

13. Előadás

Előadó: Hajnal Péter

Jegyzetelő: Hajnal Péter

2011. május 10.

1. Összeszámlálási problémák bonyolultsága

1.1. Példák

Lássunk néhány példát összeszámlálási feladatokra.

Példa. Adott egy G gráf, hány feszítőfája van? A megfelelő függvény neve legyen #SPANNING-TREE.

Példa. Adott egy G gráf, hány Hamilton köre van? A megfelelő függvény neve legyen #HAMILTON.

Példa. Adott egy φ hálózat, hány kielégítő kiértékelése van a változóinak? A megfelelő függvény neve legyen #HÁLÓZAT-SAT.

Példa. Adott egy CNF formula, hány kielégítő kiértékelése van a változóinak? A megfelelő függvény neve legyen #SAT.

Példa. Adott egy 3CNF formula, hány kielégítő kiértékelése van a változóinak? A megfelelő függvény neve legyen #3SAT.

Példa. Adott egy G gráf, hány teljes párosítása van? A megfelelő függvény neve legyen #PÁROSÍTÁS.

Példa. Adott egy G egyszerű páros gráf, hány teljes párosítása van? A megfelelő függvény neve legyen #PÁROS-GRÁF-PÁROSÍTÁS.

Példa. Adott $M_{n \times n}$ $\{0, 1\}$ -mátrix (vagy \mathbb{Z} -mátrix, vagy \mathbb{Q} -mátrix). Határozzuk meg

$$\det M := \sum_{\pi \in S_n} \prod_{i=1}^n (-1)^{\text{sign}(\pi)} M_{i, \pi(i)}$$

értékét (az M mátrix determinánsa). A megfelelő függvény neve legyen DETERMINÁNS.

Példa. Adott $M_{n \times n}$ természetes számokat tartalmazó mátrix. Határozzuk meg

$$\text{per} M := \sum_{\pi \in S_n} \prod_{i=1}^n M_{i, \pi(i)}$$

értékét (az M mátrix permanense). A megfelelő függvény neve legyen PERMANENS.

Példa. Adott $M_{n \times n}$ $\{0, 1\}$ -mátrix. Határozzuk meg $perM := \sum_{\pi \in S_n} \prod_{i=1}^n M_{i\pi i}$ értékét (az M mátrix permanense). A megfelelő függvény neve legyen 0-1PERMANENS.

Észrevétel. Néhány fenti feladatról érezzük, hogy nehéz. $3SAT$ egy \mathcal{NP} -teljes probléma: azt kérdezi, hogy a kielégítő értékelések száma 0 vagy nem. A pontos összeszámlálás nyilván nehezebb.

Más esetben kezelhető problémákat látunk. A determinánsra már Gauss ismert hatékony kiszámítási módszert. (A Gauss-elimináción alapuló determináns kiszámító algoritmus analízise munkás. Számolnunk kell a műveletek során a számok hosszának változásával az alpműveletek Turing-gép implementációjával. Ha ezt a (numerikus analízis és bonyolultsáselemélet határán lévő) feladatot valaki elvégzi, akkor azt kapja, hogy a DETREMINÁNS függvény polinom időben kiszámolható.)

#SPANNING-TREE visszavezethető a determináns kiszámításra. Ez az MSc-s Diszkrét matematika kurzusban látott Kirchoff-tétel következménye.

Néhány közbülső problémánál azonban tanácstalanok vagyunk: #PÁROS-GRÁF-PÁROSÍTÁS esetén a teljes párosítások számának 0/nem-0 voltának eldöntése BSc-s Kombinatorika anyag: a magyar-módszer alapján polinom időben megoldható. A pontos összeszámlálás azonban problémásabb.

A permanens formulája egyszerűbb mint a determinánsé (hiányoznak a váltakozó előjelek. Viszont a Gauss-elimináció nem működik.

Az azonban könnyen látható, hogy #PÁROS-GRÁF-PÁROSÍTÁS és a PERMANENS probléma ekvivalens: Egy egyenlő színosztályokkal ($|A| = |F| = n$) rendelkező egyszerű páros gráfban az alsó/felső csúcsok szomszédsági mátrixában a kifejtési tagok pontosan a teljes párosításoknak felelnek meg, a mátrix permanense éppen a teljes párosítások száma.

1.2. Összeszámlási problémákkal kapcsolatos nyelvosztályok

Emlékeztető. Egy nem-determinisztikus Turing-gép ω inputján a futásai halmaza egy gyökeres fában foglalható össze. A fa levelei ELFOGAD vagy ELVET címkével vannak címkézve. Egy futás egy gyökér-levél út. Ezeket az utakat/futásokat a levél címkéje két kategóriába sorolja.

Emlékeztető. A T nem-determinisztikus Turing-gépről tegyük fel, hogy polinomiális, azaz létezik $t(x) \in \mathbb{R}[x]$, ami egy korrekt futásának hosszát becsüli. Minden polinom szép idő-függvény, így feltehető, hogy gépünk ω -n legfeljebb $t(|\omega|)$ lépést tesz. Azaz a tanúság „levágható” úgy, hogy $t(|\omega|)$ karaktert tartalmazzon. Így az összfutások száma, felülről becsülhető $|\Sigma_{\text{tanú}}|^{t(n)}$ -nel.

