Bonyolultságelmélet jegyzet

Készítették Grolmusz Vince előadásai alapján a 2025/25. évi hallgatók

(Nem hivatalos lektorálatlan verzió)

2025. ősz

Contents

1 Kommunikációs játékok

Ennek a fejezetnek a nagy resze (majdnem minden) a szamitastudomany jegyzetbol lett atemelve.

Ezt a fejezetet ujra kell olvasni es megnezni mekkora az atfedes a szamitastudomanyon elhangzottak es a bonyelm-en elhangzottak kozott. A fo tetelek megtalahatok bizonyitasokkal: Teglalap fedes, Mehlhorn–Schmidt, AUY

Cél: van két játékos, akik bármit ki tudnak számolni gyorsan, de egymás között nehezen kommunikálnak.

Definíció 1.1 **Kommunikációs játék**

Adott $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ és $x, y \in \{0,1\}^n$. A ismeri x-et, de y-t nem, B ismeri y-t, de x-et nem. Ki akarják számolni $f(x,y)$-t. A költség az A és B között (bármely irányban) kommunikált bitek száma.

Akkor tekintjük $f(x,y)$-t kiszámoltnak, ha az egyik játékos ismeri $f(x,y)$-t, és a másik játékos tudja, hogy az egyik tudja.

Definíció 1.2 **Protokoll költsége**

A P protokoll mellett f költsége a legrosszabb (x,y) input páron $\kappa_{P(f)}$.

Megjegyzés

Megkövetelhetnénk, hogy mindketten tudják $f(x,y)$-t, ez 1 bit különbséget jelentene csak legfeljebb.

Definíció 1.3 **Protokoll**

A közös számolási módszer szabályait, hogy mikor ki, és milyen bitet küld protokollnak nevezzük. (Ez az algoritmus megfelelője több játékos esetén.)

Példa

Legyen f tetszőleges, ekkor A elküldheti x-et B-nek, aki „ingyen'' kiszámolja $f(x,y)$-t. Ennek a költsége n.

Példa **ID-függvény**

Legyen

$$\mathrm{ID}(x,y) = \begin{cases} 1, \text{ha } x = y \\ 0, \text{ha } x \neq y \end{cases}$$

Ekkor a fenti P protokollal $\kappa_P(\mathrm{ID}) = n$ teljesül.

Definíció 1.4 **Kommunikációs bonyolultság**

$\kappa(f)$ a $\kappa_P(f)$-ek minimuma az összes f-et kiszámoló P protokollon.

Tétel 1.5

$$\kappa(\mathrm{ID}) = n.$$

Ennek a bizonyításához kell a következő definíció és tétel.

Definíció 1.6 — **Kommunikációs mátrix**

Az f kommunikációs mátrixa az az $M_f \in \{0,1\}^{2^n \times 2^n}$, amelynek sorai x-szel, oszlopai y-nal vannak indexelve, és az x-hez tartozó sor y-hoz tartozó oszlopában $f(x,y)$ szerepel.

Megjegyzés

A továbbiakban a log mindig a 2-es alapú logaritmust jelenti.

Tétel 1.7 — **Mehlhorn–Schmidt**

$\kappa(f) \geq \log r(M_f)$, ahol $r(M_f)$ az M_f mátrix rangját jelöli.

Proof: Legyen P egy adott protokoll. Tegyük fel, hogy A kezd. Ekkor A kommunikál egy bitet. Ez rögzített P protokoll mellett bizonyos x-ekre 0, bizonyos x-ekre 1. Ezzel az M_f mátrixot két részre bontja: az egyik részben azon sorok vannak, amelyekre 0-t mond, a másikban azok, amelyekre 1-et. Ezek közül az egyik sorrangja $\geq \frac{1}{2} r(M_f)$.

Ezt ismételjük addig, amíg A lép. Amikor B lép, akkor ugyanez elismételhető oszloprangra, de egy mátrix sor- és oszloprangja megegyezik. Ha x és y olyan, hogy minden lépésnél a nagyobb rangú részmátrixot adják meg, akkor k lépés után a részmátrix rangja $\geq 2^{-k} r(M_f)$.

Tegyük fel, hogy a k. lépésben vége van a játéknak. Ekkor szimmetriaokokból feltehető, hogy A tudja $f(x,y)$-t, és B tudja, hogy A tudja. Mivel A tudja $f(x,y)$-t, az így kapott részmátrix minden sora homogén, azaz vagy csupa 0-t, vagy csupa 1-et tartalmaz. Ha pedig egy sor nem homogén, akkor A nem tudhatja biztosan $f(x,y)$-t. Hasonlóan, az, hogy B tudja biztosan $f(x,y)$-t, az azzal ekvivalens, hogy a kapott részmátrix minden oszlopa homogén.

Mivel homogén részmátrix rangja 1, az előbbi egyenlőtlenség szerint $1 \geq 2^{-k} r(M_f)$, azaz $2^k \geq r(M_f)$ fog teljesülni minden olyan (x,y) párra, amelyeket P k lépésben számol ki. □

Következmény 1.8

Innen könnyen kijön, hogy $\kappa(\mathrm{ID}) = n$, ugyanis $M_{\mathrm{ID}} = I_{2^n}$, és $r(I_{2^n}) = 2^n$, tehát $n \leq \kappa(\mathrm{ID})$ a Mehlhorn–Schmidt-tétel miatt. Másrészt láttuk, hogy $\kappa(f) \leq n$ minden f-re, így $\kappa(\mathrm{ID}) = n$.

Megjegyzés

Felső becslés nem ismeretes $\kappa(f)$-re. Lovász és Suchs nevéhez fűződő sejtés szerint $\exists c > 0$: $\kappa(f) \leq \log^c(r(M_f))$. Tudjuk, hogy $c > 2$ kell hogy teljesüljön. Ismert továbbá, hogy $\kappa(f) \leq r(M_f)$.

Következmény 1.9

$\mathrm{DISJ}(x,y) = \chi_{\{x \cdot y = 0\}}$, a halmazdiszjunktsági feladat. Akkor erre is $\kappa(\mathrm{DISJ}) = n$.

Proof of Következmény: Elemszám szerint rendezve az n elemű halmaz részhalmazait a sorokban, és a komplementereiket az oszlopokban

$$M_{\text{DISJ}} = \begin{bmatrix} 1 & * & * & \dots \\ 0 & 1 & * & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

felsőháromszög alakú, vagyis $\kappa(\text{DISJ}) = n$ □

Definíció 1.10 **Nemdeterminisztikus kommunikációs bonyolultság**

Alíz ismeri x-et, Bob ismeri y-t, E.T. ismeri mindkettőt, és f-et is. Utóbbi meg akarja győzni a játékosokat, hogy tudja. Ezt egy bizonyítással teszi, amit függetlenül A-nak, és B-nek is el kell fogadnia. Egy fix E.T. által az (x, y) párra adott bizonyítás hossza, amikor azt akarja bizonyítani, hogy $f(x, y) = 1$ legyen $\kappa_1^{\text{E.T.}}(f(x, y))$. Legyen továbbá

$$\kappa_1^{\text{E.T.}}(f) := \max_{\{x,y:f(x,y)=1\}} \kappa_1^{\text{E.T.}}(f(x, y)),$$

végül

$$\kappa_1(f) = \min_{\text{E.T.}} \kappa_1^{\text{E.T.}}(f)$$

a legjobb E.T. által a legrosszabb esetben adott bizonyítás hossza. Hasonlóan definiáljuk a $\kappa_0(f)$-et is.

Megjegyzés

$\max \kappa_0(f), \kappa_1(f) \leq \kappa(f)$ teljesül, hiszen reprodukálhatja az adott esetben a protokoll által megszabott kommunikációját

Példa

Ha $x \neq y$, akkor az $(i, x_i = 0)$ pár (ahol $y_i = 1$) megadása $\log(n) + 1$ bit hosszú, és bizonyítja, hogy az ID feladat nem teljesül. Egyenlőségre nem látszik kapásból hasonló jó bizonyítás.

Tétel 1.11 **Az ND kommunikációs bonyolultság jellemzése fedő téglalapokkal**

$\kappa_1(f)$ az a legkisebb t szám, hogy M_f egyesei lefedhetők 2^t darab csupa 1-es részmátrixxal

Megjegyzés

M_f-et már ismerjük, a kommunikációs mátrix. A tételben részmátrix alatt az oszlopok, és sorok egy-egy részhalmazait kiválasztva, a metszetekből álló részt értjük. Figyelem, ez nem feltétlenül egy összefüggő téglalap!

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

-ben az első és utolsó sor, és oszlopok által meghatározott rész is egy ilyen csupa egyes részmátrix.

Következmény 1.12

Láttuk, hogy $M_{\mathrm{ID}} = I_{2^n}$, ezt pedig csak úgy fedhetjük le csupa 1-es téglalapokkal, ha különkülön kiválasztjuk az átlóelemeket. Következik, hogy $\kappa_1(\mathrm{ID}) = n$.

Proof of Az ND kommunikációs bonyolultság jellemzése fedő téglalapokkal:

($\kappa_1(f) \leq t$)

Tekintsük a fedő téglalapokat. Alíznak van egy sora, Bobnak egy oszlopa. A protokollban megállapodnak a 2^t darab fedőmátrix egy sorrendjében. E.T. bizonyítása az lesz, hogy hanyadik részmátrixban van az (x, y) metszet, ez t bittel kódolható, leellenőrzik, hogy benne van-e az adatuk, és mivel ez csupa egyesből áll, így szükségszerűen $f(x, y) = 1$. %feltesszük, hogy E.T. nem hazudik?

