math.e

Hrvatski matematički elektronički časopis

Sokoban je NP --težak

kompjuterske igre složenost algoritama sokoban teorija igara tetris

Autori: Stjepan Požgaj, Mladen Vuković

1 Uvod

Kod raznih igara često je važno pitanje određivanja pobjedničke strategije. No, vrlo često se razmatraju i problemi računarske složenosti u vezi igara (vidi [2]). U ovom članku se razmatra računarska složenost povezana s igrom Sokoban.

Sokoban je poznata stara računalna igra u kojoj igrač (skladištar) mora odgurati sve kutije u skladištu na odgovarajuće pozicije, s tim da odjednom smije gurati najviše jednu kutiju. Osnovnu verziju igre možete isprobati na sljedećoj poveznici: https://sokoban.info/.

U ovom članku dat ćemo dokaze NP-težine dviju varijanti igre Sokoban. Ti se dokazi svode na to da za svaku instancu 3-SAT problema (ili u drugom slučaju problema P-3-SAT) generiramo početnu konfiguraciju igre koja će biti rješiva ako i samo ako je zadana formula ispunjiva. Većina rezultata u ovom članku dio je članka [5].

Osnovne pojmove teorije složenosti algoritama potrebne za razumijevanje ovog članka predstavit ćemo u točki 2. U točki 3 objasnit ćemo standardnu verziju igre Sokoban te još jednu njenu varijantu. Kao prirodno pitanje nameće se što uopće znači da je igra Sokoban NP-teška. To ćemo objasniti u točki 4.

U točki 5 ćemo dati dokaz NP-težine neklasične varijante igre Sokoban. Po našem mišljenju taj je dokaz nešto jednostavniji od dokaza iste tvrdnje za standradnu verziju igre. Dokaz NP–težine standardne verzije igre dan je u točki 6.

U ovoj točki dati ćemo opise osnovnih pojmova iz teorije složenosti algoritama koje koristimo u ovom članku.

2 Osnovni pojmovi teorije složenosti algoritama

Naglašavamo da ćemo dati samo intuitivne opise a ne stroge definicije. Razlog tome je, naravno, veličina teksta članka. Za formalne defincije trebalo bi, primjerice, detaljno definirati Turingove strojeve i sve pojmove u vezi njih. Kolika je to veličina materijala možete vidjeti, primjerice u [12].

U čitavom članku pojam algoritma ćemo koristiti u intutivnom smislu.[1]

Neka je Γ proizvoljan konačan skup. Skup Γ ćemo nazivati alfabet. Svaki konačan niz elemenata alfabeta Γ nazivamo riječ. Skup svih riječi alfabeta Γ označavamo s Γ^*. Ako je $s \in \Gamma^*$ neka riječ tada s $|s|$ označavamo duljinu riječi s. Svaki podskup od Γ^* nazivamo jezik.

U ovom članku razmatrat ćemo samo algoritme koje imaju jedan ulazni podatak (točnije, za zadani alfabet Γ ulazni podatak je neka riječ $s \in \Gamma^*$). Zatim, pretpostavljamo da svaki algoritam za svaki ulazni podatak završi u konačno mnogo koraka te da na kraju kao izlazni podatak daje samo "DA" ili "NE". Razlog takvim ograničenjima je to što promatramo samo tzv. probleme odlučivanja. To su problemi kod kojih nas zanima ima li neki objekt neko zadano svojstvo. Neki primjeri problema odlučivanja su: ispunjivost formula logike sudova, prostost zadanog prirodnog broja i bojanje grafa s tri boje. Problem trgovačkog putnika, problem ruksaka i problem rasporeda su primjeri problema koji nisu problemi odlučivanja. Formalno se problemi odlučivanja definiraju kao odgovarajući jezici. Primjerice, problem PRIMES koji se sastoji od određivanja je li zadani prirodan broj prost, definira se formalno kao $\{n \in \mathbb{N} : n \text{ je prosti broj}\}$.

Vremenska složenost nekog algoritma definira se kao funkcija $f : \mathbb{N} \to \mathbb{N}$, gdje je $f(n)$ maksimalni broj koraka koji algoritam izvrši za svaki ulazni podatak duljine n.[2]

Sasvim analogno se defnira prostorna složenost nekog algoritma (primijetite da je ovdje nužno imati Turingove strojeve kako bi mogli brojati registre koje algoritam koristi).

Reći ćemo da je neki algoritam vremenski polinoman ili samo kratko da je polinoman, ako je njegova vremenska složenost neki polinom. S P označavamo klasu svih problema odlučivanja za koje postoje polinomni algoritmi koji ih rješavaju. Neki primjeri problema koji pripadaju klasi P su sljedeći: ispitivanje relativne prostosti dva prirodna broja, odre đivanje je li graf povezan i egzistencija rješenja sistema linearnih algebarskih jednad\v zbi. Zanimljivo je da je 2004. godine dokazano da problem PRIMES tako đer pripada klasi P (članak o tome je objavljen u najcjenjenijem matematičkom časopisu Annals of Mathematics).

Sasvim analogno definira se klasa PSPACE koja sadrži sve probleme odlučivanja za koje postoji algoritam koji je polinomno prostorno složen.