Definíció.

$$\#P = \{f : \Sigma^* \rightarrow \mathbb{N} \mid \exists T \text{ polinomiális nem-determinisztikus Turing-gép,} \\ \text{hogy } \omega\text{-n az elfogadó futások száma} = f(\omega)\}$$

$$\mathcal{MAJ}\text{-P} = \{L \subseteq \Sigma^* \mid \exists T \text{ polinomiális nem-determinisztikus Turing-gép,} \\ \text{hogy } \omega \in L \text{ pontosan akkor teljesül, ha } \omega\text{-n} \\ \text{az elfogadó futások száma} > \text{ az elvető futások száma}\}$$

$\oplus\mathcal{P} = \{L \subseteq \Sigma^* \mid \exists T \text{ polinomiális nem-determinisztikus Turing-gép,}$
hogy $\omega \in L$ pontosan akkor teljesül, ha
 ω -n páros sok elfogadó futás van}

$\mathcal{MAJ}\text{-}\mathcal{P}$ szokásos jele \mathcal{PP} . Szokták véletlen gépként is értelmezni: akkor fogad el, ha az elfogadás valószínűsége több mint $1/2$. Ennek az elfogadó szabálynak azonban nincs gyakorlati haszna. Az új elnevezéssel inkább azt hangsúlyozzuk, hogy a valódi jelentősége az osztálynak az, hogy az összeszámlálási problémákat fogja meg.

Az első osztály ami valóban függvényeket tartalmaz (ez a klasszikus összeszámlálási probléma formalizmusa: bizonyos struktúrákhoz rendel egy elemszámot, a megszámlálendő objektumok számát). A második kettő osztály szorosan kapcsolódik az elsőhöz, de ezek már nyelvosztályok. Ezen kettő szimpatikus a bonyolultságelmélet szempontjából, ahol alapvetően nyelvosztályokkal foglalkozunk.

A fenti feladatok mindegyike $\#\mathcal{P}$ -beli. Egyetlen kivétel a DETERMINÁNS. Ez negatív értéket is felvehet, így ekkor elfogadó számítási utak számaként nem interpretálható. Ennek ellenére a DETERMINÁNS függvény

$$\det_o M := \sum_{\pi \in S_n, \text{sign}(\pi)=0} \prod_{i=1}^n M_{i, \pi(i)}$$

és

$$\det_1 M := \sum_{\pi \in S_n, \text{sign}(\pi)=1} \prod_{i=1}^n M_{i, \pi(i)}$$

két „fele” már $\#\mathcal{P}$ -beli, ha a mátrix elemeinek természetes számokat engedünk meg (ezek neve legyen DETERMINÁNS_0 és DETERMINÁNS_1).

Mielőtt elkezdünk bizonyítani egy fontos megjegyzést teszünk.

Megjegyzés. Feltettük, hogy a tanú szalag ω input mellett $t(|\omega|)$ tanúkaraktert tartalmaz. A futások száma azonban nem szükségszerűen $|\Gamma|^{t(|\omega|)}$ (Γ a tanúszalag ábécé-je). A géppel azonban elolvastathatjuk az összes karaktert. Ekkor problémák keletkezhetnek.

Ezzel lelassulhat gépünk: Ez nem baj, polinomiális idejű marad.

Az elfogadó futások száma, viszonya az elvető futások számához képest, az elfogadó futások számának paritása változhat. Ez sem okoz gondot a következő megfontolások után.

Ha leálló állapotba jutunk, akkor olvassuk el a többi tanúkaraktert és csak akkor fogadjuk el, ha eredetileg elfogadó volt leállásunk és utána a $000 \dots 00$ volt a tanúszalag eredetileg nem olvasott része (feltesszük, hogy $0 \in \Gamma$). Ezzel gépünk futása *standard* lesz: minden tanúkaraktert elolvas és az elfogadó futások száma nem változik (így ennek paritása sem). Ezzel beláttuk, hogy *standard* nem-determinisztikus gépekkel dolgozva is ugyanahhoz a $\#\mathcal{P}$, $\oplus\mathcal{P}$ osztályhoz jutunk.

$\mathcal{MAJ}\text{-}\mathcal{P}$ esetén más megfontolás szükséges (most $0, 1 \in \Gamma$ feltevessel élünk): Ha leálló állapotba jutunk, akkor olvassuk el a többi tanúkaraktert (erről feltehetjük, hogy legalább egy ilyen van). Akkor fogadjunk el, ha 1 volt a tanúszalag első, eredetileg nem olvasott karaktere. Akkor vessük el, ha 0 volt a tanúszalag első, eredetileg nem olvasott karaktere. Ezzel gépünk futása *standard* lesz: minden tanúkaraktert elolvas és az elfogadó/elvető futások száma kiegyensúlyozódik. Az eredeti helyzet

visszaállításához a következő módosítást tehetjük: Akkor is fogadjunk el, ha eredetileg elfogadó volt leállításunk és utána a 000...00 volt a tanúszalag eredetileg nem olvasott része (persze ezzel a korábbi elvető leállításunkra vonatkozó szabályunkat felülírtuk). Illetve akkor is vessünk el, ha eredetileg elvető volt leállításunk és utána a 111...11 volt a tanúszalag eredetileg nem olvasott része. Így az eredeti futásoknak egy olyan futáshalmaz felel meg, amiben az elfogadó/elvető futások száma között pontosan kettő a különbség annak megfelelően, hogy eredetileg hogy állt le a gép. Ezzel gépünk futása standard lesz: minden tanúkaraktert elolvas és az elfogadó futások nagyságrendi viszonya az elvető futások számához képest nem változik.