($\kappa_1(f) \geq t$)

Legyen

$$H_\alpha = \{(x, y) : \text{A-nál } x, \text{ B-nél } y \text{ van, és } \alpha \text{ üzenetet hallják, akkor elfogadják a bizonyítást}\}.$$

Ha $(x_1, y_1), (x_2, y_2) \in H_\alpha$, akkor $(x_1, y_2), (x_2, y_1) \in H_\alpha$, hiszen az α bizonyítást Alíz elfogadta (x_1, y_1)-re, az ő nézőpontjából semmi nem különbözteti meg a szituációt attól, mintha (x_1, y_2) lenne a felállás, ezt pedig Bob is elfogadja, hiszen számára (x_1, y_2), és (x_2, y_2) ugyanolyan, és ez utóbbit elfogadta α-ra. Következik, hogy minden α-ra H_α megfelel egy részmátrixnak. Ha E.T. legfeljebb t bitből bizonyítani tudja, hogy $f(x, y) = 1$, ez szolgáltat lazannyát és 2^t darab csupa egyes részmátrixot. □

Randomizálva azonban gyorsan is lehet a következő Simon és Rabin nevéhez fűződő protokollal. A generál egy véletlen p prímet $\in \{1, .., n^2\}$ (ahol $\log x, \log y \leq n$), és elküldi az $(x \bmod p, p)$ üzenetet, B pedig leellenőrzi, hogy $x \equiv y \bmod p$ teljesül-e, és ezt mondjuk százszor megismétlik.

Ha egyszer is az teljesül, hogy inkongruensek, akkor az eredeti számok sem lehettek egyenlőek, ha mindig kongruensek, és mégsem egyenlőek, akkor százszor teljesült az, hogy $p|x - y \neq 0$.

A $\leq 2^n$ számoknak legfeljebb n darab prímosztója lehet, és n^2-ig nagyjából $\pi(n^2) \sim \frac{n^2}{2\log(n)}$ darab prím van. Annak a valószínűsége, hogy egyszer teljesül a kongruencia

$$\mathbb{P}(p|x - y) \leq \frac{n}{\frac{n^2}{2\log(n)}} = \frac{2\log n}{n} \to 0.$$

Egy kommunikáció $4 \log n$ bitet küld, ergo összesen $400 \log n$ bitnyi kommunikáció történik.

Nem (teljesen) triviális protokollok:

Példa

Tekintsünk egy fagráfot, aminek van két részfája. Kérdés, hogy az n csúcsú T fa T_1, T_2 részfáinak van-e közös csúcsa. Alíz kapja T_1-et, Bob T_2-t értelemszerűen, és mindketten ismerik T-t. Ez eldönthető lenne a DISJ játék speciális eseteként, de adunk egy okosabb protokollt.

Alíz megmondja T_1 egy tetszőleges v csúcsát (ez ugye $\log n$ bit kommunikáció). Majd Bob kiszámolja T_2-ben a v-hez legközelebbi w csúcsot, mivel fában egyértelmű út van két csúcs között, ez értelmes. Ezt visszaküldi Alíznak, ellenőrzi, hogy $w \in T_1$, ha igen, ez metszetbeli, és készen vagyunk, ha nem, akkor azt mondja, hogy a két fa diszjunkt. Ugyanis, ha a legközelebbi w pont nem része a fának, de egy további u pont része lenne T_1-nek, az uw szakasz T_2-ben van, az uv szakasz pedig T_1-ben, vagyis u közelebb van v-hez, mint w.

Példa

Most Alíz és Bob két részgráfot kap egy G gráfból úgy, hogy G_A független csúcsokból áll, G_B pedig egy teljes részgráf. Kérdés, hogy van-e metszet?

Világos, hogy ha van, legfeljebb 1 pontból állhat.

1. Alíz megnézi, hogy van-e legalább $\frac{n}{2}$ fokú v csúcs a gráfjában, ha igen, akkor $(1, v)$-t küldi el, ha nem, 0-t.
2. Bob megnézi, hogy van-e $< \frac{n}{2}$ fokú w csúcs G_B-ben, ha igen, $(1, w)$-t küld, ha nincs, 0-t.

Ezek után Bob tudja, hogy G_A v-ből, és a nem-szomszédaiból áll, ez legfeljebb $\frac{n}{2}$ csúcsból áll, és iteratíven folytathatjuk ezt az eljárást amíg lehet. Ha Bob talál egy kis fokszámú w csúcsot, akkor az ő gráfjának a többi csúcsa ennek a szomszédai közül kerül ki, és ismét rekurzíven folytatható az eljárás. Mi történik, ha mindketten 0-t küldenek? Alíz gráfjában minden csúcs kisebb mint $\frac{n}{2}$ fokú, G_B-ben pedig minden csúcs legalább $\frac{n}{2}$ fokú, ez a két feltétel kizárja egymást, így a két gráf diszjunkt. Addig ismételgetik a fenti lépést, amíg nem mondanak mindketten nullát. Egy lépés $\log n + 1$ bit, és $\log n$ lépésben persze kimerítik a gráfot, vagyis $O(\log^2 n)$ bitre van összesen szükség.

Tétel 1.13 **Aho–Ullman–Yanakakis**

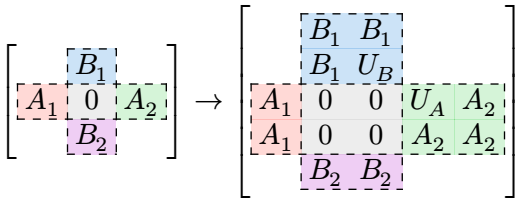
Minden f-re

$$\kappa(f) \leq (2 + \kappa_0(f))(2 + \kappa_1(f)).$$

Lemma 1.14

Ha M egy $0-1$ mátrix, H egy azonosan nulla részmátrixa, H sorai alkossák az A, oszlopai a B mátrixot, ekkor $\rho(A) + \rho(B) \leq \rho(M)$, ahol $\rho(M)$ a sor/oszloppermutációval képezhető legnagyobb négyzetes felsőháromszög részmátrix méretét jelölli, aminek a főátlója csupa 1-ből áll.

Proof of Lemma: A lemma azon múlik, hogy A és B-t külön-külön mozgathatjuk, a csupa nulla metszet nem fog változni, és a másik mátrixhoz nem nyúltunk hozzá, diszjunkt sorokból/oszlopokból áll. Egy permutációval megfelelő helyre visszük A-ban a maximális U_A felsőháromszög mátrixot, ezt B-ben is elvégezve (U_B) kapunk egy $\begin{bmatrix} U_B & \\ 0 & U_A \end{bmatrix}$ felsőháromszöget M-ben.



□

Proof of Aho–Ullman–Yanakakis: Világos, hogy $\rho(M_f) \leq r(M_f)$, és $\log \rho(M_f) \leq \kappa_1(f)$ teljesülnek, mert egy csupa 1 főátlójú felsőháromszög mátrix teljes rangú, illetve ref{NDKB jell} miatt.

Indukcióval belátjuk, hogy $\kappa(f) \leq (2 + \log \rho(M_f))(2 + \kappa_0(f))$. Ha $\rho(M_f) = 1$, akkor nem is kell kommunikálni, mert egy ilen mátrixban vagy csak egyesek állnak, vagy pontosan egy sorában vagy oszlopában vannak egyesek. Az általános lépésben tekintsük a kommunikációs mátrix nullásainak a fedését $2^{\kappa_0(f)}$ darab csupa nulla részmátrixszal. Alíz megnézi, hogy fedi-e az ő x inputjának egy részét olyan csupa 0 részmátrix, hogy a hozzá tartozó sorokból alkotott A mátrixra $\rho(A) \leq \frac{\rho(M_f)}{2}$, ha igen, akkor elküldi az (1, a csupa nulla részmátrix sorszáma) üzenetet, ez legfeljebb $1 + \kappa_0(f)$ bit kommunikáció, ha nincs ilyen részmátrix, akkor 0-t küld. Bob hasonlóan megnézi, hogy van-e az y-jához olyan fedő csupa 0 mátrix, amely oszlopaihoz tartozó B mátrixra $\rho(B) \leq \frac{\rho(M_f)}{2}$, ha igen (1,a fedő mátrix sorszáma), ha nincs ilyen, akkor pedig 0-t küld.

Mi történik, ha mindketten 0-t küldenek?

Akkor $f(x, y) = 1$, hiszen ha 0 lenne, akkor a metszetüket lefedné egy csupa 0 részmátrix, de az eddigi kommunikáció szerint az ezen fedőmátrixhoz tartozó sorok, és oszlopok ρ értékei összesen többet adnak, mint $\rho(M_f)$, ellentmondásban a lemmánkkal. □

Definíció 1.15 **Kommunikációs bonyolultságok**

$f \in \mathrm{P}^{\mathrm{CC}}$, ha $\exists c > 0 : \kappa(f) \leq \log^c n$.
$f \in \mathrm{NP}^{\mathrm{CC}}$, ha $\exists c > 0 : \kappa_1(f) \leq \log^c n$.
$f \in \text{co-NP}^{\mathrm{CC}}$, ha $\exists c > 0 : \kappa_0(f) \leq \log^c n$.

A fenti tétel következményeként adódik, hogy $\mathrm{P}^{\mathrm{CC}} = \mathrm{NP}^{\mathrm{CC}} \cap \text{co-NP}^{\mathrm{CC}}$.

Láttuk továbbá, hogy $\mathrm{NP}^{\mathrm{CC}} \neq \text{co-NP}^{\mathrm{CC}}$, mert ID benne van a jobb oldalban, de a balban nincs.

$\mathrm{P}^{\mathrm{CC}} \neq \text{co-NP}^{\mathrm{CC}}$ szintén az ID miatt (így $\mathrm{P}^{\mathrm{CC}} \neq \mathrm{NP}^{\mathrm{CC}}$ is teljesül).