Sada ćemo opisati nekoliko pojmova iz logike sudova kako bismo mogli definirati problem SAT. Propozicionalne varijable ovdje označavamo s $x_0, x_1, x_2, \ldots$ Ako je F neka formula logike sudova, tada ćemo s $\overline{F}$ označavati njenu negaciju. Literal je svaka propozicionalna varijabla x_i i njena negacija $\overline{x}_i$. Elementarna disjunkcija je disjunkcija literala. Primjerice, $x_2 \vee x_7 \vee \overline{x}_{33} \vee x_8 \vee \overline{x}_{201}$ i $x_4 \vee x_6$ su primjeri elementarnih disjunkcija. Konjunktivna normalna forma (ili kratko: knf) je svaka konjunkcija elementarnih disjunkcija. Neki primjeri knf su

$(x_8 \vee \overline{x}_1 \vee x_4 \vee x_5) \wedge (x_1 \vee x_2)$ i $(x_{13} \vee \overline{x}_{23}) \wedge x_{33} \wedge (x_{23} \vee \overline{x}_4 \vee x_9)$. Knf nazivamo 3-knf ako svaka njena elementaran disjunkcija sadrži tri literala. Problem ispunjivosti logike sudova, koji se kratko označava sa SAT, sastoji se određivanja je li zadana knf ispunjiva, tj. postoji li interpretacija propozicionalnih varijabli tako da je za tu interaprtaciju knf istinita, odnosno formalno ovako:

$$\mathsf{SAT} = \{F : F \text{ je knf za koju postoji interpretacija } I \text{ takva da } I(F) = 1\}.$$

Problem 3-SAT u obzir uzima samo 3-knf. Sasvim analogno se definira problem 2-SAT.
Mi ćemo u ovom članku najviše spominjati klasu problema NP. Jako grubo rečeno klasa NP sadrži sve probleme odlučivanja za koje postoji polinomni algoritam čiji ulazni podatak je par koji uz zadani objekt sadrži i posebni "certifikat". Primjenom "certifikata" algoritam onda odlučuje ima li zadani objekt traženo svojstvo. Pokušat ćemo to objasniti na dva primjera. Problem SAT pripada klasi NP jer očito postoji polinomni algoritam kojemu na ulaz dajemo neku knf F i neku interpretaciju I (to je "certifikat") te onda algoritam određuje vrijedi li $I(F) = 1$ (možete li opisati jedan takav algoritam?). Zatim, problem egzistencije potpuno povezanog podgrafa zadane veličine u nekom grafu također pripada klasi NP. Za taj problem postoji polinomni algoritam koji na ulazu ima zadani graf G, $k \in \mathbb{N}$ i neki podgraf G' od G (ovdje je podgraf "certifikat") te onda algoritam određuje je li G' potpuno povezan i veličine k.

Očito vrijedi P $\subseteq$ NP. Pitanje vrijedi li i obratna inkluzija je centralni problem teorijskog računarstva već više od 50 godina. \v Stovi\v se, problem P vs. NP jedan je od tzv. sedam Milenijskih matemati\v ckih problema.

Neka su Γ_1 i Γ_2 alfabeti. Za neku funkciju $g : \Gamma_1^* \to \Gamma_2^*$ kažemo da je vremenski polinomno izračunljiva ako postoji polinom $p : \mathbb{N} \to \mathbb{R}$ i neki algoritam tako da za svaki $s \in \Gamma_1^*$ algoritam kao izlazni rezulat daje $g(s)$ i za to mu ne treba više od $p(|s|)$ koraka.

Ako imamo dva problema odlučivanja, označimo ih L_1 i L_2, tada je problem L_1 polinomno reducibilan na problem L_2 ako možemo svaku instancu problema L_1 (rješivu i nerješivu) svesti u polinomnom vremenu na neku instancu problema L_2. Budući da ovaj pojam polinomne reducibilnosti bitno koristimo u ovom članku, odlučili smo ipak dati nešto formalniju definiciju. Neka su $L_1 \subseteq \Gamma_1^*$ i $L_2 \subseteq \Gamma_2^*$ dva problema odlučivanja. Kažemo da je problem L_1 polinomno reducibilan na problem L_2, i pišemo $L_1 \leq_p L_2$, ako postoji vremenski polinomno izračunljiva funkcija g tako da za svaki $s \in \Gamma_1^*$ vrijedi sljedeća ekvivalencija:

$$s \in L_1 \text{ ako i samo } g(s) \in L_2.$$

U daljnjem tekstu članka dati ćemo primjere polinomne reducibilnosti.
Za neki problem odlučivanja L kažemo da je NP–težak ako za svaki problem $L' \in$ NP vrijedi $L' \leq_p L$. Zatim, govorimo da je neki problem L jedan NP–potpun problem ako je on NP–težak i još vrijedi $L \in$ NP. Cook i Levin su 1970. godine nezavisno dokazali da je SAT jedan NP–potpun problem. Problem 3-SAT je tako đer NP–potpun. (No, vrijedi 2-SAT $\in$ P.)

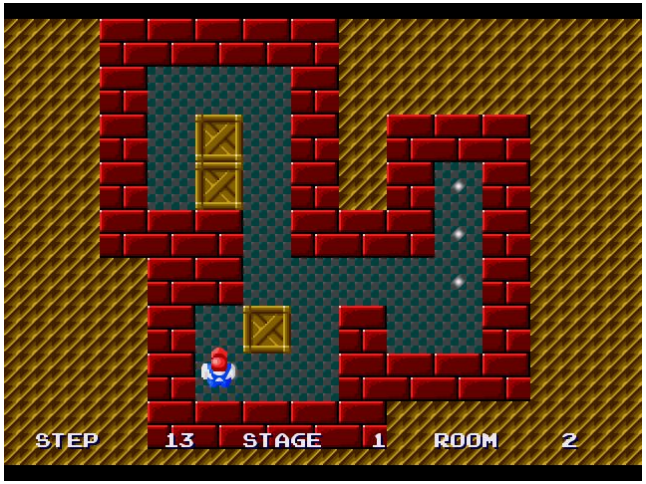
Označimo s Dif problem odre đivanja ima li zadana diofantska jednadžba cjelobrojnih rješenja. Već smo naveli da za taj problem ne postoji algoritam koji ga rješava. Iz toga posebno slijedi $Dif \notin$ NP. No, može se pokazati da je Dif jedan NP–težak problem (vidi [12]).

3 Igra Sokoban

U igri Sokoban igrač upravlja skladištarem koji se može kretati u 4 smjera po pravokutnoj mreži. Na slici 1 prikazan je primjer jednog početnog stanja igre. Skladištar ne može pristupiti poljima na kojima je zid. Može se pomaknuti na slobodno susjedno polje, isto kao i na polje na kojem je kutija (ako je slobodno polje na koje će svojim pomicanjem gurnuti kutiju). U standardnoj verziji igre odjednom može gurnuti najviše jednu kutiju. Zatim, u standardnoj verziji skladištaru nije dozvoljeno vući kutije prema sebi.