1.3. # \mathcal{P} -teljes problémák

Definíció. Egy $\tau \in \Sigma^* \rightarrow \mathbb{N}$ függvény # \mathcal{P} -teljes függvény, ha

- (i) $\tau \in \#\mathcal{P}$,
- (ii) minden $f \in \#\mathcal{P}$ -hez ($f : \Gamma^* \rightarrow \mathbb{N}$) létezik R polinom idejű redukciós algoritmus (ez $R : \Gamma^* \rightarrow \Sigma^*$ függvényt számol ki), amelyre $\tau(R(\omega))$ -ból kiszámolható $f(\omega)$ polinom időben.

Korábban már megjegyeztük, hogy ha egy összeszámlálási probléma eredménye olyan, hogy 0 mihioltának eldöntése \mathcal{NP} -teljes, akkor egy nehéz problémával állunk szemben. Mostmár ezt matematikailag pontosabban és erőteljesebben is megfogalmazhatjuk.

1. Tétel. #HAMILTON, #SAT, #3CNF # \mathcal{P} -teljesek.

Bizonyítás. Nem bizonyítjuk. A megfelelő döntési problémák \mathcal{NP} -teljesek. Azaz egy \mathcal{NP} Turing-gépre az eldöntési probléma redukálható egy HAMILTON/SAT/3SAT problémára. A redukció lényege, hogy a T által elfogadandó inputok hamaza HAMILTON körrel rendelkező gráfba/kielégíthető CNF formulába/kielégíthető 3CNF formulába képződjenek, illetve a T által elvetendő inputok hamaza HAMILTON körrel nem rendelkező gráfba/kielégíthetetlen CNF formulába/kielégíthetetlen 3CNF formulába képződjenek. A klasszikus redukciók vizsgálata azt mutatja, hogy azok vagy eleve bijekciók vagy könnyen módosíthatók úgy, hogy a redukciótól elvártakat egy bijekció valósítsa meg. Ennek ellenőrzését, a szükséges ötletek megkeresését az érdeklődő hallgatóra bízjuk. ■

Példaink sora végén azonban párosítási problémákkal találkozunk. Ezek döntési megfelelői hatékony algoritmusokkal megoldhatók. Ez azonban az összeszámlálási feladatokról nem mond semmit. Talán meglepő, hogy a fenti probléma is # \mathcal{P} -teljes.

2. Tétel (Valiant). #PÁROS-GRÁF-PÁROSÍTÁS és a PERMANENS probléma # \mathcal{P} -teljes.

Ennek bizonyítás időigényes. Nem végezzük el, nem vizsgaanyag. Az érdeklődő hallgató a jegyzet appendix-ében megtalálja.

1.4. $\#P$ ereje

Észrevétel. Hogyan viszonyul két olyan polinomiális gép ereje, amelyek közül az egyik hozzáférhet egy $\mathcal{MAJ-SAT}$ orákulumhoz, a másik egy $\#SAT$ orákulumhoz? Erejük megegyezik: $\#SAT$ orákulumból $\mathcal{MAJ-SAT}$ kérdések triviálisan megválaszolhatóak. A fordított viszony bináris kereséssel oldható meg.

Jelölés. $\mathcal{P}^{\mathcal{MAJ-P}} = \mathcal{P}^{\#P} (= \mathcal{P}^{\mathcal{P}^{\mathcal{P}}})$.

Marad a kérdés: hogyan viszonyul a $\mathcal{P}^{\mathcal{MAJ-P}} = \mathcal{P}^{\#P}$ nyelvosztály a korábbi osztályainkhoz? Erről a következő tétel azt mondja, hogy az új osztály meglepően erős. A „meglepő” jelző azért jogos, mert az 1989-ben megjelent eredmény nem várt nagy áttörés volt a bonyolultságelméletben. 1998-ban Seinosuke Toda Gödel-díjat kapott ezen cikkéért.

3. Tétel (Toda-tétel). $\mathcal{P}^{\mathcal{MAJ-P}} \supseteq \mathcal{PH}$

A bizonyítás meghaladja ezen kurzus kereteit.

1.5. $\oplus P$ ereje

Az összeszámlálási problémák bonyolultságának tárgyalását azzal zárjuk, hogy a paritási probléma (habár bizonyos értelemben könnyebb mint az általános összeszámlálás) mégis nehéz.

4. Tétel (Valiant–Vazirani-tétel).

$$\mathcal{NP} \subset \mathcal{RP}^{\oplus P}$$

Bizonyítás. A bizonyítandó szerint egy \mathcal{RP} algoritmus minden \mathcal{NP} problémát meg tud oldani, ha egy $\oplus P$ -beli orákulumhoz hozzáférhet. Persze tudjuk, hogy elegendő $3SAT$ -ot megoldani és $\oplus P$ ereje benne rejlik $\oplus SAT$ -ban, azaz abban a problémában, hogy adott $\varphi(x_1, \dots, x_n)$ $3SAT$ probléma kielégítő értékeléseinek száma páros-e vagy páratlan.

Az állításnál erősebb tételt igazolunk. $3SAT$ -ot úgy oldjuk meg, hogy egy olyan orákulumot használunk, ami egy adott φ problémára megmondja, hogy nem kielégíthető, illetve egyetlen kielégítő értékelése van. Amennyiben kielégítő értékeléseinek száma több mint egy, akkor outputja tetszőleges lehet. Ezen probléma neve $USAT$ és nyilván könnyebb mint $\oplus SAT$ ($\oplus SAT$ egy speciális esete).