2 NP-n túl

2.1 Polinomialis hierarchia

Definíció 2.1 **Polinomiális reláció**

Azt mondjuk, hogy $P(x, y_1, y_2, ..., y_l)$ egy polinomiális reláció, ha $\exists c$ úgy, hogy $\forall i : |y_i| \leq |x|^c$ es $P(x, y_1, ..., y_i)$ kiszámolható $|x|$-ben polinomimális időben.

Definíció 2.2 Σ_i

Tetszőleges L nyelvre $L \in \Sigma_i \Leftrightarrow \exists P(x, y_1, ..., y_i)$ polinomiális reláció úgy, hogy $x \in L \Leftrightarrow \exists y_1 \forall y_2 \exists y_3 ... Q y_i$ úgy, hogy $P(x, y_1, y_2, ..., y_i)$ teljesül. Ahol Q a következőképpen van definiálva:

$$Q = \begin{cases} \forall \text{ ha } i \text{ paros} \\ \exists \text{ ha } i \text{ paratlan} \end{cases}$$

Definíció 2.3 Π_i

Tetszoleges L nyelvre $L \in \Pi_i \Leftrightarrow \exists P(x, y_1, ..., y_i)$ polinomialis relacio ugy, hogy $x \in L \Leftrightarrow \forall y_1 \exists y_2 \forall y_3 ... \tilde{Q} y_i$ ugy, hogy $P(x, y_1, y_2, ..., y_i)$ teljesul. Ahol $\tilde{Q}$ a kovetkezokeppen van definialva:

$$\tilde{Q} = \begin{cases} \forall \text{ ha } i \text{ paratlan} \\ \exists \text{ ha } i \text{ paros} \end{cases}$$

Példa

Pár nevezetes bonyolultsági osztály amit már ismerünk:
1. NP $= \Sigma_1$
2. co-NP $= \Pi_1$
3. P $= \Sigma_0 = \Pi_0$

Megjegyzés

1. Minden i-re $\Sigma_i \subseteq \Sigma_{i+1}$. Valasszuk ugy a polinomimalis relaciot hogy az utolso valtozotol ne fuggjon.
2. Minden i-re $\Pi_i \subseteq \Pi_{i+1}$. Valasszuk ugy a polinomimalis relaciot hogy az elso valtozotol ne fuggjon.
3. Minden i-re $\Pi_i \subseteq \Sigma_{i+1}$.
4. Minden i-re $\Sigma_i \subseteq \Pi_{i+1}$.

Ezen osztalyokat a kovetkezo hierarchiaval tudjuk vizualisan jellemezni.

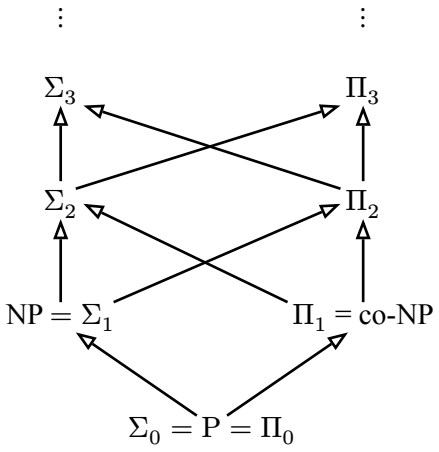


Figure 1: Polinomiális hierarchia vizualizáció

Definíció 2.4 **Polinomialis Hierarchia**

$$\mathrm{PH} = \bigcup_{i=1}^{\infty} \Sigma_i = \bigcup_{i=1}^{\infty} \Pi_i$$

Definíció 2.5

$$\mathrm{INDEPENDENT} := \{(G, m) : \alpha(G) \geq m\}$$

azaz, azon G grafok es m szamok parosai, melyekre G fuggetlensegi szama nagyobb mint m.

Definíció 2.6

$$\mathrm{EXACT\_INDEPENDENT} := \{(G, m) : \alpha(G) = m\}$$

azaz, azon G grafok es m szamok parosai, melyekre G fuggetlensegi szama pont m.

Állítás 2.7

$$\mathrm{EXACT\_INDEPENDENT} \in \Sigma_2$$

Proof: $\exists H \subseteq V(G)$ fuggetlen csucshalmmaz es $|H| = m$
$\forall H' \subseteq V(G)$ csucshalmazra, ahol $|H| = m + 1$ mar H' osszefuggo. □

Megjegyzés

A letezest ($\exists$) es a mindent ($\forall$) nem kell polinomialis idoben szamolni, csak a H-t es H'-t kell polinomialis idoben ellenorizni.

Tétel 2.8

Ha $\exists i \geq 1$ amire $\Sigma_i = \Pi_i$, akkor $\Sigma_{i+1} = \Pi_{i+1}$, amibol tovabb kovetkezik, hogy $\mathrm{PH} = \Sigma_i = \Pi_i$. Azt mondjuk, hogy a polinommialis hierarchia *osszeomlik* az i-edik szintre.

Proof: Mivel tudjuk, hogy $\Sigma_i \subseteq \Sigma_{i+1}$, ezert eleg azt belatnunk, hogy $\Sigma_{i+1} \subseteq \Sigma_i$ es ezzel belatjuk, hogy $\Sigma_i = \Sigma_{i+1}$. Hasonlo modon be tudjuk latni hogy $\Pi_i = \Pi_{i+1}$.

Legyen $L \in \Sigma_{i+1}$ tetszoleges nyelv, bizonyitsuk be hogy $L \in \Sigma_i$. Mivel $L \in \Sigma_{i+1}$, ezert letezik egy P polinomialis relacio, melyre

$$x \in L \Leftrightarrow \exists y_1 \forall y_2 \exists ... Q y_{i+1} P(x, y_1, ..., y_i).$$

Tovabba, letezik egy $L' \in \Pi_i$ nyelv, melyre

$$x \in L \Leftrightarrow \exists y_1 : (x, y_1) \in L'.$$

Figyelem, itt csak annyi tortent hogy beillesztettuk egy extra y_1 valtozot a letezes ($\exists$) kvantorral a Π_i definicio ele, igy kaptunk egy definiciot Σ_{i+1}-re.

Mivel $\Sigma_i = \Pi_i$, ezert $L' \in \Sigma_i$, tehat letezik egy polinomialis relacio S ugy, hogy

$$x \in L \Leftrightarrow \exists y_1 \exists y_2 \forall ... Q y_{i+1} S(x, y_1, ..., y_i).$$

Csoportosithatjuk y_1-et es y_2-t.

$$x \in L \Leftrightarrow \exists (y_1, y_2) \forall ... Q y_{i+1} S(x, (y_1, y_2), ..., y_i).$$

A jobboldalon i darab kvantor van es pont abban a sorrendben mint ahogy kell lenniuk Σ_i definiciojahoz. Tehat belattuk, hogy ha $L \in \Sigma_{i+1}$ akkor $L \in \Sigma_i$. □

Tétel 2.9 **Savitch**

Ha $f(n) \geq n$, akkor

$$\mathrm{NSPACE}(f(n)) \subseteq \mathrm{DSPACE}(f^2(n))$$

Proof: Legyen $L \in \mathrm{NSPACE}(f(n))$ egy tetszőlegese nyelv, a célunk megmutatni, hogy L felismerhető egy determinisztikus Turing-géppel $f^2(n)$ tárban.

Figyeljük meg, hogy ha egy Turing-gép t tárat használ futása alatt, akkor legfeljebb $O(2^{c \cdot t})$ különböző konfigurációba kerülhet.

Tudjuk, hogy van egy nemdeterminisztikus Turing-gép mely felismeri az L nyelvet, tehát a konfigurációs gráfban van út a kezdőállapotból a reprezentáns elfogadó állapotok csúcsába. Mivel legfeljebb $2^{c \cdot f(n)}$ konfiguráció van, ezért egy elfogadó út hossza legfeljebb $2^{c \cdot f(n)}$.

Ha tudunk mutatni egy determinisztikus Turing-gépet ami el tudja dönteni egy gárfban, hogy adott s és t csúcsok között van-e út $O(\log^2 n)$ tárban, akkor a konfigurációs gráfra alkalmazva $O(f^2(n))$ méretű tárat használó eljárást adnánk L felismerésére.

Megmutatjuk, hogy $O(\log^2 n)$ tárban el tudjuk dönteni, hogy s és t között megy-e út egy adott G gráfban. Legyen $\text{st-conn}(k, s, t)$ az algoritmus ami eldönti, hogy legfeljebb k hosszú út van-e s és t között. Nyilvan ha van olyan $u \in V(G)$ csúcs mely s-ből elérhető legfeljebb $k/2$ hosszú úton és u-ból t elérhető legfeljebb $k/2$ hosszú úton, akkor s-ből t is elérhető legfeljebb k hosszó úton.

Tehát a következőképpen néz ki a rekurziónk:

$$\begin{cases} \text{st-conn}(0, s, t) = \left(s \stackrel{?}{=} t\right) \\ \text{st-conn}(1, s, t) = (st) \stackrel{?}{\in} E(G) \\ \text{st-conn}(k, s, t) = \exists ? \, u \in V(G) : \text{st-conn}(k/2, s, u) \wedge \text{st-conn}(k/2, u, t). \end{cases}$$

Látszik, hogy a rekurzió mélysége $O(\log n)$ és mindegyik rekurzív hívásban csak a függvény argumentumait kell tárolnunk amiket bitekben $O(\log n)$ tárban meg tudjuk oldani. Tehát az st-conn algoritmus $O(\log^2 n)$ tárban működik, és ezzel készen is vagyunk, mivel

$$\left(\log\left(2^{c\cdot f(n)}\right)\right)^2 = (c\cdot f(n)\cdot \log 2)^2 = c^2\cdot f^2(n) = O(f^2(n)).$$

□

Következmény 2.10

$$\mathrm{NPSPACE} = \mathrm{PSPACE}$$

Proof: Polinom négyzete polinom. □

2.2 PSPACE teljesség

Definíció 2.11 **PSPACE teljesség**

Azt mondjuk, hogy L PSPACE teljes, ha $L \in \mathrm{PSPACE}$ és $\forall L' \in \mathrm{PSPACE}$ nyelvre $L' \propto L$. Tehát L' visszavezethető L-re polinomiális *időben*.

