Problem *Pokrivač grana* je $\mathcal{NP}$-kompletan.

    Problem pokrivač grana: Za dati usmereni graf $G = (V, E)$ i prirodni broj k, treba ustanovati da li u G postoji pokrivač grana sa $\leq k$ čvorova. Pokrivač grana grafa G je takav skup u kome je svaka grana grafa G susedna bar jednom od čvorova tog skupa.

    Rešenje:

    - Jeste u klasi $\mathcal{NP}$, jer se za pretpostavljeni podskup od $\leq k$ čvorova lako proverava za polinomijalno vreme da li čini pokrivač grana.
    - Transformisaćemo proizvoljan ulaz za problem klika u ulaz za pokrivač grana tako da je rešenje problem klika "DA"AKKO je rešenje odgovarajućeg problema pokrivač grana "DA".

      Ako je $G = (V, E)$, k ulaz za problem klika i $n = |V|$, mi tvrdimo da se ovaj problem može svesti na problem pokrivač grana sa ulazom $\bar{G}(V, \bar{E})$, $n - k$.

      Neka je $C = (U, F)$ klika u G. Skup čvorova $V \setminus U$ pokriva sve grane u $\bar{G}$, jer u $\bar{G}$ nema grana koje povezuju dva čvora iz U. Dakle, $V \setminus U$ je pokrivač grana za $\bar{G}$, veličine $n - k$.

      Obrnuto, ako je D pokrivač grana u $\bar{G}$, onda u $\bar{G}$ ne može da postoji grana koja povezuje dva čvora iz $V \setminus D$. Dakle, $V \setminus D$ je klika u G. tj. ako u $\bar{G}$ postoji pokrivač grana veličine $n - k$, onda u G postoji klika veličine k.

      Redukcija je izvršena u polinomijalnom vremenu (sve što treba uraditi je konstruisati $\bar{G}$).

15. Dokazati da problem Pokrivač grana ostaje $\mathcal{NP}$-kompletan i ako se uvede ograničenje da svi čvorovi u grafu moraju imati paran stepen.

    Rešenje: Izvršićemo redukciju sa klasičnog problema pokrivač grana. Neka je $G = (V, E)$ proizvoljan neusmeren graf i neka je U podskup čvorova grafa G koji su neparnog stepena.

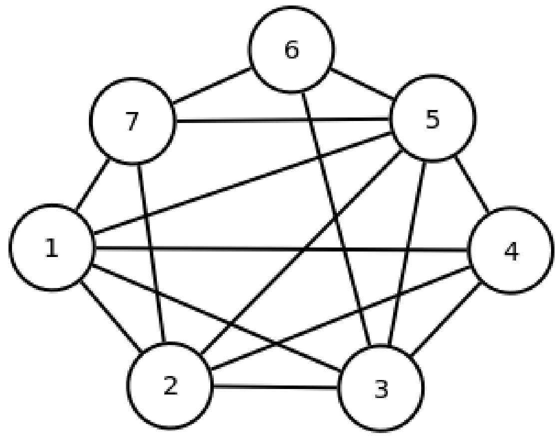
    Dodajemo tri nova čvora: x, y i z koja su međusobno povezana i povezujemo x sa svim čvorovima iz U - time dobijamo novi graf G' čiji su svi čvorovi parnog stepena. (Skup U ima paran broj čvorova. Razmisliti zašto!)

    Graf G' ima pokrivač grana veličine k AKKO graf G ima pokrivač grana veličine $k - 2$.

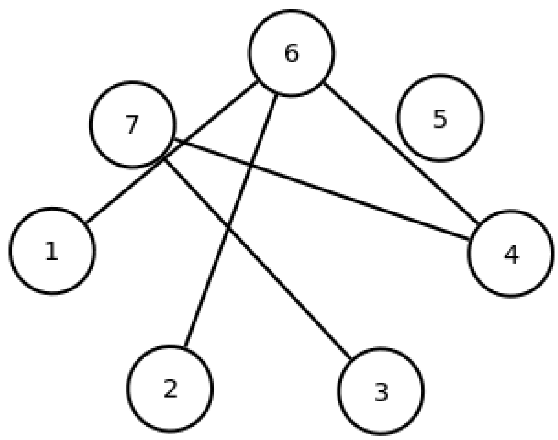
    Za proizvoljni pokrivač grana grafa G veličine $k - 2$ ako dodamo čvorove x i y, dobijamo pokrivač grana grafa G' veličine k.

    Obrnuto, ako u G' postoji pokrivač grana veličine k, onda taj pokrivač mora da sadrži bar 2 od 3 čvora x, y i z (oni su međusobno povezani). Tada je i skup od k čvorova koji se dobija zamenom ta dva čvora čvorovima x i y ne pokrivaju ni jednu granu iz G, njihovim uklanjanjem iz skupa dobijamo pokrivač grana grafa G veličine $k - 2$.