U ovom ćemo radu razmatrati još jednu verziju igre. U tom slučaju ćemo pretpostaviti da skladištar istovremeno može gurati do k kutija, gdje je k neki prirodan broj. U toj verziji igre skladišatar smije prema sebi vući najviše jednu kutiju.

Sa Sokoban(k, l) ćemo označiti igru u kojoj skladištar može istovremeno gurati do k kutija i prema sebi vući najviše l kutija, gdje su k i l prirodni brojevi. Tada standardnu verziju igre možemo kratko zapisti kao Sokoban(1, 0), a drugu malo prije opisanu verziju igre možemo kratko označiti sa Sokoban(k, 1), za neki $k \geq 1$.



Slika 1: Prikaz igre Sokoban. Cilj skladištara je sve kutije smjestiti na pozicije označene kružićima.

4 Verzija igre Sokoban kao problem odlučivanja

NP-teški problemi imaju svojstvo da se svaki problem iz klase složenosti NP može polinomno reducirati na njih, dok oni sami ne moraju pripadati klasi NP. Jedan od najpoznatijih načina za dokazivanje da je neki problem NP-težak je polinomno svođenje problema 3-SAT na njega (vidi [12]). To će biti i ideja dokaza u ovome članku. No, prvo moramo objasniti kako uopće neki problem u vezi igre Sokoban možemo promatrati kao problem odlučivanja.

Promatrat ćemo problem odlučivanja SOKOBAN u kojem je pitanje ima li igra Sokoban s nekom početnom konfiguracijom rješenje. Preciznije, za problem SOKOBAN(1, 0) zanima nas možemo li za neku početnu konfiguraciju klasične igre sve kutije staviti na željena mjesta. Za verziju igre Sokoban(k, 1) gledat ćemo nešto lakši problem. U problemu SOKOBAN(k, 1) treba za početnu konfiguraciju odrediti može li skladištar doći do neke određene pozicije (neovisno o pozicijama kutija na kraju igre). Želimo samo spomenuti da je u [5] navedeno da bi se tvrdnja mogla dokazati i za teži problem ali da bi tada alati korišteni u dokazu trebali biti daleko kompliciraniji od ovih koje ćemo mi predstaviti.

5 SOKOBAN(∞, 1)

U ovoj točki dokazat ćemo sljedeći teorem:

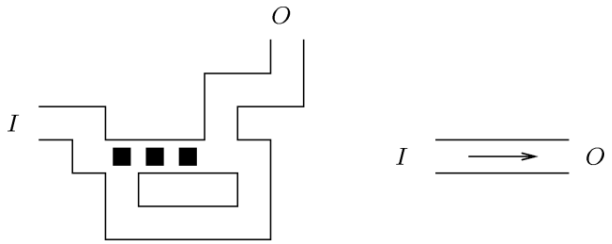
Theorem 1. *Za svaki $k \geq 5$ problem* SOKOBAN*(k, 1) je* NP*-težak problem.*

Kao što smo već najavili, za dokaz ovog teorema koristit ćemo činjenicu da je problem 3-SAT jedan NP-potpun problem (to je jedna od posljedica Cook, Levinovog teorema; vidi [12]). Naš cilj je dokazati da je problem 3-SAT polinomno reducibilan na problem SOKOBAN(k, 1), za svaki $k \geq 5$. To znači da za svaku 3-knf F moramo konstruirati početnu konfiguraciju igre za koju će skladištar moći doći do zadanog polja ako i samo ako je 3-knf F ispunjiva.

U tu svrhu koristit ćemo četiri konstrukcijska elementa: *usmjerivač, porednik, prekidač i usmjereno raskrižje*.

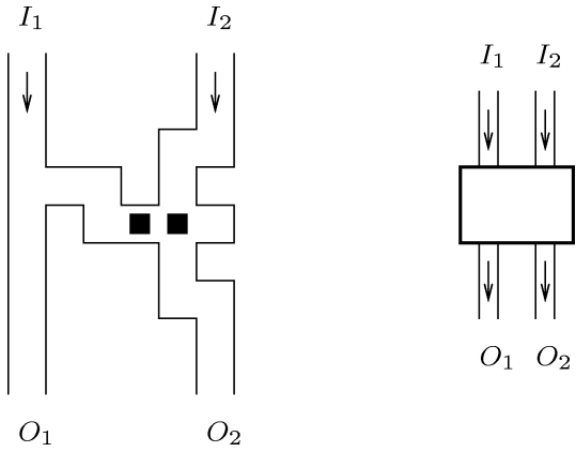
Ako je I oznaka ulaza a O oznaka izlaza u nekom konstrukcijskom elementu, tada s $I \to O$ označavamo proizvoljni (usmjereni) put od I do O. Sada redom opisujemo svaki od prije navedena četiri konstrukcijska elementa.

- *usmjerivač* je prikazan na slici 2.
  Primijetimo da je put $I \to O$ otvoren jer skladištar mora samo dva puta gurnuti tri uzastopne kutije: prvo udesno, nakon čega mora krenuti donjim puteljkom, te nakon toga ulijevo kako bi oslobodio prolaz koji je zatvorio prethodnim guranjem. Nakon izlaska skladištara, konstrukticijski element izgledat će i kao prije njegovog ulaska u koridor. Primijetimo da skladištar ni na koji način ne može proći putem $O \to I$ (iz tog razloga ovaj konstrukcijski element nazivamo usmjerivač).



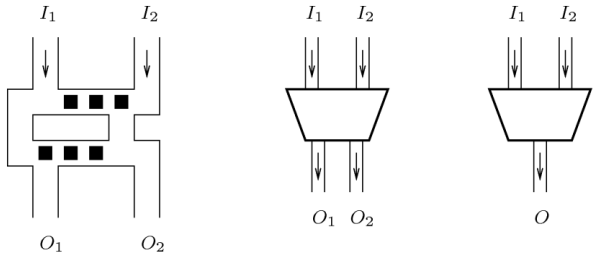
Slika 2: Usmjerivač i njegov shematski prikaz.