Definíció 2.12 **tqbf – Totally Quantified Boolean Formula**

Azt mondjuk, hogy φ egy teljesen kvantifikált Boole-formula, ha olyan alakba írható, hogy

$$\varphi = Q_1x_1Q_2x_2...Q_lx_lf(x_1, x_2, ..., x_l),$$

ahol $Q_i \in \{\forall, \exists\}$ és x_i Boole változók és $f(x_1, ..., x_l)$ egy konjunktív normál formula (CNF).

Példa

$$\varphi = \forall x\exists y\exists z((x \vee z) \wedge y)$$

Ez a formula igaz.

Megjegyzés

Egy tqbf vagy igaz vagy hamis. Nem olyan mint egy CNF ahol az a kérdés hogy van-e helyes behelyettesítése, hanem magát a kvantálás megválaszolja, hogy a formula igaz vagy hamis.

Definíció 2.13 **TQBF – True Quantified Boolean Formula**

$$\mathrm{TQBF} := \{\varphi,\ \text{ahol } \varphi \text{ egy tqbf és } \varphi = \mathrm{true}\}.$$

Azaz TQBF az igaz teljesen kvantifikált Boole formulák nyelve.

Tétel 2.14

A TQBF nyelv PSPACE teljes.

Proof:

1. $\mathbf{TQBF \in PSPACE}$

Végezzünk teljes indukciót a kvantorok számára. Ha $(n-1)$ kvantoros tqbf ellenörzését el tudjuk végezni $\text{poly}(n)$ tárban, akkor n kvantoros tqbf ellenörzésénél csak 1-el több bitet kell tárolnom a kvantor típsára és meg x_n értékét.

2. $\forall L \in \mathbf{PSPACE} : L \propto \mathbf{TQBF}$

Röviden megemlítjük, hogy ebben az esetben nem tudjuk azt a trükköt eljátszani amivel bizonyítottuk, hogy $\text{SAT} \in \text{NPC}$, mert a Turing-gép összes szabályos lépését leíró formula hossza már bőven nem polinomiális lesz. Ez a trükk azért működött a SAT feladatnál, mert ott polinomiális időben kellett ellenőriznünk, itt viszont a tárnak kell polinomiálisnak lennie.

Az ötlet, hogy újra felhasználjuk az st-conn feladatot. Ha fel tudjuk írni az st-conn feladatot mint egy polinomiális méretű tqbf a kezdő csúcsra és az elfogadó csúcsok reprezentására a konfiguráció gráfra, akkor készen lennénk. Mivel bármilyen polinomiális tárban felismerhető nyelvet át tudunk írni polinomiális időben egy polinomiálisan hosszú tqbf-re.

Nézzük mit ad a köztes csúcs trükk amit már használtunk a Savitch tétel bizonyításában:

$$\text{st-conn}(k,s,t) \Leftrightarrow \exists u \in V : \text{st-conn}(k/2,s,u) \wedge \text{st-conn}(k/2,u,t).$$

A probléma ezzel a felírással, hogy bár feleztük a k paramétert, de a formula hossza nőt, tehát összességében nem értünk el érdembeli javulást.

A trükk az, hogy kihasználjuk az univerzális kvantort ($\forall$), hogy ne kelljen dupláznunk a formula méretét:

$$\text{st-conn}(k,s,t) \Leftrightarrow \exists u \in V : \forall (x,y) \in \{(s,u),(u,t)\} : \text{st-conn}(k/2,x,y).$$

Ha az L nyelvet t tárban felismerte egy Turing-gép, akkor legfeljebb $2^{c \cdot t}$ konfigurációja van. Tehát a konfigurációs gráfnak legfeljebb $2^{c \cdot t}$ csúcsa van. Mivel a jobboldali formula mérete polinomiális és k értékét mindig felezzük, ezért a végső formula mérete polinomiális lesz.

□

Definíció 2.15 — **Generalized Geography játék**

Legyen (G,u) egy rendezett pár, ahol G egy irányított gráf és $u \in V(G)$ a gráf egy adott csúcsa. A játékot Alíz és Bob játsza a következő szabályok alapján:

- Alíz kezd az u csúcsból.
- Alíz és Bob felváltva lépnek.
- A jelenlegi csúcsból csak belőle kifele menő élel keresztül szabad lépni a következő csúcsba.
- Már látogatott csúcsba tilos lépni.
- Ha a soron következő játékosnak már nincs szabályos lépése, akkor az ellenfél nyer.

Megjegyzés

Ez a játék az általánosítása az ország-város játéknak, ahol felváltva sorolunk városokat azzal a megkötéssel, hogy a következő város azzal a betűvel kezdőthet amivel az előző végződött és az veszít aki már nem tud várost mondani.

Definíció 2.16 — **Generalized Geography osztály**

$$\text{GG} = \{(G,u) : \text{Alíznak van nyerő stratégiája } u\text{-ból indulva}\}.$$

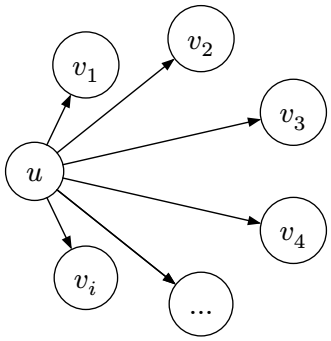
Tétel 2.17

A GG nyelv PSPACE teljes.

Proof:

1. GG $\in$ PSPACE

Végezzünk teljes indukciót a leghosszabb út hosszára. Ha polinomiális tárban el tudjuk dönteni, hogy Bob-nak nincsen stratégiája legfeljebb $(n-1)$ hosszú útra, akkor tudjuk, hogy Alíznak van nyerő stratégiája.



Nézzük meg u-nak az összes ki-szomszédjára, hogy Bob-nak nincs nyerő stratégiája. Mivel minden szomszédra polinomiális tárban eldönthetjük, és a tárat újra tudjuk használni, ezért az egész feladatot el tudjuk dönteni polinomiális tárban.

2. TQBF $\propto$ GG

A bizonyítás ezen részén azt kell belátnunk, hogy ha kapunk egy teljesen kvantifikált Boole-formulát, akkor arra tudunk adni egy irányított gráfot, amiben pontosan akkor van nyerő stratégiája Alíznak, ha a tqbf igaz. Legyen példál a tqbf a következő:

$$\varphi = Q_1 x_1 Q_2 x_2 ... Q_l x_l f(x_1, x_2, ..., x_l).$$

Az irányított gráfot két részből fogjuk felépíteni: kvantifikált értékadások (bal), és ellenőrzés (jobb). Az értékadás részben mindegyik x_i változóra létrehozunk egy kétirányú elágazást, ahol a balra vezető út azt jelenti, hogy x_i igaz, míg a jobbra vezető út azt, hogy x_i hamis.

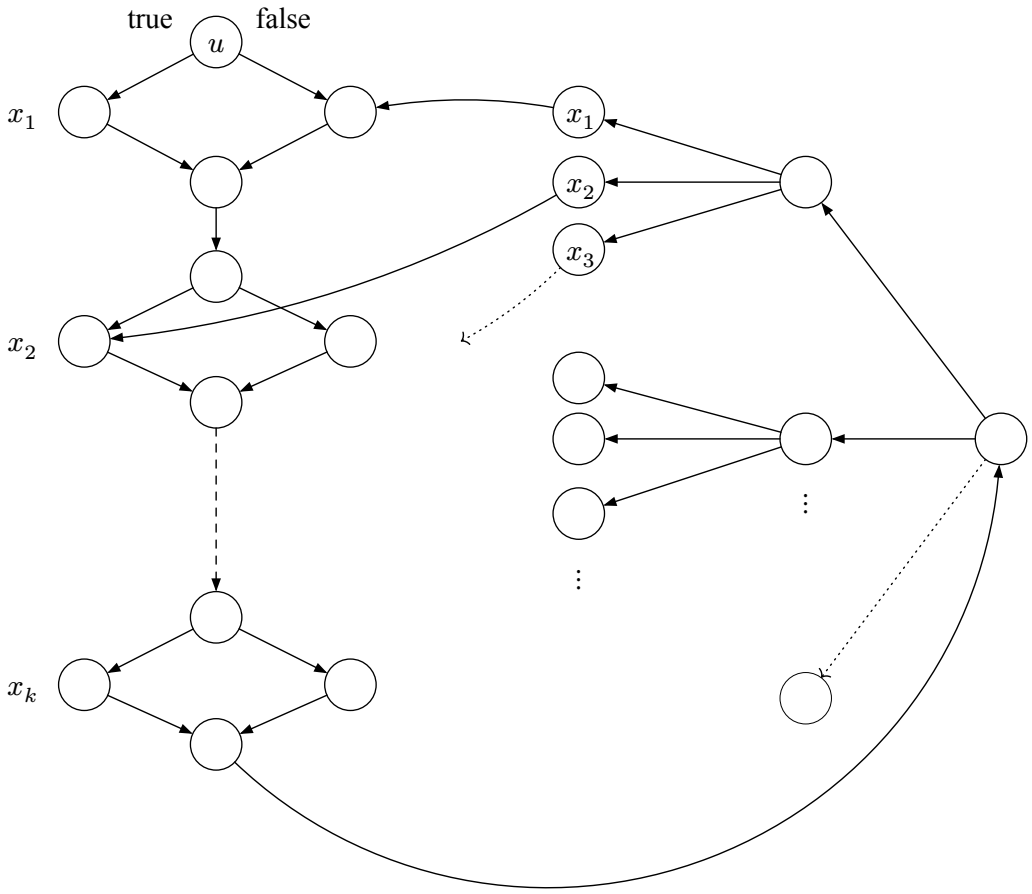
Az értékadásokat az alapján írjuk le, hogy éppen hogyan vannak kvantifikálva. Ha egzisztenciális kvantort ($\exists$) látunk, akkor Alízt kényszerítjük lépésre, ha univerzális kvantort ($\forall$), akkor Bobot kényszerítjük lépésre. Ha nem pontosan felváltva szerepelnek a kvantorok, akkor adunk az ellenfélnek egy triviális lépést ahol nincs választása csak előre menni.