16. Dat je neusmereni graf $G = (V, E)$ sa skupom čvorova $V = \{1, 2, 3, 4, 5, 6, 7\}$ i skupom grana $E = \{$ (1,2), (1,3), (1,4), (1,5), (1,7), (2,3), (2,4), (2,5), (2,7), (3,4), (3,5), (3,6), (4,5), (5,6), (5,7), (6,7) $\}$ (svaka grana navedena je samo jednom). Dokazati da postoji klika veličine 5 svođenjem na problem pokrivač grana. Korišćenjem istog svođenja razmotriti da li postoji klika veličine 6. **Rešenje:** Dati neusmereni graf je predstavljen sledećom slikom:



Komplement ovog grafa je:



Pošto je skup od $7 - 5 = 2$ čvorova $\{6, 7\}$ pokrivač grana u komplementu, onda u početnom grafu postoji klika veličine 5. Pošto ne postoji pokrivač grana u komplementu veličine $7 - 6 = 1$, onda u početnom grafu ne postoji klika veličine 6.

17. Problem 3-obojivost je $\mathcal{NP}$-kompletan (nećemo dokazivati - dokaz u knjizi).

    Ispravno bojenje grafa je takvo pridruživanje boja čvorovima tako da je svakom čvoru pridružena neka boja, a da su susednim čvorovima pridružene različite boje.

    Problem Dat je neusmeren graf $G = (V, E)$, ustanoviti da li se graf G može obojiti sa 3 boje.

    Problem k-obojivost je $\mathcal{NP}$-kompletan za $k \neq 1, 2$, jer je u klasi $\mathcal{NP}$, a na njega se svodi (kao specijalan slučaj) problem 3-obojivosti grafa.

18. Neka je $G = (V, E)$ neusmereni graf čijem je svakom čvoru pridružen neki posao. Dva čvora su povezana granom ako se odgovarajući poslovi ne mogu izvršavati istovremeno. Svaki skup poslova, takvih da bilo koja dva posla iz skupa nisu međusobno povezana, može se izvršiti u jednom koraku. Dokazati da je sledeći problem $\mathcal{NP}$-kompletan: dat je graf $G = (V, E)$ i prirodan broj k, ustanoviti da li se svi poslovi koji odgovaraju čvorovima mogu izvršiti za k koraka.

    Rešenje:

    - Jeste u klasi $\mathcal{NP}$, jer za svako moguće razlaganje skupa od $|V|$ čvorova na k podskupova za polinomijalno vreme se može proveriti da li je to razlaganje na nezavisne skupove.
    - Svodimo problem k-obojivosti grafa na ovaj problem. Proizvoljni algoritam koji rešava ovaj problem može se iskoristiti za k-bojenje grafa G, iz razloga što čvorovi koji odgovaraju istovremeno obavljenim poslovima čine nezavisan skup (između njih nema grana), pa se iz istog razloga mogu obojiti jednom istom bojom.

19. Pokazati da je sledeći problem (poznat kao *pokrivanje skupa*) $\mathcal{NP}$-kompletan. Dat je konačni skup X i familija $\mathcal{F}$ podskupova od X, $X = \cup_{S \in \mathcal{F}} S$. Naći podskup $C \subset \mathcal{F}$ čija je veličina manja ili jednaka k, čiji elementi pokrivaju sve elemente iz X.

    Rešenje:

- Jeste u klasi $\mathcal{NP}$, jer za dati podskup $C \subset \mathcal{F}$ može se u polinomijalnom vremenu proveriti da li je $|C| \leq k$ i da li su svi elementi iz X u nekom od skupova iz C.
- Svodimo problem pokrivač grana na ovaj problem. Za dati graf $G = (V, E)$ i zadato k (ulaz za problem pokrivač grana) definišemo F_v kao skup grana incidentnim sa čvorom $v \in V$. Važi $F_v \subset X$, jer za skup X uzimamo skup svih grana u grafu G, tj. važi $X = E$. Tada nam je ulaz za problem pokrivanje skupa $(X, \mathcal{F}, k)$.

  Dokažimo da važi: Graf G ima pokrivač grana veličine $\leq k$ AKKO postoji pokrivanje skupa X veličine $\leq k$.

  $(\Longrightarrow)$ Neka je S pokrivač grana veličine $\leq k$ grafa G. S odgovara kolekciji C podskupova od X. Ovih podskupova ima $\leq k$ i tvrdimo da ovi podskupovi pokrivaju X. Uočimo element iz X (to je grana $e \in E$). S obzirom da je S pokrivač grana u G, u S se nalazi bar jedan od čvorova grane $e = (x, y)$, tj. $x \in S \vee y \in S$. Stoga, C sadrži bar jedan skup pridružen čvoru od e, tj. $F_x \in C \vee F_y \in C$. $\Rightarrow C$ pokriva e.

  $(\Longleftarrow)$ Neka je C pokrivač skupa X veličine $\leq k$. Neka je S skup pridruženih čvorova. $|S| = |C|$. Uočimo e, postoji skup iz C koji sadrži $e \Rightarrow S$ mora da sadrži bar 1 od čvorova grane e.

20. Problem Hamiltonov ciklus je $\mathcal{NP}$-kompletan (nećemo dokazivati).

    Hamiltonov ciklus u grafu je prost ciklus koji sadrži svaki čvor grafa tačno jednom. Ciklus je prost, ako, sem prvog i poslednjeg čvora se niti jedan drugi čvor ne pojavljuje u putu dva puta.

    Problem: Ustanoviti da li dati graf sadrži Hamiltonov ciklus.

    Može se dobiti redukcijom problema 3SAT ili problema pokrivač grana.