- *porednik* je prikazan na slici 3.
  Kod ovog konstrukcijskog elementa je na početku put $I_1 \to O_1$ otvoren, a put $I_2 \to O_2$ je zatvoren. Kako bi skladištar prošao putem $I_2 \to O_2$, prvo mora osloboditi prolaz, a to može samo ako prvo uđe u koridor kroz ulaz I_1, obje kutije gurne do zida i zatim bližu kutiju povuće prema sebi. Primijetimo kako je *usmjerivač* sastavni dio ovog konstrukcijskog elementa. Naime, da njega nema na ulazima I_1 i I_2, skladištar bi mogao ući na ulaz I_1, osloboditi prolaz $I_2 \to O_2$ i izaći iz koridora na mjestu gdje je ušao.



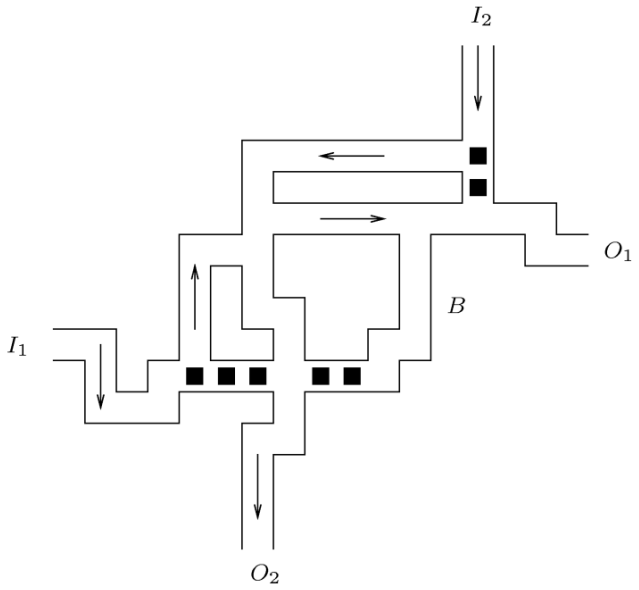
Slika 3: Porednik i njegov shematski prikaz.

- *prekidač* je prikazan na slici 4.
  Njegova funkcija je da omogući samo jednu vrstu prolaza. U objašnjenju ćemo se fokusirati na slučaj desnog shematskog prikaza sa slike 4. Ako skladištar jednom prođe putem $I_1 \to O$ zauvijek će mu se zatvoriti put $I_2 \to O$. Jednako je i obrnutom slučaju.



Slika 4: Prekidač i njegova dva shematska prikaza. Konstrukcijski element prikazan desnim shematskim prikazom dobijemo ako izlaze O_1 i O_2 sa slike spojimo u jedinstveni izlaz O.

- *usmjereno raskrižje* je prikazano na slici 5.
  U ovom konstrukcijskom elementu skladištar na početku može proizvoljan broj puta proći putem $I_1 \to O_1$. Tijekom jednog od prolazaka putem $I_1 \to O_1$ skladištar može odlučiti osloboditi prolaz $I_2 \to O_2$ tako da prođe kroz prolaz označen s B, gurne kutije do zida i zatim najbližu kutiju povuće prema sebi. Nakon što prođe putem $I_2 \to O_2$, prolaz $I_1 \to O_1$ ostat će trajno zatvoren.

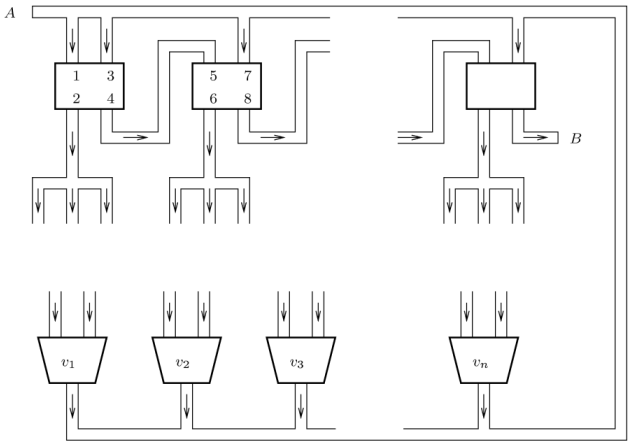


Slika 5: Usmjereno raskrižje

Očito je usmjereno raskrižje najkompliciraniji konstrukcijski element. Kasnije ćemo vidjeti da nam on omogućava da tvrdnju dokažemo i za instance 3-SAT problema čiji inducirani graf nije planaran. O tome ćemo više reći više u dokazu NP-težine standardne verzije igre u kojem nam je planarnost tog grafa ključna.

Sada možemo objasniti i zbog čega je u iskazu teorema uvjet $k \geq 5$. U usmjerenom raskrižju potrebno je da skladištar u jednom trenutku gura barem 5 kutija u nizu. Kada bismo, primjerice, uspjeli konstruirati usmjereno raskrižje u kojem se guraju najviše 4 kutije, tada bi tvrdnja vrijedila i za $k = 4$.

Koristeći prethodno opisane konstrukcijske elemente prelazimo na glavni dio dokaza teorema 1. Neka je $F \equiv C_1 \bigwedge \ldots \bigwedge C_m$ proizvoljna 3-knf s propozicionalnim varijablama $x_1, \ldots, x_n$. Ideja konstrukcije prikazana je na slici 6.