A bizonyító \mathcal{RP} algoritmus $3SAT$ megoldására egyszerű: Véltelen változó halmazokat generálunk: R_1, R_2, \dots, R_{n+1} , és véletlen paritásokat generálunk $\pi_1, \pi_2, \dots, \pi_{n+1}$ (a véletlen generálások függetlenül uniform eloszlással történnek). Majd kérdéseket teszünk fel az $USAT$ orákulumhoz:

- φ ,
- „ φ teljesül, továbbá R_1 -ben azon változók számának paritása, ami 1 értéket kap π_1 -e”
- „ φ teljesül, továbbá R_1 -ben azon változók számának paritása, amelyek 1 értéket kapnak π_1 -e, továbbá R_2 -ben azon változók számának paritása, amelyek 1 értéket kapnak π_2 -e”

- „ φ teljesül, továbbá R_1 -ben azon változók számának paritása, amelyek 1 értéket kapnak π_1 -e, továbbá R_2 -ben azon változók számának paritása, amelyek 1 értéket kapnak π_2 -e, továbbá R_3 -ban azon változók számának paritása, amelyek 1 értéket kapnak π_3 -e”

És így tovább az utolsó kérdésig az orákulumhoz: φ teljesül, továbbá R_1, R_2, \dots, R_{n+1} mindegyikében azon változók számának paritása, amelyek 1 értéket kapnak rendre π_1, \dots, π_{n+1} . (Ezen kérdések mindegyike könnyen megfogalmazható 3CNF alakban.) Ha bármikor az orákulum elfogad, akkor tudjuk hogy a kérdés biztos kielégíthető, így speciálisan φ is. Ha mindig nemleges választ kapunk az orákulumtól, akkor VALÓSZÍNŰLEG-ROSSZ állapottal állunk le.

Nyilván csak akkor hibázhatunk, ha $I_\varphi = \{x : \varphi(x) = 1\}$ nem üres és nem egyelemű. Ekkor alkalmas k (természetesen n -nél, a változók számánál nem nagyobb) pozitív egészre $2^{k-1} < |I_\varphi| \leq 2^k$.

Az alábbi lemma alapján készen vagyunk (a szokásos hibajavítást eredményező iterált futtatásokkal).

5. Lemma. *Legyen $I \subset \{0, 1\}^n$, amelyre $2^{k-1} < |I| \leq 2^k$. Legyenek R_1, R_2, \dots, R_{k+1} véletlen változó/koordináta halmazok, $\pi_1, \pi_2, \dots, \pi_{k+1}$ véletlen paritások. Legyen I_ℓ azon I -beli vektorok halmaza, amelyekre teljesül, hogy R_1, R_2, \dots, R_ℓ mindegyikére az általuk leírt pozíciókba rendre π_1, \dots, π_ℓ paritású sok 1-es esik. Ekkor*

$$\mathbb{P}[I_{k+1} \text{ halmaz 1 elemű}] > \frac{1}{8}$$

A lemma bizonyítása: Legyen x, y két különböző elem $\{0, 1\}^n$ -ből. Azt mondjuk, hogy R_1, R_2, \dots, R_{k+1} sorozat megkülönbözteti őket, ha valamelyik i -re x -nek és y -nak az R_i -be eső bitjei közül az 1-esek számának paritása különböző. Ekkor

$$\mathbb{P}[x \text{ és } y \text{ nem megkülönböztetett}] = \frac{1}{2^{k+1}}.$$

Valóban, legyen D azon koordináták halmaza, ahol x és y különbözik. R_i akkor és csak akkor különbözteti meg x, y -t, ha D -t páratlan sok elembe metszi. Mivel D egy nem üres halmaz, ezért ez egy $1/2$ valószínűségű esemény.

Ezen események különböző i -kre függetlenek, ami alapján a fenti egyenlőség nyilvánvaló. Így, ha $x, y \in I$ két különböző elem, akkor

$$\mathbb{P}[x \text{ nem megkülönböztetett valamelyik más } I\text{-beli elemtől}] < \frac{|I|}{2^{k+1}} \leq \frac{1}{2},$$

$$\mathbb{P}[x \text{ megkülönböztetett mindegyik más } I\text{-beli elemtől}] > \frac{1}{2}.$$

Azt mondjuk, hogy x túléli a $\pi_1, \pi_2, \dots, \pi_{k+1}$ sorozatot, ha $x \in I$ és R_1, R_2, \dots, R_{k+1} mindegyikére az általuk leírt pozíciókba rendre π_1, \dots, π_{k+1} paritású sok 1-es esik. Ennek valószínűsége $1/2^{k+1}$.

Legyen x egy elem $\{0, 1\}^n$ -ből. „ $I_{k+1} = \{x\}$ ” pontosan az az esemény, hogy x I mindegyik más elemétől megkülönböztetett az R_1, R_2, \dots, R_{k+1} sorozat által, továbbá x túléli a $\pi_1, \pi_2, \dots, \pi_{k+1}$ sorozatot. Ennek valószínűsége nagyobb mint

$$\frac{1}{2} \cdot \frac{1}{2^{k+1}} = \frac{1}{2^{k+2}}.$$

Az „ $I_{k+1} = \{x\}$ ” események, ahogy x végigfut I elemein diszjunktak és együtt kiadják az „ $|I_{k+1}| = 1$ ” eseményt. A diszjunkság miatt a valószínűség nagyobb mint

$$|I| \cdot \frac{1}{2^{k+2}} > 2^{k-1} \cdot \frac{1}{2^{k+2}} = \frac{1}{8},$$

ahogy bizonyítani kellett. ■

2. APPENDIX: Valiant-tétel bizonyítása

Először nézzük meg melyik problémát is vizsgáljuk.