Például, ha egymás után van $\forall x_i$ és $\forall x_{i+1}$, akkor egymás után kéne lépni kettőt Bobnak, de ezt a szabályok nem engedik. Ezért azt csináljuk hogy Bob lép egyet, azután beszúrunk egy választás nélküli direkt élt a következő választási lehetőséghez, ezzel kényszerítve Alízt és visszaadva Bobnak a lépés lehetőségét.

A második részben, ahol ellenőrizzük a formulát, úgy írjuk fel a gráfot hogy a gyűjtő csúcsba lépést Bobra kényszerítjük, így Alíz jön soron. Alíz rámutat egy blokk-ra ahol ő tudja hogy minden változó hamis. Így Bob bármelyik változót választja a blokkból az hamis lesz. Azt hogy egy változó hamis úgy mutatjuk meg, hogy miután Bob kiválasztotta, adunk Alíznak egy ingyen lépést és utána bekötjük a gráf első felébe oda ahol választottuk x_i értékét. Ha x_i tagadása szerepel a blokkban akkor abba a csúcsba kötjük be ami azt reprezentálja, hogy x_i hamis, különbe abba ami azt hogy x_i igaz.

Mostmár könnyű meggondolni, hogy ebben az irányított gráfban az, hogy Alíznak van nyerő stratégiája az ekvivalens azzal, hogy a tqbf igaz. Mivel az hogy Alíznak van nyerő stratégiája pont azt jelenti hogy amikor Alíz nyer akkor létezik (∃) olyan lépés, hogy Bob bármit lép (∀) még úgy is Alíz fog nyerni.

A következő ábra talán jobban elmagyarázza az érvelést.



□

3 Interaktív bizonyítások

Példa

Interaktív protokoll gráf *nem* izomorfizmusra.

Artúrnak vagy két gráfja G és G' melyekről el szeretné dönteni, hogy izomorfak-e. Artúr csak egy buta halandó ember, akkor is ha király, csak polinomiális idejű algoritmust tud lefuttatni a fejében. Szerencsére Merlin okosabb mint Artúr és a saját mágiájával bármit ki tud számolni a fejében egy lépés alatt, viszont nem feltétlenül mond mindig igazat.

Artúr a két gráf közül kiválaszt egy gráfot, permutálja a csúcsok számozását, és megkérdezi Merlintől, hogy melyik gráfot mutatja most éppen. Amire Merlin megmondja, hogy G vagy G' a mutatott gráf.

Ha a két gráf izomorf, akkor Merlinnek sincs esélye kitalálni melyik gráfot mutatja éppen Artúr, és így a legfeljebb tippelhet.

Tehát Artúr megkérdezi Merlint a fent leírt módon 100-szor, hogy a jelenleg mutatott gráf melyik. Ha Merlin mindegyik alkalommal jól válaszolt akkor vagy nem izomorf a két gráf és így Merlinnek egyértelmű melyik mutatja Artúr, vagy végig tippelt és így $\frac{1}{2^{100}}$ eséllyel mindig pont jót mondott.

Azaz, a gráf *nem* izomorfizmusra a fent leírt protokoll egy interaktív bizonyítás.

Definíció 3.1 **Interaktív protokoll**

Azt szeretnénk eldönteni, hogy egy adott w szóra és L nyelvre $w \overset{?}{\in} L$.

A bizonyítást Merlin és Artúr együtt fogják végezni. Merlin bizonyít, míg Artúr ellenőrzi Merlin bizonyításait. Artúr egy randomizált Turing-gép, míg Merlin bármit ki tud számolni az input alapján egy lépés alatt.

Először Merlin szól és mond egy polinomiális hosszú üzenetet. Erre Artúr Merlin üzenete és w függvényében polinomiálisan sok véletlen számot felhasználva válaszol. Ezt az interakciót megismételi a két fél amíg Artúr el nem szánja magát és vagy elfogadja a w szót vagy elutasítja.

Azt mondjuk, hogy a protokoll elfogadja az L nyelvet, ha $w \in L$ esetén van olyan Merlin, hogy Artúr legalább $1 - \frac{1}{2^{|w|}}$ valószínűséggel elfogadja w-t, és $w \notin L$ eseté minden Merlin esetén Artúr legfeljebb $\frac{1}{2^{|w|}}$ valószínűséggel fogadja el w-t, azaz téved.

Definíció 3.2 **IP osztály**

$$\mathrm{IP} := \{L : \quad w \in L\text{-et interaktív protokollal lehet bizonyítani}\}$$

Példa

$\mathrm{NP} \subseteq \mathrm{IP}$

Proof of példa: $w \in L \in \mathrm{NP}$, tehát létezik egy polinomiális tanu $w \in L$-re. Ha pont ezt a polinomiális tanut válaszolja Merlin akkor az egy jó egylépéses protokoll. □

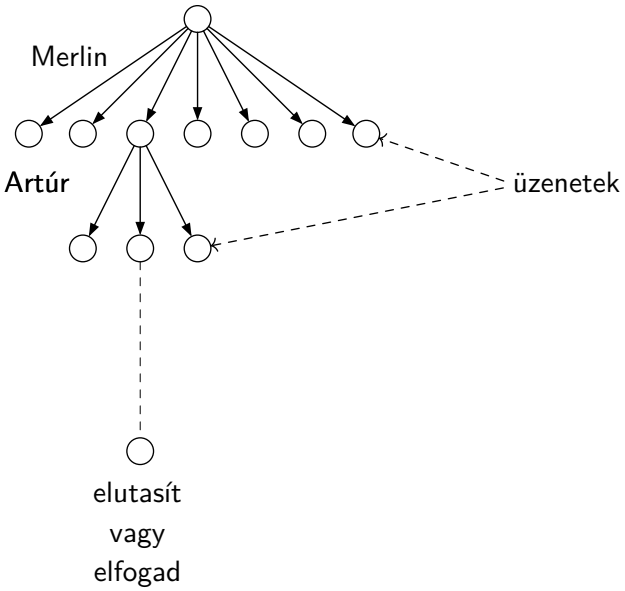
Tétel 3.3

$$\mathrm{IP} \subseteq \mathrm{PSPACE}$$

A bizonyítás előtt bevezjük a következő segítő fogalmat.

Definíció 3.4 **Protokoll fa**

A protokoll fa egy fix L nyelvre és fix w szóra az összes lehetséges Artúr–Merlin interakciót ábrázolja egy faként a következő módon.



Megjegyzés

Habár egy szinten exponenciálisan sok csúcs lehet, ezt a gráfot be tudjuk járni polinomiális tárban.

Proof of Tétel 3.3: Ha Artúr válaszol akkor vegyük a súlyozott átlagát a lehetséges válaszoknak az elutasítási valószínűségét.

Ha viszont Merlin válaszol akkor a maximális elutasítási valószínűséget adjuk meg.

TODO: Jobban leírni a bizonyítást. □

Tétel 3.5 **Shamir**

$$\mathrm{IP} = \mathrm{PSPACE}$$

A bizonyításhoz először bevezetünk pár segéd fogalmat és bizonyítunk valamit róluk. A tétel bizonyítása és segéd állítások az eredeti cikk alapján lettek feldolgozva, ami elérhető a következő linken: https://dl.acm.org/doi/pdf/10.1145/146585.146609

Definíció 3.6 **Egyszerű tqbf**

Azt mondjuk, hogy a φ teljesen kvantifikált Boole-formula *egyszerű*, ha minden x_i változójára igaz, hogy x_i kvantálásának helye és előfordulásának helye között legfeljebb egy darab univerzális kvantor ($\forall$) van.

Példa

A következő formula egy egyszerű tqbf:

$$\varphi = \forall x_1 \forall x_2 \exists x_3 [(x_1 \vee x_2) \wedge \forall x_3 (x_2 \wedge x_3 \wedge x_3)].$$

A következő formula nem egy egyszerű tqbf, mivel a kékkel jelölt x_1 változó kvantálása és második használata (negáltja) között kettő pirossal jelölt univerzális kvantor ($\forall$) is van.

$$\varphi = \forall x_1 \forall x_2 [(x_1 \wedge x_2) \wedge \forall x_3 (\overline{x}_1 \wedge x_3)]$$

Lemma 3.7

Minden teljesen kvantifikált Boole-formula egyszerű alakra hozható polinomiális időben.

Proof: Ha az eredeti tqbf nem egyszerű akkor van egy első x_i változó melynek a kvantálása és használata között több mint egy univerzális kvantor van. Ebben az esetben nézzük meg az első használatát x_i, mely már sérti a feltételt és cseréljük le x_i-t $\exists x_i^1 \left[(x_i \wedge x_i^1) \vee (\overline{x}_i \wedge \overline{x}_i^1)\right]$.

Tehát x_i kvantálása utáni első univerzális kvantor után létrehozunk egy új változót, x_i^1, mely értéke pont x_i.