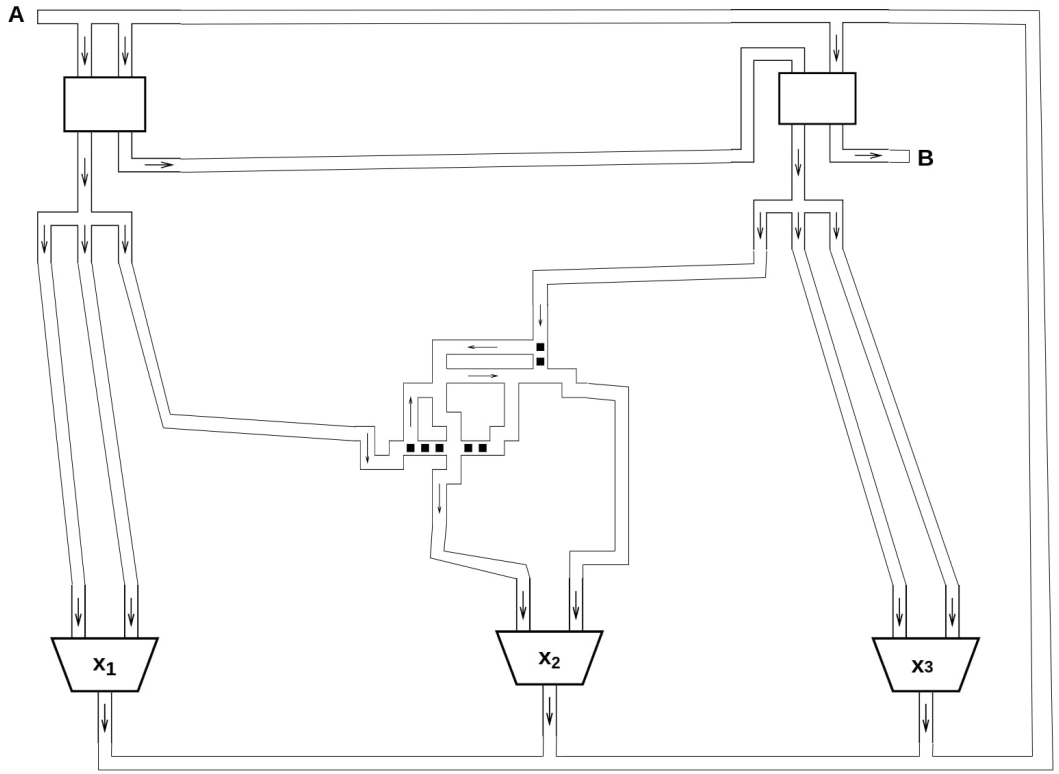
Slika 6: Shematski prikaz konstrukcije početnog stanja igre Sokoban(k, 1) za proizvoljnu 3-knf.

Na slici 6 nalazi se m porednika, koje redom pridružujemo elementarnim disjunkcijama, dok prekidače pridružujemo varijablama. Prvi ulaz i-tog prekidača pridružen je literalu x_i, a njegov drugi ulaz je pridružen literalu $\bar{x_i}$. Zatim, prvi izlaz i-tog porednika vodi do ulaza prekidača koji je pridružen prvom literalu u elementarnoj disjunkciji C_i. Drugi izlaz i-tog porednika pridružen je drugom literalu iz C_i te je treći izlaz pridružen trećem literalu. Kao što smo već spomenuli, usmjerena raskrižja imaju među ostalim funkciju da se putevi mogu sijeći (bez ikakvih garancija da je graf induciran početnom konfiguracijom igre planaran).

Skladištar počinje svoje putovanje na poziciji A i cilj mu je doći do pozicije B. Ovdje je ključno primijetiti da skladištar mora redem proći kroz sve porednike kako bi stigao od ulaza A do izlaza B. Nakon prolaska kroz i-ti porednik skladištar mora barem jednom literalu koji se pojavljuju u elementarnoj disjunkciji C_i pridružiti vrijednost "istina". Prekidači skladištaru onemogućuju da istovremeno nekom literalu i njegovoj negaciju pridruži vrijednost "istina". Time smo dokazali da vrijedi sljedeća ekvivalencija:

formula F je ispunjiva ako i samo ako skladištar može proći put $A \to B$.

Time je teorem potpuno dokazan.



Slika 7: Graf induciran s 3-knf $(x_1 \vee \bar{x_1} \vee \bar{x_2}) \wedge (x_2 \vee x_3 \vee \bar{x_3})$. Ovo je jedan od najjednostavnijih grafova u kojima je barem jedno usmjereno raskrižje.

Sada na jednom primjeru želimo ilustrirati neke detalje iz prethodnog dokaza. Na slici 7 je prikazana početna konfiguracija igre Sokoban(k, 1), $k \geq 5$, koja je pridružena sljedećoj 3-knf:

$$F \equiv (x_1 \vee \bar{x_1} \vee \bar{x_2}) \wedge (x_2 \vee x_3 \vee \bar{x_3})$$

Budući da je prva klauzula od F jednaka $x_1 \vee \bar{x_1} \vee \bar{x_2}$ tada prvi izlaz prvog porednika vodi k prvom ulazu prekidača varijable x_1, a drugi izlaz tog prekidača k njegovom drugom ulazu jer on odgovara literalu $\bar{x_1}$. Isto je tako treći izlaz spojen s prvim ulazom drugog prekidača jer on odgovara literalu x_2, a slično je i za drugi porednik.

Put koji spaja treći izlaz prvog porednika i drugi ulaz drugog prekidača siječe se s putem koji povezuje prvi izlaz drugog porednika i prvi ulaz drugog prekidača pa je neophodno korištenje usmjerenog raskrižja.

6 SOKOBAN(1, 0)

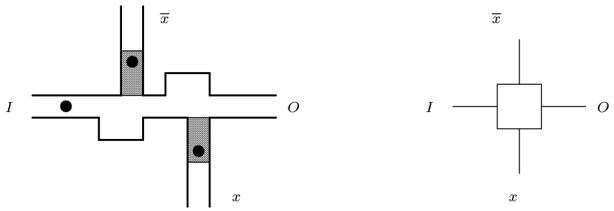
U ovom poglavlju dokazujemo NP-težinu standardne verzije igre. To iskazujemo u sljedećem teoremu.

Theorem 2. *Problem* SOKOBAN*(1, 0) je* NP*-težak.*

U ovoj verziji igre skladištar smije gurati najviše jednu kutiju i zabranjeno mu je vući kutije prema sebi. U ovom ćemo slučaju zahtijevati da na kraju igre sve kutije budu smještene na predviđenim pozicijama. To je različito od prethodno razmatranog slučaja gdje smo samo promatrali može li skladištar doći do svog cilja.