Észrevétel. A következő függvények ugyanazok (a megfelelő kódolás mellett).

- (i) G egyszerű páros gráf \mapsto teljes párosítások száma,
- (ii) $M_{n \times n}$ 0-1 mátrix $\mapsto \text{per } M$,
- (iii) G egy párhuzamos élek nélküli irányított gráf $\mapsto \#\{V(G) \text{ parkettázásainak száma irányított körökkel}\}$,

Bizonyítás. (i)-ben, ha a két színsztály mérete különbözik, akkor a hozzárendelt érték biztos 0. A nem érdektelen esetben az alsó pont-felső pont szomszédsági mátrix egy 0-1-mátrix, amely permanense a gráfbeli teljes párosítások száma.

Ha az (ii)-ben szereplő mátrixot egy n pontú irányított gráf (hurokélek megengedet-tek, párhuzamos élek nem) „odavezet él” relációját leíró mátrixnak gondoljuk, akkor a permanenset kiadó 1-es szorzatok halmaza azonosítható azon körhalmazok hal-mazával, amik csúcshalmazai diszjunktan kiadják $V(G)$ -t.

A bizonyítás teljes lesz, ha észrevesszük, hogy mindkét fenti gondolatmenet meg-fordítható. ■

Mielőtt a bizonyításhoz fognánk az előző problémahármas súlyozott változatát is bevezetjük.

Észrevétel. A következő függvények ugyanazok (a megfelelő kódolás mellett).

- (i) $M_{n \times n}$ egész mátrix $\mapsto \text{per } M$,
- (iii) G egy párhuzamos élek nélküli \mathbb{Z} -súlyozott irányított gráf $\mapsto \sum \{\prod_{e \in E(\Phi)} w_e : \Phi \text{ csúcsparkettázás irányított körökkel}\}$

6. Tétel (Valiant-tétel). 0-1 PERMANENS egy $\#\mathcal{P}$ -teljes függvény.

Megjegyzés. Valiant tétele valójában egy kicsit gyengébb. Ő Turing-redukciót használva igazolta (1979) a teljességet. Az alábbi bizonyítás Ben-Dor és Halevi 1993-as bizonyításához áll legközelebb.

Bizonyítás. A bizonyítást a $\#3\text{SAT}$ -ra való visszavezetéssel végezzük. A $\#3\text{SAT}$ azt számolja ki, hogy egy φ 3CNF-nek hány kielégítő értéke van.

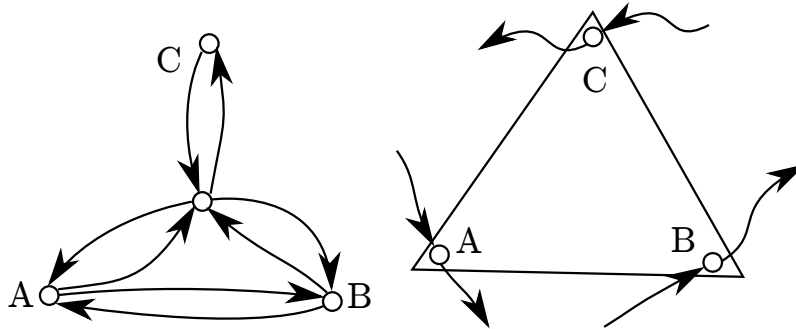
A 3CNF-hez egy alkalmas irányított gráfot konstruálunk, amely irányított körökkel történő csúcs-parkettázásainak száma nagyon szoros összefüggésben áll a formu-la kielégítő értékeléseinek számával. Így a csúcs-parkettázások számából könnyen számolható a kielégítő értékelések száma. Konstrukciónk több lépésben történik.

Először egy súlyozott gráfot konstruálunk meg, ahol a súlyok a $\{-1, 0, 1, 2, 3\}$ halmazból kerülnek ki. Ezekután a -1 súlyt kiküszöböljük (ára egy nagy pozitív súly lesz). Végül a problémát megoldó irányított gráfot (0-1 súlyozás) ismertetjük.

1. lépés: Súlyozott irányított gráfjainkban a 0 súlyt az él hiánya, az 1 súlyt a szokásos éljelölés, a -1 , 2 és 3 súlyt a súly feltüntetése és szín jelöli.

Először csak néhány összetevőt ismertetünk.

Minden klózhoz felvesszük a következő klóz- \triangle -gel izomorf részgráfot. (Olvasat: klóz-háromszög) Az ábránk bal oldalán a belső struktúra látszik, a jobb oldalán az, hogy a klóz- \triangle hogyan illeszkedik a gráfunk többi részéhez.

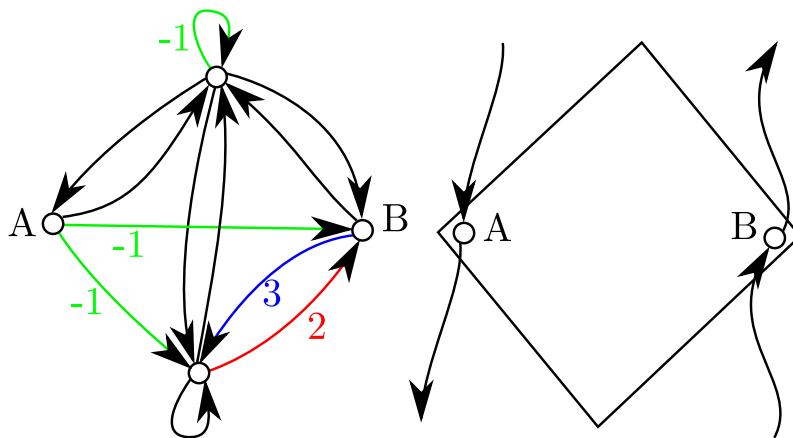


1. ábra.