Ezt a módosítást addig csináljuk amíg nem jutunk egyszerű formulához. □

Definíció 3.8 **Formula aritmetizáltja**

Egy teljesen kvantifikált Boole-formulához rendelt aritmetizáltja egy aritmetikai kifejezés, melyet úgy kapunk, hogy a formulában a következő előfordulásokat a megadott párjukra cseréljük le:

$$\begin{array}{lll} \text{True} \mapsto 1, & \text{False} \mapsto 0, & x \mapsto x \\ x \vee y \mapsto x + y & x \wedge y \mapsto x \cdot y & \overline{x} \mapsto (1 - x) \end{array}$$

A kvantorok viszont a következőképpen cseréljük le:

$$\forall x(\ldots) \mapsto \prod_{x \in \{0,1\}} (\ldots) \quad \exists x(\ldots) \mapsto \sum_{x \in \{0,1\}} (\ldots)$$

Megjegyzés

Az aritmetizált értéke egy egész szám.

Nyilván látszik, hogy a teljesen kvantifikált Boole-formula pontosan akkor igaz, ha az aritmetizáltja nem nulla.

Példa

$$\varphi = \forall x_1 \exists x_2 [(x_1 \wedge x_2) \vee \exists x_3 (\overline{x}_2 \wedge x_3)].$$

Ennek a tqbf-nek aritmetizáltja a következő:

$$f = \prod_{x_1 \in \{0,1\}} \sum_{x_2 \in \{0,1\}} \left[(x_1 \cdot x_2) + \sum_{x_3 \in \{0,1\}} (1 - x_2) \cdot x_3 \right].$$

Figyeljük meg, hogy φ aritmetizáltja nem feltétlenül egy polinomiális méretű szám, a következő példa mutatja hogy 2^{2^n} nagyságú is lehet:

$$\prod \prod \cdots \prod (...) \geq 2^{2^n},$$

ahol a belső $(...)$ kifejezés ≥ 2 és n darab produktum szerepel egymás mellett.

Állítás 3.9

Ha $0 < f < 2^{2^n}$, $f \in \mathbb{Z}$, akkor $\exists p$ prím úgy, hogy $2^n < p < 2^{2n}$és $f \not\equiv 0 (\mathrm{mod}\, p)$.

Proof: Tegyük fel, hogy $f \neq 0$. Ha $f \equiv 0 (\mathrm{mod}\, p_i)$ minden $2^n < p_i < 2^{2n}$, akkor a Kínai maradék tétel miatt $f \equiv 0 (\mathrm{mod}\, \Pi p_i)$.

A prímszám tétel azt állítja hogy ha $\pi(x)$ jelöli a az x-ig terjedő prímszámok számát, akkor

$$\pi(x) \sim \frac{x}{\log(x)}.$$

A mi esetünkben 2^n és 2^{2n} közé eső prímek száma a következő:

$$\frac{2^{2n}}{\log(2^{2n})} - \frac{2^n}{\log(2^n)} = \frac{2^{2n}}{2n} - \frac{2^n}{n} \geq 2^n.$$

Ebből az következik, hogy a 2^n és 2^{2n} közötti prímek szorzata legalább $(2^n)^{2^n} = 2^{n2^n} \geq 2^{2^n}$.
Viszont $f \leq 2^{2^n}$, ezért nem lehet 0 modulo egy nagyobb szám mint 2^{2^n}.
Ezzel ellentmondásra jutottunk és emiatt valóban létezik egy ilyen prím.

Ha viszont $f = 0$, akkor bármilyen prím jó, mert $f \equiv 0 (\mathrm{mod}\, p)$ bármilyen p-re. □

Definíció 3.10

Azt mondjuk, hogy f aritmetizáltnak a funkcionális formája $f(x_1)$, amit úgy kapunk, hogy f-ben szereplő első Π_{x_1} avagy Σ_{x_i} jelet eltörlünk és így már a kifejezés függ x_1 értékétől és egy függvényt kapunk.

Példa

Tekintsük a következő igaz tqbf-et:

$$\varphi = \forall x_i[\overline{x}_1 \vee \exists x_2 \forall x_3 (x_1 \wedge x_2) \vee x_3].$$

Ennek a tqbf-nek az aritmetizáltja a következő:

$$f = \prod_{x_1 \in \{0,1\}} \left[(1 - x_1) + \sum_{x_2 \in \{0,1\}} \prod_{x_3 \in \{0,1\}} (x_1 \cdot x_2 + x_3) \right],$$

melnyek értéke 2. Ezen aritmetizált funkcionális formája a következő:

$$f(x_1) = \left[(1 - x_1) + \sum_{x_2 \in \{0,1\}} \prod_{x_3 \in \{0,1\}} (x_1 \cdot x_2 + x_3) \right].$$

Egy mindenható Merlin persze egyből megmondja, hogy $f(x_1) = x_1^2 + 1$.

Egy tqbf aritmetizáltja exponenciálisan nagy fokszámú polinom is lehet, például

$$\varphi = \forall x_1 \forall x_2 ... \forall x_n (x_1 \vee x_2 \vee ... \vee x_n)$$

aritmetizáltja

$$f = \prod_{x_1 \in \{0,1\}} \prod_{x_2 \in \{0,1\}} ... \prod_{x_n \in \{0,1\}} (x_1 + x_2 + ... x_n),$$

melynek funkcionális formájában 2^{n-1}-ed fokon fog szerepelni $(x_1 + c)$ tag. Nyilván ilyen komplikált polinomokat Artúr nem tud kezelni.

Állítás 3.11

Ha φ egy egyszerű tqbf, akkor az aritmetizáltjának a funkcionális formája $f(x_1)$ egy polinom melynek foka legfeljebb lináris φ méretében.

Proof: Mivel φ egy egyszerű tqbf, ezért x_1 kvantálása és használata között legfeljebb egy univerzális kvantor lehet. A polinom fokát csak a szorzások befolyásolják és ebből legfeljebb egy lehet x_1 előtt ami duplázza x_1 fokát. □

Proof of $\mathrm{IP} = \mathrm{PSPACE}$***:*** Azt bizonyítjuk hogy interaktív protokollt tudunk adni a TQBF nyelv felismerésére. Az előbb láttuk hogy minden tqbf hozható *egyszerű* tqbf alakra, továbbá azt is hogy egy tqbf pontosan akkor igaz, ha aritmetizáltja nem nulla. Azt is láttuk, hogy ha $f \neq 0$, akkor van egy intervallumon egy prím amire $f \not\equiv 0 (\mathrm{mod}\, p)$.

Tehát a következőben csak arra adunk interaktív protokollt, hogy egy egyszerű tqbf aritmetizáltja $f \not\equiv 0 (\mathrm{mod}\, p)$.

Interaktív protokoll

A protokll azt fogja bizonyítani, hogy $f \not\equiv 0 (\mathrm{mod}\, p)$, ahol f már φ-nek az aritmetizáltja.

1. Először Merlin elküldi $f (\mathrm{mod}\, p)$ értékét Artúrnak és a funkcionális formáját egyszerű polinom alakra hozva $f(x_1)$.
2. Ha Merlin egy $\prod$ törlésével kapta a függvényt, akkor Artúr ellenőrzi, hogy $f(0) \cdot f(1) \equiv f (\mathrm{mod}\, p)$, ha viszont $\sum$ törlésével kapta akkor azt ellenőrzi, hogy $f(0) + f(1) \equiv f (\mathrm{mod}\, p)$.

Miután Artúr ellenőrizte f-et és $f(x)$-et, véletlenül választ $\xi \in \{0, 1, ..., p-1\}$ számot és behelyettesíti ξ-t és megkapja az $f(\xi)$ kifejezést. Merlinnek elküldi $f(\xi)$-t.

3. Erre Merlinnek ki kell számolnia $f(\xi)$-t és meg kell adnia egyszerű polinom alakra hozva a funkcionális formáját.
4. A protokoll így folytatódik tovább míg ki nem ürül a kifejezés.

Ha Merlin becsületesen játszik, akkor Artúr mindig elfogad.

Ha viszont Merlin csal, akkor csak ott van értelme csalnia hogy f értékéről hazudik. Ekkor viszont $f(x)$ polinomot is meg kell hamisítani vagy különben egyből lebukna Artúr egyszerű ellenörzésével. Tehát Merlin f helyett f'-t mond.

Feltéve hogy $f \not\equiv f' \pmod p$, akkor

$$\mathbb{P}((f - f')(\xi) = 0) \approx \frac{1}{2^n},$$

azaz annak a valószínűsége, hogy a két polinom értéke pont megegyezik ξ-ben exponenciálisan kicsi. Mivel $p > 2^n$ különböző szám lehet ξ és $f - f'$ fokszáma lineáris n-ben.

Következik, hogy annak a valószínűsége, hogy a sok csalás után megússza Merlin $\frac{1}{2^n}$. □

3.1 Zero Knowledge Proofs

Definíció 0.1 **Zero Knowledge Proof**

Egy zero knowledge proof (ZK) két szereplő interakcója: Merlin (a bizonyító) és Artúr (az ellenörző).

Egy ZK-nak a következő három tulajdonságot kell teljesítenie:

1. Ha a bizonyítandó állítás igaz, akkor egy őszinte Artúrt meg tudja győzni egy őszinte Merlin.
2. Ha az állítás hamis, akkor még csalással sem tudja Merlin meggyőzni Artúrt, legfeljebb egy kis valószínűséggel téved Artúr.
3. Ha az állítás igaz, akkor Artúr nem tud meg semmi extra információt azon kívül hogy igaz az állítás Merlin bizonyításából.

Megjegyzés

Az első két feltétel tehát csak annyit jelent, hogy ez egy interaktív bizonyítás. A lényeg pont az hogy Artúr nem tud meg semmi mást azon kívül hogy igaz az állítás.

Példa **Gráf nem izomorfizmus**

Adott két gráf G_1, G_2 és el szeretnénk dönteni, hogy $G_1 \not\simeq G_2$.

Artúr újra címezi a csúcsokat és megmutatja az egyik gráfot. Erre Merlinnek meg kell mondania, hogy G_1-et vagy G_2-t mutatta fel Artúr.