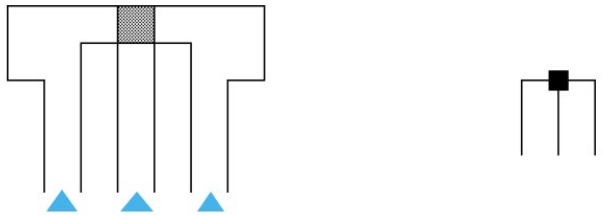
Slično kao i u prethodnom dokazu, ključnu ulogu igraju konstrukcijski elementi koji će biti korišteni prilikom generiranja početne konfiguracije igre. Koristit ćemo dva konstrukcijska elementa: *selektor* i *klauzulu*. Za razliku od otvora konstrukcijskih elemenata iz prijašnjeg dokaza, ovi neće biti usmjereni, već je ulazak i izlazak moguć kroz sve njih. Sada redom opisujemo nove konstrukcijske elemente.

- *selektor* je prikazan na slici 8.
  U ovom konstrukcijskom elementu nalaze se tri kutije i četri pozicije na kojima se na kraju igre moraju nalaziti kutije. Uz ulazni i izlazni otvor u konstrukcijskom elementu se nalaze još dva otvora označena s x i $\overline{x}$ čiju ćemo funkciju objasniti naknadno.



Slika 8: Selektor i njegov shematski prikaz. Kružići označavaju mjesta na kojima su pozicionirane kutije, a osjenčana polja predstavljaju pozicije do kojih skladištar mora dogurati kutije.

- *klauzula* je prikazana na slici 9.
  Ovaj konstrukcijski element sadrži tri otvora i jednu poziciju namijenjenu za smještanje kutije.



Slika 9: Klauzula i njezin shematski prikaz. Isto kao u prethodnom konstrukcijskom elementu, osjenčani kvadratić predstavlja mjesto do kojeg skladištar mora dogurati točno jednu kutiju. Plavim trokutićima su označeni otvori ovog konstrukcijskog elementa.

Osnovna ideja dokaza teorema 2 je slična dokazu teorema 1 gdje smo dokazali da je problem 3-SAT polinomno reducibilan na svaki problem SOKOBAN(k, 1), gdje je $k \geq 5$. No, kao što smo već napomenuli, u ovom ćemo slučaju promatrat ćemo samo instance problema 3-SAT čiji je inducirani graf **planaran**. Prvo ćemo definirati pojam induciranog grafa (sa zadanom formulom), a onda ćemo definirati jednu važnu verziju problema SAT.

Neka je $F \equiv C_1 \bigwedge \ldots \bigwedge C_m$ neka 3-knf s propozicionalnim varijablama $x_1, \ldots, x_n$. Neka je skup čvorova grafa $G_F = (V, E)$, koji je induciran formulom F, jednak sljedećem skupu:

$$V = \{C_1, C_2, \ldots, C_m\} \ \cup \ \{x_1, \ldots, x_n\}.$$

Zatim, neka je skup bridova grafa G zadan ovako:

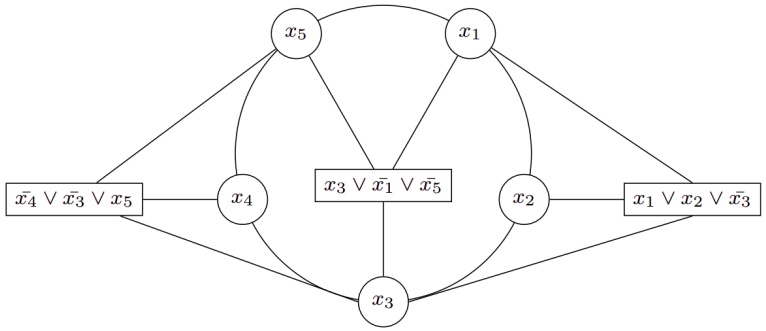
$$E \quad = \quad \{(x_i, x_{i+1}) \mid 1 \leq i < n\} \ \cup \ \{(x_n, x_1)\}$$

$$\cup \{(x_j, C_i) \mid x_j \in C_i \ \text{ ili } \ \bar{x_j} \in C_i, \ i \in \{1, \ldots, m\}, \ j \in \{1, \ldots, n\}\}$$

Dakle, čvorovi grafa su elementarne disjunkcije C_i i varijable formule F. Varijable su bridovima spojene u jedan veliki ciklus, a svaka elementarna disjunkcija C_i je spojena s onim varijablama koje se, ili njihova negacija, pojavljuju u njima.

Sada želimo na jednom primjeru ilustrirati kako izgleda graf koji je induciran zadanom formulom. Na slici 10 prikazan je planarni graf sljedeće 3-knf:

$$\left(x_1 \vee x_2 \vee \bar{x_3}\right) \wedge \left(x_3 \vee \bar{x_1} \vee \bar{x_5}\right) \wedge \left(\bar{x_4} \vee \bar{x_3} \vee x_5\right)$$



Slika 10: Graf induciran s 3-knf.

Označimo s P-3-SAT problem u kojem se razmatraju samo instance problema 3-SAT za koje je inducirani graf planaran. Ovaj problem je također NP-potpun (vidi [??]).

Sada dokazujemo da je problem P-3-SAT polinomno reducibilan na problem SOKOBAN(1, 0). U tu svrhu za proizvoljnu 3-knf F čiji je inducirani graf planaran, konstruiramo početnu konfiguraciju igre kao na slici 11. Takva početna konfiguracija sadrži točno n selektora koji redom odgovaraju varijablama formule F. Na slici 11 je m klauzula koje pridružujemo elementarnim disjunkcijama C_i.