A csúcsok, ahol az „alkatrész” és a „külvilág” találkozhat megbetűzöttek. A „névtelen” csúcsok az alkatrész „belsejében” vannak.

A háromszöghöz kapcsolódó külső éleket három párba csoportosítjuk: (A -ból kilépő, B -be belépő), (B -ből kilépő, C -be belépő), (C -ből kilépő, A -ba belépő). Tegyük fel, hogy mindhárom párba függetlenül egy-egy feltételt teszünk: vagy mindkét él a kör-parkettázásban szerepeljen vagy egyik sem. Az így kapott feltételrendszer (nyolc lehetőség) közül az egyik nem teljesíthető: ha a hat csatlakozó külső él mindegyikét a kör-parkettázás élének írjuk elő, akkor háromszög középső csúcsa nem fedhető. A maradék hét feltételrendszer azonban lokálisan — a háromszögön belül — egyértelműen kielégíthető.

Konstrukciónk következő összetevője egy literál- \diamond (olvasd: literál-négyzet):



2. ábra.

Ennek is ismertetjük egy körfedéshez való lehetséges viszonyát:

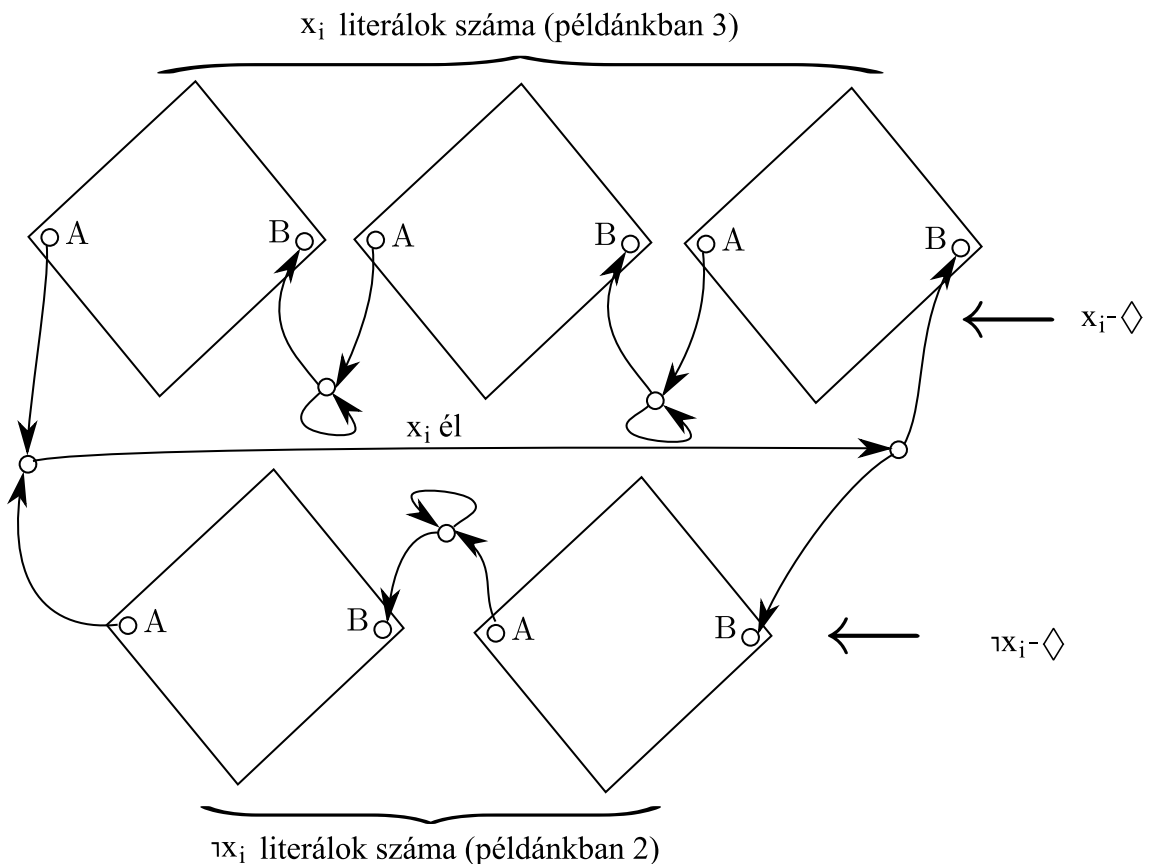
- (a) A -ra illeszkedő két külső él egyszerre szerepel vagy nem szerepel a csúcsparkettázásban, továbbá ugyanez igaz a B -re illeszkedő két külső élre is,
- (b) a csúcsparkettázó irányított körök egyike vagy A -n belép és B -n kilép a négyzetből vagy pedig fordítva: B -n belép és A -n kilép.

Most egy pontos előírás több belső csúcsparkettázás-töredéket is megenged egy literál- \diamond -en belül. Számoljuk ki a benti felhasznált élek szorzatát és a lehetőségekre összegezzük. Ekkor minden (a)-hoz tartozó előírás esetén 0-t kapunk, míg minden (b)-beli előírás esetén 4-et kapunk. (Az esetek végignézését az érdeklődő hallgatónak kell elvégeznie.)

Az összegzés lényege, hogy az előíráshoz tartozó parkettázások csoportosíthatók aszerint, hogy a négyzeten kívül hogy néznek ki. A súly-szorzatok összegzésében egy csoport közös külső része ugyanazt a tényezőt adja, ami kiemelhető. A zárójelben maradó részt számoltuk ki felül.