Példa **Hamilton kör létezése**

Adott egy G gráf, mely ismert Merlin és Artúr számára is. Merlinnek be kell bizonyítania, hogy ő ismer egy Hamilton-kört G-ben.

1. Merlin újra címezi G csúcsait egy permutációval, ezzel létrehoz egy H gráfot, mely persze izomorf G-vel.
2. Fölírja H éleit egy-egy lapra és fejjel lefelé elhelyezi őket az asztalra és mostmár nem nyúlhat hozzájük addig amíg Artúr meg nem engedi.
3. Most Artúr a következő kettő eset közül választ:
   - Merlin, bizonyítsd be, hogy H valóban izomorf G-vel.
   - Merlin mutasd meg a Hamilton-körödet H-ban.

1. Ha azt kell bizonyítania Merlinnek, hogy $G \simeq H$, akkor felfordítja az összes lapot és elmondja a permutációt amivel újra cimkézte a

csúcsokat. Ezzel Artúr meg tud győződni, hogy valóban izomorf a két gráf.

2. Ha viszont H-ban kell mutatnia Merlinnek egy Hamilton-kört, akkor felfedi csak azokat az éleket amik Hamilton-kört alkotnak.

Proof: Az előző bizonyítás valóban egy zero knowledge proof, mivel ha Merlin ismert egy Hamilton-kört G-ben, akkor erről meggyőződött Artúr. Ha Merlin nem ismert egy Hamilton-kört G-ben, akkor még Artúr kérdése előtt el kellett döntenie, hogy vagy egy valid izomorf gráfot kreál vagy egy valid Hamilton kört, ami nem izomorf G-vel. Mivel Merlin nem tudja előre Artúr kérdését ezért mindig $1/2$ eséllyel lebukik.

Másrészről, Artúr nem tudott meg semmi többet a Hamilton körről mint, hogy létezik, mivel akkor is amikor Merlin mutat neki egy Hamilton-kört akkor is csak H-ban mutat fel n élet ami n csúcsot tartalmaz, de ezt már eddig is tudta Artúr. □

4 Véletlen bonyolultság osztályok

Volt, hogy $L \in \mathrm{NP}$ pontosan akkor, ha létezik polinomiális tanu minden $w \in L$-re.

Definíció 4.1 RP

$L \in \mathrm{RP} \iff \exists T$ Turing gép amely $w \in L$ esetén legalább $1/2$ eséllyel elfogad és $w \notin L$ esetén 1 valószínűséggel elutasít.

Ezt mind várhatóan polinomiális időben.

Megjegyzés

A következő táblázat ilusztrálja az esetek valószínűségét:

Válasz / Valóság	Igaz	Hamis
Igaz	$\geq 1/2$	$\leq 1/2$
Hamis	0	1

Megjegyzés

Ha $L \in \mathrm{RP}$, akkor feltehető, hogy polinomiális ideig működik az őt elfogadó Turing-gép.

Proof of Megjegyzés: RP definíciójából a várható lépésszám polinomiális, tehát $\mathbb{E}(t(w)) \leq |w|^c$ valamilyen $c > 0$-ra.

A Markov egyenlőtlenség alapján:

$$\mathbb{P}(t(w) \geq 8|w|^c) < \frac{1}{8}.$$

Kezdjük el futtatni a T Turing-gépet mely a definícióban szerepel. Ha T leáll $8 \cdot |w|^c$ lépésen belül, akkor azt válaszoljuk, amit T. Ha viszont több mint $8 \cdot |w|^c$ lépésig fut, akkor elutasítunk.

Figyeljük meg, hogy a $w \notin L$, akkor nem tudunk tévedni, mivel vagy leáll az adott idő alatt és elutasítunk, vagy nem áll le és akkor is elutasítunk.

Ha viszont $w \in L$, akkor kétszer futtatjuk T-t az előző feltételekkel és csak akkor utasítjuk el w-t, ha mindkétszer elutasítanánk. Ekkor annak az esélye, hogy így is elutasítjuk, az

$$\left(\frac{5}{8}\right)^2 < \frac{1}{2}.$$

□

Megjegyzés

Az a sejtés, hogy $\mathrm{RP} = \mathrm{P}$. Ezt még nem sikerült bizonyítani, de nem sokan lepődnének meg ha valóban így lenne.

Megjegyzés

Érthetően $\mathrm{P} \subseteq \mathrm{RP} \subseteq \mathrm{NP}$.

Definíció 4.2 P/f

P/f azon nyelveket jelöli, melyekre létezik egy Turing-gép, ami elfogadja a w, a párt polinomiális időben, ahol a egy legfeljebb $f(|w|)$ méretű szó amit megsúgunk és csak az input hosszától függ.

Megjegyzés

Az előző definícióban fontos, hogy a súgás kizárólag az input hosszától függ, tehát minden k hosszú inputra ugyanazt a súgást adjuk meg a Turing-gépnek.

Megjegyzés

$\mathrm{P}/1 \stackrel{?}{=} \mathrm{P}$, ahol $\mathrm{P}/1$ azt jelenti, hogy mindegyik input hosszhoz kizárólag egy betűt/bitet súghatunk.

A válasz negatív, sőt $\mathrm{P}/1$-ben nem is mindegyik nyelv rekurzív, nem hogy polinomiális időben felismerhető nyelvek.

A bizonyításhoz tekintsünk egy tetszőleges nem rekurzív nyelvet a $\{0,1\}$ ábécé fölött, ilyen nyilván van mert majdnem mindegyik nyelv nem rekurzív. Mivel megszámlálhatóan sok Turing-gép van és mindegyik Turing-gép legfeljebb egy nyelvet ismer fel és megszámlálhatatlanul sok nyelv van adott ábécé fölött, ezért madjnem minden nyelv *nem* rekurzív.

A $\{0,1\}$ ábécé fölötti L nyelvet felfoghatjuk úgy mint a természetes számok egy részhalmazát $L \subseteq \mathbb{N}$, ahol egy szó egy természetes szám bináris alakja. Eltekintve attól hogy mit kezdünk a kezdő nullásokkal…

Legyen K az a nyelv, amiben a $\overbrace{11111.....11111}^{x}$ szavak vannak, ahol $x \in L$. Tehát K egy unáris nyelv ahol pontosan az x hosszú szavak vannak, ahol $x \in L$.

Ez a nyelv nyilván nem rekurzív, mert L sem az.

Definíció 4.3 P/poly

$$\mathrm{P}/\mathrm{poly} = \bigcup_{k=1}^{\infty} \mathrm{P}/n^k$$

Definíció 4.4 $\mathrm{P}/\log$

$$\mathrm{P}/\log = \bigcup_{k=1}^{\infty} \mathrm{P}/k \cdot \log n$$

Definíció 4.5 $L^{=n}$

$$L^{=n} = L \cap \Sigma_0^n$$

Tehát L-nek a pontosan n hosszú szavai.

Tétel 4.6 **Adleman**

$$\mathrm{RP} \subseteq \mathrm{P/\ poly}$$

Proof: Legyen $L \in \mathrm{RP}$, azt kell belátnunk, hogy $L \in \mathrm{P/\ poly}$. Legyen továbbá $n = |w|$ és tegyük fel, hogy $\Sigma_0 = \{0, 1\}$.

Ugye azt kell belátnunk, hogy van polinomiális hosszú súgás, amivel már el tudja dönteni egy Turing-gép egy szó tagságát, legyen ez a polinom p. Így feltehetjük, hogy L-et (RP értelemben) felismerő T Turing-gép egy $p(n)$ hosszú véletlen bitsorozatot használ.

Tehát egy $r \in \Sigma_0^{p(n)}$ véletlen bitsorozat meghatározza T működését.

Azt mondjuk, hogy S *jó* $L^{=n}$-re, ha $\forall w \in L^{=n}$-re $\exists r \in S$ úgy, hogy T lefogadja a (w, r) párost. Szavakban annyit jelent, hogy S bitsorozatok halmaza, melyben van olyan bitsorozat amellyel T elfogadja w-t.

Nyilván $S = \Sigma_0^{p(n)}$ jó, mivel ebben benne van az összes $p(n)$ hosszú bitsorozat. A továbbiakban ezt a halmazt szeretnénk szűkíteni.

Tudjuk, hogy adott $w \in L^{=n}$-re annak a valószínűsége, hogy T elutasítja w-t véletlen r-el az legfeljebb $1/2$.

Ha S pontosan k darab véletlen $p(n)$ hosszú bitsorozatból áll, akkor annak a valószínűsége, hogy T elutasítja w-t minden $r \in S$ bitsorozatra az legfeljebb $1/2^k$.

Így annak a valószínűsége, hogy $\exists w \in L^{=n}$, hogy T elutasítja w-t minden $r \in S$ bitsorozatra, az

$$< \frac{|L^{=n}|}{2^k} \leq \frac{2^n}{2^k}.$$

Ha $k = n + 1$, akkor az előző legfeljebb $1/2$. Mivel ez a valószínűség kisebb 1-nél ezért kell léteznie egy olyan bitsorozatnak a k közül ami jó az összes w-re.

Tehát adott n hosszú w inputra súgni fogunk egy k bitsorozatból álló S halmazt, amit képzelhetünk úgy mint k darab $p(n)$ bitsorozat konkatenálva, mert az osztály definíciójában csak egy bitsorozatot súghatunk. Látjuk, hogy a súgás mérete $k \cdot p(n)$, ami polinomiális n-ben.

Végül lefuttatjuk mind a k darab $p(n)$ hosszú bitsorozatra T-t. □

Definíció 4.7 **Ritka nyelv**

Azt mondjuk, hogy L ritka, ha $\exists c > 0$ úgy, hogy $\forall n\ |L^{=n}| \leq n^c$.