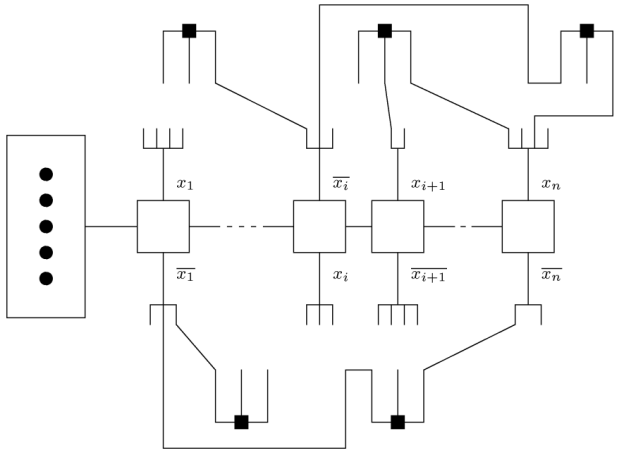
Selektori su spojeni onim klauzulama u kojima se pripadne varijable ili njihove negacije nalaze. Preciznije, ako je $x_j \in C_i$ tada će j-ti selektor biti spojen s klauzulom koja odgovara elementarnoj disjunkciji C_i kroz otvor x_j, a ako je $\bar{x_j} \in C_i$ biti će spojen kroz otvor $\bar{x_j}$.

Primijetimo da se u danoj početnoj konfiguraciji nalazi još jedan konstrukcijski element koji do sada nismo opisali. Taj novi konstrukcijski element nazivat ćemo *rezervoar*. U njemu su na početku smještene kutije. Naime, u svakom od n selektora su na početku tri kutije i četiri mjesta namijenjena za kutije, a u svakoj od m klauzula jedno mjesto namijenjeno za kutije, što znači da fali $n + m$ kutija kako bi na kraju igre svaka kutija bila na točno jednom namijenjenom mjestu.

Preostalo je dokazati da je formula F ispunjiva ako i samo ako za početnu konfiguraciju koju smo iz nje izgradili skladištar može smjestiti sve kutije na zadane pozicije. Neka se na početku skladištar nalazi u rezervoaru.

Pretpostavimo prvo da je F ispunjiva formula. Tada postoji interpretacija I za koju je $I(F) = 1$. Skladištar kreće iz rezervoara i prolazi redom po selektorima. Svaku kutiju koja mu stoji na putu mora staviti u otvor koji pripada literalu l_i za koji je $I(l_i) = 0$. Na taj način blokira taj otvor.

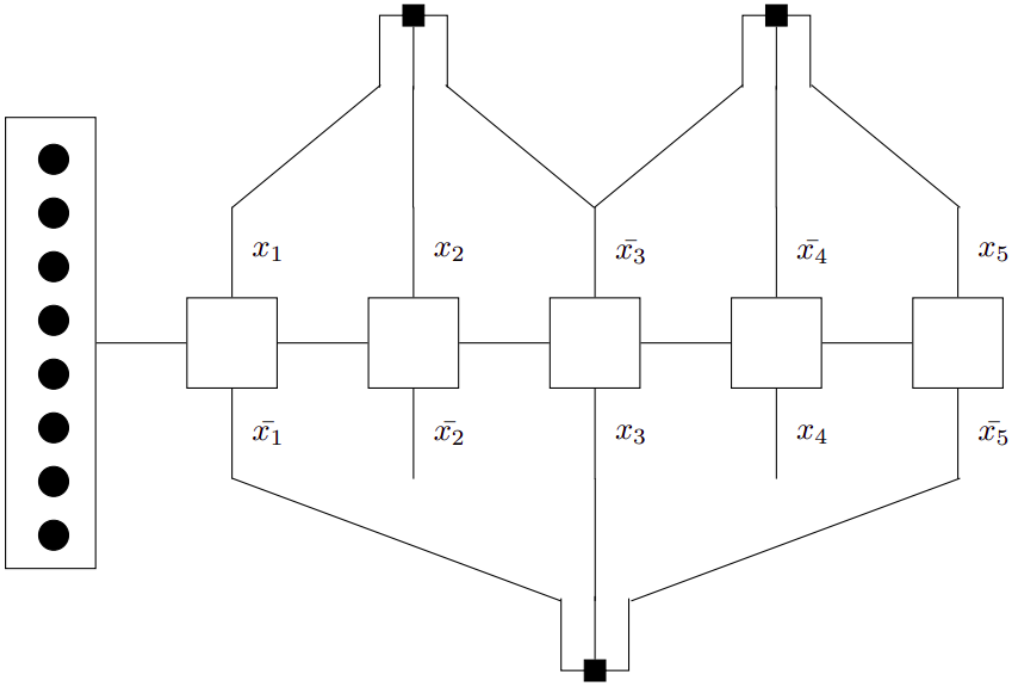
Nakon toga mora dogurati kutije do slobodnih mjesta u svakoj od klauzula. To je moguće zbog toga što su otvori selektora koji pripadaju literalima l_i za koje vrijedi $I(l_i) = 1$ ostali otvoreni i zato što, zbog ispunjivosti formule F, za svaku klauzulu postoji barem jedan literal za koji je $I(l_i) = 1$. Nakon što popuni tražena mjesta unutar klauzula, skladištar pomoću preostalih kutija iz rezervoara mora blokirati preostale otvore selektora. Tako je uspio sve kutije dovesti na željene pozicije.



Slika 11: Shematski prikaz konstrukcije početnog stanja igre Sokoban(1, 0) za proizvoljnu 3-knf čiji je inducirani graf planaran.

Pretpostavimo sada da igra s početnom konfiguracijom generiranom formulom F rješiva. Cilj nam je pronaći interpretaciju I za koju je $I(F) = 1$. Kojim god poretkom skladištar smještao kutije na željene pozicije, kroz selektore mora proći redom od prvog do posljednjeg ulazeći u njih kroz glavni ulaz I. Naime, u selektor ne smije ući kroz otvor x_i ili $\bar{x_i}$ jer bi u tom slučaju kutiju s tog ulaza gurnuo u kut iz kojeg ga više ne može izvaditi. Skladištar također ne smije ući u selektor kroz glavni izlaz O jer bi prije toga morao ući u neki od selektora kroz otvor x_i ili $\bar{x_i}$. Prolazeći redom kroz selektore, skladištar mora kutiju koja mu je na putu staviti u neki od otvora x_i ili $\bar{x_i}$, jer bi gurajući kutiju prema sljedećem selektoru zablokirao svoj prolaz u trenutku kad kutija koju gura dođe do kutije iz sljedećeg selektora. Ako u i-tom selektoru skladištar prvo blokira otvor x_i, definirat ćemo $I(x_i) = 0$, a ako prvo blokira otvor $\bar{x_i}$ tada ćemo definirati $I(x_i) = 1$. Očito na taj način dobivamo interpretaciju I za koju je $I(F) = 1$, što smo i trebali pokazati.