Legyen adva egy φ 3CNF formula. Most leírjuk, hogyan kapcsoljuk össze a fenti alkatrészeket.

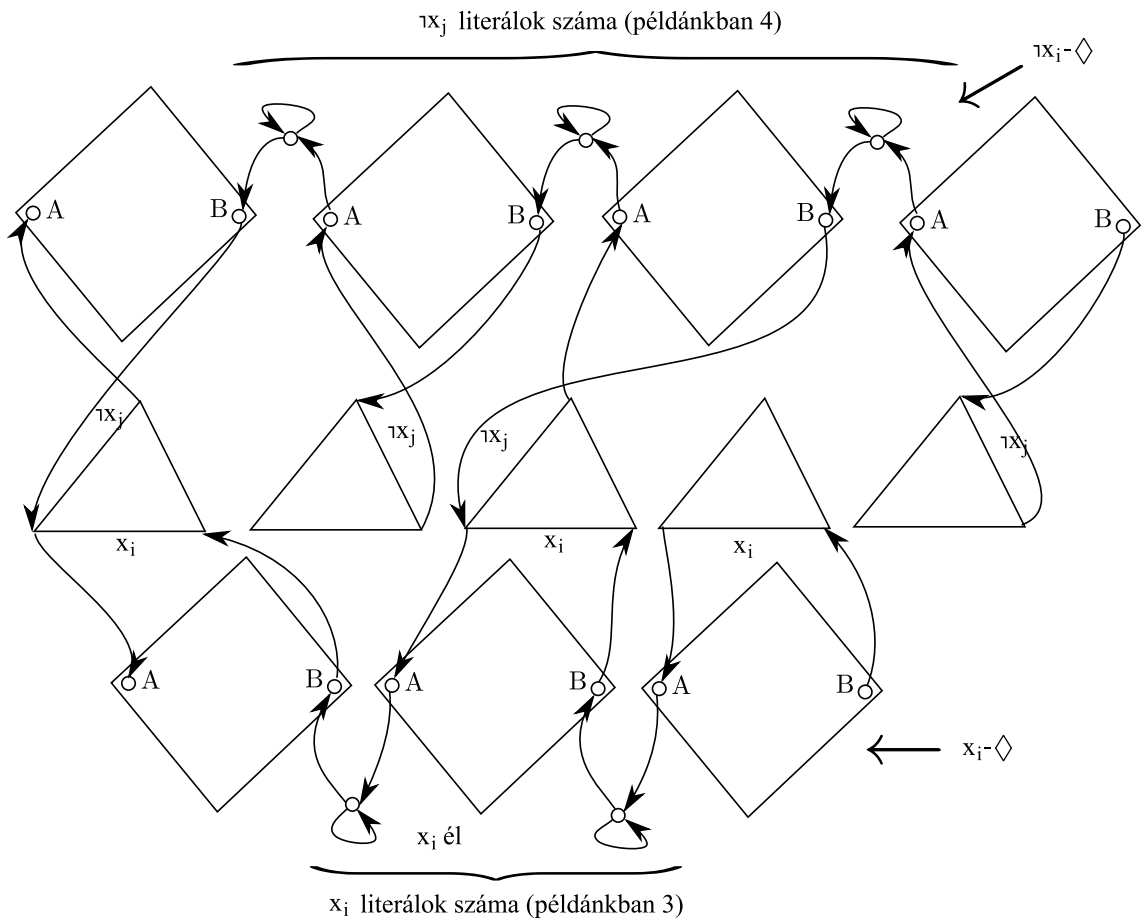
A formulában lévő literálok mindegyikére felveszünk egy-egy literál- \diamond -et. Minden x_i változóhoz felveszünk egy változó-élt és a változó, illetve negált változónak megfelelő φ -beli literáloknak megfelelő literál- \diamond -ekkel két irányított körben, az alábbi ábrán látható módon összekötjük őket:



3. ábra. Az x_i -változó élhez tartozó literál-négyzetek két körbe fűzésére egy példa (negátatlan és negált kör)

Ismét csak a betűzött csúcsoknál lesz kapcsolódási pont a „külvilággal”.

Végül minden klózra felvesszünk egy-egy klóz- Δ -t. A három oldal (AB , BC és CA a klózon belüli három literálnak felel meg). Az egyes literáloknak megfelelő oldalakat a literál- \diamond -okkal azonosítjuk és az alábbi módon összekötjük őket:



4. ábra.

Ezzel leírtuk a φ -hez rendelt G irányított súlyozott gráfot.

Irányított körök egy csúcs-parkettázásának minden csúcsot le kell fedni, a változó-élek kezdőpontjait is. Ezekből azonban egyetlen él indul ki: a változó-él. A megfelelő kör folytatása két irányba történhet: a változónak megfelelő négyzetek felé vagy negált változók felé. Így egy körökkel történő csúcs-parkettázásnak megfelel egy értékadás. Persze több csúcs-parkettázás ugyanazt az értékadást írhatja le. A csúcs-parkettázások között kijelölünk néhány speciálisat, amiket *standardnak* nevezünk. Ezekben az a kör amelyik egy x_i -változó-élt tartalmazza a választás után (x_i vagy $\neg x_i$ köre felé fordul-e) mindig belép a megfelelő négyzetbe, majd a másik oldalon kilép, így végig járja a teljes kört és záródik az x_i -változó-él köre. A nem választott kör négyzeteiben és a klózok háromszögeiben lévő csúcsok még fedetlenek. A négyzetek csúcsait fedjük úgy, hogy a négyzetbe belépünk, majd kilépünk. A háromszögek lefedését ezekután végezzük el. Ha egy klóz háromszög mindhárom oldalához be nem járt négyzet társul, akkor (és csak akkor) a háromszög fedése nem oldható meg (különben egyértelmű mit kell tenni). Ez viszont éppen az az eset, amikor a csúcs-parkettázás által leírt értékadás a klózt és így a formulát hamissá teszi. Tehát pontosan a kielégítő értékadásokhoz tartozik standard csúcs-parkettázás. Ezekhez is több. Azonban össz súlyszorzatuk mindegyiknek ugyanaz:

A négyzeteken belül függetlenül választható lehetőségek állítják össze a teljes listát. Legyen m a négyzetek száma, azaz φ -ben leírt literálok száma, azaz a klózik számának háromszorosa. Egy kielégítő értékadáshoz tartozó standard csúcs-parkettázások össz súly-szorzata 4^m . Egy nem kielégítő értékadáshoz nem tartozik standard csúcs-parkettázás.

A standard csúcs-parkettázások pontosan azok a csúcs-parkettázások, amelyek minden négyzethez úgy viszonyulnak, hogy egy körük belép, majd kilép. A nem standard csúcs-parkettázások tehát pontosan azok, amikre teljesül, hogy legalább egy négyzetben A és B fedése is a négyzethez viszonyítva belső vagy külső körök által megoldott. Azaz a négyzet belső körei vagy a négyzet összes csúcsát, vagy A kivételével, vagy B kivételével, vagy A, B kivételével a maradék csúcsokat parkettázzák. Ezek azonban össz szorzat-súlyban 0-t adnak.

Tehát a standard és nem-standard csúcs-parkettázások az össz szorzat-súlyban $4^m \cdot K$ -et adnak, ahol K a φ kielégítő értékadásainak száma.

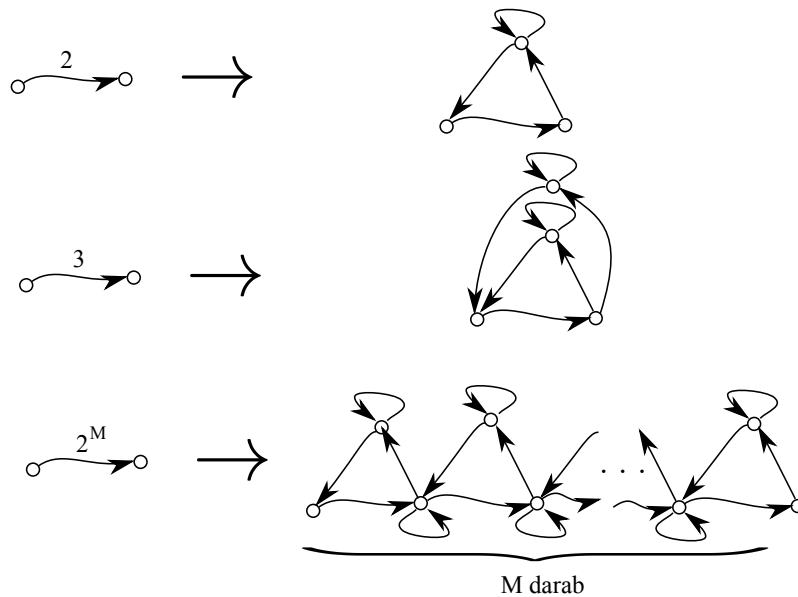
2. lépés: Ezt a lépést a mátrix nyelvezettel legegyszerűbb leírni. Tehát az előző lépésben definiált G súlyozott irányított gráfunk helyett nézzük a B „odavezet él” relációt leíró mátrixot (elemei $-1, 0, 1, 2, 3$ számokból kerülnek ki a súlyozást is kódolandó). Ennek a mátrixnak a permanense 4^m -szer a csúcs-parkettázások száma (speciálisan nem-negatív). A következő becslések teljesülnek B permanensére (N a csúcsok száma, M egy alkalmasan választott paraméter):

$$0 \leq \text{per} B \leq N!3^N < 2^M.$$

Rendeljünk a szomszédsági mátrixunkhoz egy olyan \tilde{B} mátrixot, ahol a -1 -ek helyére 2^M -et írunk. Ekkor $\text{per} \tilde{B} \equiv \text{per} B \pmod{2^M + 1}$.

A fenti két összefüggés alapján $\text{per} \tilde{B}$ alapján $\text{per} B$ könnyen számolható.

3. lépés: Legyen \tilde{G} a B mátrixnak megfelelő súlyozott irányított gráf (élsúlyai már nem-negatívok). A súlyokat az alábbi lokális helyettesítésekkel kiküszöböljük:



5. ábra.

Legyen \hat{G} az így kapott irányított (súlyozatlan) gráf. Ennek körparkettázásai

és \tilde{G} körparkettázásait élsúly-szorzat multiplicitással számolva két olyan sokaságot adnak, amelyek között természetes bijekció létesíthető.

\hat{G} a redukciónk végeredménye (az M paraméterrel együtt). A fentiek alapján ha ennek megállapítjuk a csúcs-parkettázásainak számát (egy 0-1PERMANENS kérdés), modulo $2^M + 1$ vesszük, elosztjuk 4^m -mel, akkor megkapjuk a φ -t kielégítő értékadások számát. Azaz redukciónk a tételt bizonyítja. ■