Na slici 12 je prikazana početna konfiguracija igre Sokoban(1, 0) pridružena formuli sa slike 10.



Slika 12: Početna konfiguracija dobivena iz 3-knf $(x_1 \vee x_2 \vee \bar{x_3}) \wedge (x_3 \vee \bar{x_1} \vee \bar{x_5}) \wedge (\bar{x_4} \vee \bar{x_3} \vee x_5)$, koja je korištena i u slici 10 gdje je pokazano da je inducirani graf te formule planaran. Sve klauzule koje su u planarnom grafu sa slike 10 unutar ciklusa koji povezuje varijable potrebno je staviti s jedne, a one koje su izvan ciklusa s druge strane pravca na kojem su selektori. Tada je moguće spojiti klauzule i selektore linijama koje se ne presjecaju. Skladištar se na početku nalazi u rezervoaru.

7 Zaključak

U ovom članku dani su dokazi NP-težine dviju verzija igre Sokoban.

Culberson je dokazao da je standardna verzija igre Sokoban PSPACE-potpuna (vidi [3]). To je dokazano i za stratešku društvenu igru *Scotland Yard* (vidi [10]). Takva, dakako, ne mora biti svaka igra. Primjerice, za problem odlučivanja pobjednika u popularnoj igri *GO* dokazano je tek da je PSPACE-težak, dok je pripadnost klasi PSPACE otvoren problem. (vidi [8]).

Za neke druge klasične igre kao što su *Minesweeper* (vidi [6]), *potapanje brodova* (vidi [9]) i *tetris* (vidi [4]) je dokazano i da su NP-potpune.

No za razne verzije igre Sokoban je još otvoreno pitanje jesu li NP-potpune.

Bibliografija

[1] Čačić, Vedran: Komputonomikon, Izračunljivost za računarce. PMF-Matematički odsjek, 2022.

[2] Berlekamp, Elwyn R., John H. Conway ,Richard K. Guy:Winning ways for your mathematical plays. Vol. 2. Academic Press , London, 1982 , 0-12-091152-3; 0-12-091102-7 .

[3] Culberson , Joseph C. Sokoban is PSPACE-complete Sokoban is PSPACE-complete . Proceedings of the International Conference on Fun with Algorithms , 65–76, 1998.

[4] Demaine , Erik D. , Susan Hohenberger David Liben-Nowell Tetris is hard, even to approximate. Warnow , Tandy Binhai Zhu : Computing and Combinatorics , 351–363, 2003 , 978-3-540-45071-9 .

[5] Dor , Dorit Uri Zwick Sokoban and other motion planning problems. Computational Geometry , 13(4):215–228, 1999 , 0925-7721 . https://www.sciencedirect.com/science/article/pii/S0925772199000176 .

[6] Kojić , Vedran Minesweeper problem je NP-potpun. Math.e , 12:0–0, 2007. http://e.math.hr/old/minesweeper/index.html.

[7] Lichtenstein1982PlanarFA} Lichtenstein , David Planar formulae and their uses. SIAM Journal of Computation , 11:329–343, 1982.

[8] Lichtenstein , David Michael Sipser Go is polynomial-space hard. Journal of ACM , 27:393–401, 1980.

[9] Sevenster , Merlijn Battleships as a decision problem. ICGA Journal , 27:142–149, 2004.

[10] Sevenster , Merlijn The Complexity of Scotland Yard , 209–246. Amsterdam University Press , 2007 , 9789053563564 . http://www.jstor.org/stable/j.ctt45kdbf.12 , 2022-06-24

[11] Vuković , Mladen Izračunljivost, predavanja i vježbe . PMF-Matematički odsjek , 2009.

[12] Vuković , Mladen Složenost algoritama, predavanja i vježbe . PMF-Matematički odsjek , 2019. https://www.pmf.unizg.hr/_download/repository/SA-skripta-2019.pdf .

[1]Pitanje što je algoritam još uvijek je aktualno. U prvi tren prirodno se je zapitati zašto uopće treba definirati formalno pojam algoritma. Potreba za formalnom definicijom javila se tridesetih godina prošlog stoljeća kada se željelo pokazati da za neke probleme ne postoje algoritmi koji ih rješavaju. Alonso Church je prvi dokazao da problem ispitivanja valjanosti formule logike prvog reda nije algoritamski rješiv (za svoj dokaz nije koristio Turingove strojeve već je definirao posebnu teoriju koja se naziva λ račun koji je teorijska osnova danas vrlo važnog funkcijskog programiranja). Godine 1970. Yuri Matiyasevic dokazao je da ne postoji algoritam koji bi za proizvoljnu diofantsku jednadžbu odredio ima li ona cjelobrojno rješenje. Pitanjem formalne definicije algoritma i problemima u vezi toga, bavi se dio matematičke logike i teorijskog računarstva koji se naziva *Izračunljivost* (eng. *Computability*). Na diplomskom studiju *Računarstvo i matematika* na Matematičkom odsjeku PMF-a u Zagrebu postoji kolegij pod naslovom *Izračunljivost*. Više u vezi izračunljivosti možete pronaći u [1] i [11].
[2]Već smo istaknuli da je uz svaki algortiam nužno zadati pripadni alfabet Γ. Budući da je skup Γ konačan, tada za svaki $n \in \mathbb{N}$ postoji samo konačno mnogo riječi $s \in \Gamma^*$ duljine n. Zatim, pretpostavka da svaki algoritam za svaki ulazni podatak obavezno staje u konačno mnogo koraka garantira da je dobro definiran broj $f(n)$.

ISSN 1334-6083