Tétel 4.8 **Mahaney**

Ha L ritka és NP-teljes, akkor $\mathrm{P} = \mathrm{NP}$.

Definíció 4.9 **Unáris nyelv**

Azt mondjuk, hogy az L nyelv unáris, ha $L \subseteq \{1\}^*$, azaz olyan szavakból áll melyekben csak 1-es szerepel. (pl.: 1111 vagy 11)

Tétel 4.10 **Berman**

Ha L unáris és NP-teljes, akkor $\mathrm{P} = \mathrm{NP}$.

Proof: TODO □

Lemma 4.11

DFS a CNF fában, akkor megállásig vegy n darab kielégíthető csúcson mentem át vagy nullán... TODO

Proof: kinda trivi. □

Tétel 4.12 **Fortune**

Ha A egy ritka nyelv és $\overline{\mathrm{SAT}} \propto A$, akkor $\mathrm{P} = \mathrm{NP}$.
Itt a $\overline{\mathrm{SAT}}$ azt a nyelvet jelöli amiben azon konjunktív normál formák vannak, melyeket nem lehet kielégíteni.

Proof: Mondjuk, hogy $|A| \leq n^c$, ekkor ugyanazt az érvelést követve, mint a Tétel 4.10 bizonyításában, ha nem találtunk kielégítő értékadást a fában DFS-el, akkor legfeljebb $n \cdot n^c$ lépést tettünk. Ha viszont találtunk kielégítő értékadást, akkor még gyorsabban megtaláltuk mint az előző esetben. □

Definíció 4.13

Azt mondjuk, hogy a B nyelv $P/c\log n$ időben visszevezethető az A nyelvre, ha van olyan Turing-gép, mely $c\log n$ méretű súgással $|w|$-ben polinomiális időben ki tudja számolni $f(w)$, amelyre $w \in B \iff f(w) \in A$.

Jelölésben $B \underset{P/c\log n}{\propto} A$.

Következmény 4.14

Ha A ritka és $\overline{\mathrm{SAT}} \underset{P/c\log n}{\propto} A$, akkor $\mathrm{P} = \mathrm{NP}$.

Proof: A T Turing-gép tippelje meg az $a_{|w|}$ súgást, ha a T gépg több mint $(n+1)|A^{=n}|$ lépést tesz meg akkor leállítjuk és tippelünk egy új súgást. □

Definíció 4.15

$\mathrm{NP}/f(n)$ azon L nyelvek osztályát jelöli, melyekre van egy olyan T nem determinisztikus Turing-gép, mely $w \in L$-et egy $f(|w|)$ méretű súgással elfogadja.

Lemma 4.16

Ha A ritka és $A \in \mathrm{NP}$, akkor $\overline{A} \in \mathrm{NP}/c\log n$.

Proof: Legyen a w inputra a $|w|$ hosszú szavak száma A-ban, azaz $l = \left|A^{=|w|}\right|$. A T nem determinisztikus Turing-gép (nem determinisztikusan) legenerálja az összes $|w|$ hosszú szót és mindegyikre leelenörzi, hogy eleme-e A-nak, így megkapjuk $A^{=|w|}$-t.

Ezután végigmegy $A^{=|w|}$ összes w_i elemén és leelenőrzi, hogy $w_i \stackrel{?}{=} w$. Ha van olyan i, melyre $w_i = w$, akkor $w \in A \Rightarrow w \notin \overline{A}$, különben $w \in \overline{A}$. Ezzel így akkor felismertük $\overline{A}$-t a súgással. □

Proof of Mahaney: Ha $A \in \mathrm{NPC}$ és A ritka, akkor $\mathrm{SAT} \propto A$ és $\overline{\mathrm{SAT}} \propto \overline{A}$, sőt ez a két visszavezetés ugyanazzal az f-el megy.

A Lemma 4.16 miatt, ha A ritka és $A \in \mathrm{NP}$, akkor $\overline{A} \in \mathrm{NP}/c \log n$. Tehát $\overline{\mathrm{SAT}} \propto \overline{A} \in \mathrm{NP}/c \log n$, ami miatt $\overline{\mathrm{SAT}} \in \mathrm{NP}/c \log n$.

Mivel $A \in \mathrm{NPC}$, ezért $\overline{\mathrm{SAT}} \underset{P/c \log n}{\propto} A$ és így a Következmény 4.14 miatt $\mathrm{P} = \mathrm{NP}$. □

Definíció 4.17 **BPP**

BPP azon L nyelvek osztályát jelöli, melyekre létezik egy randomizált T Turing-gépt, hogy $w \in L$ esetén $> \frac{2}{3}$ eséllyel elfogad és $w \notin L$ esetén $< \frac{1}{3}$ eséllyel fogad el. És a futásidő várhatóan polinomiális.

A nevét a *Bounded error Probabilistic Polynomial* kifejezésből kapja.

Megjegyzés

A következő táblázat ilusztrálja az esetek valószínűségét:

Válasz / **Valóság**	**Igaz**	**Hamis**
Igaz	$\geq 2/3$	$\leq 1/3$
Hamis	$\leq 1/3$	$\geq 2/3$

Megjegyzés

A BPP definíciójában is hasonlóan le lehet cserélni a *várhatóan* polinomiális időt konkrét polinomiális időre a Megjegyzés 4.2 érvelésével.

Megjegyzés

A sejtés az, hogy $\mathrm{P} = \mathrm{BPP}$.

Megjegyzés

Nyilván $\mathrm{P} \subseteq \mathrm{RP} \subseteq \mathrm{BPP}$.

Definíció 4.18 **ZPP**

$$\mathrm{ZPP} = \mathrm{RP} \cap \text{co-RP}$$

Megjegyzés

A fenti definíció ekvivalens azzal, hogy olyan L nyelvek vannak ZPP-ben, melyekre van olyan T probabilisztikus Turing-gép, mely hiba nélül felismeri L-et. És ezt mind várhatóan polinomiális időben teszi meg.

Az osztály a nevét a *Zero (error) Probabilistic Polynomial* kifejezésből kapja.

Megjegyzés

A tanult Schwartz-Zippel lemma érvén azt kapjuk, hogy a polinom különbség feladat co-RP-beli. Azaz, azon L nyelv, melyben (p, q) polinom párosok vannak, ahol $p \not\equiv q$.

Állítás 4.19

Ha a T probabilisztikus Turing-gép azonosan egyenletes eloszlás független bitekez használ, akkor ekvivalens definíciókat kapunk az előzőkkel.

Proof: Nem bizonyítjuk. □

Tétel 4.20

$$\mathrm{BPP} \in \Sigma_2 \cap \Pi_2$$

Proof: TODO □

Állítás 4.21

$\mathbb{F}_2^m$ S_X *kevés* eltolásával fedhető, ha $x \in L$.
Azaz $|S_X| \geq (1 - 2^{-n})2^m$.
Azaz $\exists r_1, ..., r_k \in \mathbb{F}_2^m$, ahol $k = \lceil \frac{m}{n} \rceil + 1$ úgy, hogy

$$\bigcup_{i=1}^{k} (S_X + r_i) = \mathbb{F}_2^m.$$

Proof: TODO □

5 Boole-hálózatok

Definíció 5.1 **Boole-hálózat**

A $\mathcal{C}$ egy *Boole-hálózat*, ha egy irányított körmentes gráf (DAG), aminek csúcsai címkézettek. Továbbá, a következő speciális csúcsokkal rendelkezik:

- input csúcsok: $u : d_{\text{in}}(u) = 0$
- output csúcsok: $u : d_{\text{out}}(u) = 0$
- belső csúcsok: minden más.

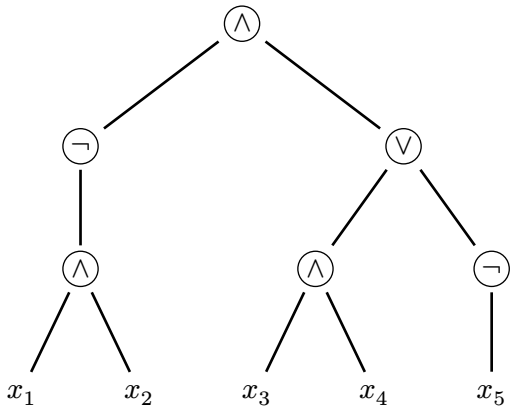
A belső csúcsok mindegyikén adott egy Boole függvény, melynek változói a beéleken keresztül kapott csúcsok értékei.

A $\mathcal{C}$ Boole-hálózat inputjaira írunk biteket és az kiszámolja a belső csúcsokon lévő Boole-függvények segítségével az értékeket, addig ameddig el nem érnek az értékek az output csúcsokba. Az input csúcsok értéke az input, az output csúcsok értéke az output.

Megjegyzés

A következő ábra illusztálja, hogy hogyan képzelhetünk el egy Boole-hálót. Figyelem, az ábrázolt Boole-hálózat egy fa, viszont lehetnek kereszt élek is, azaz irányítatlan körök, csak irányított körök tilosak.

A legalján vannak az inputok, a legtetején az output. A legalsó szinttől egyesével felfele számoljuk ki a szintek értékeit az előző szint segítségével.



Megjegyzés

Angolul *Boolean-circuit*-nek hívják a Boole-hálózatokat. Létezik olyan is, hogy Boolean-network de az teljesen más.

Definíció 5.2

A $\mathcal{C}$ Boole-háló mérete a csúcsainak száma.

Definíció 5.3

A $\mathcal{C}$ Boole-háló mélysége a leghosszabb s–t út hossza, ahol s egy input csúcs és t egy output csúcs.

Megjegyzés

Minden $f : \{0,1\}^n \to \{0,1\}$ függvényt ki lehet számolni 3 mély, exponenciális méretű, Boole-hálózattal, ami csak a $\wedge, \vee, \neg$ függvényeket alkalmazza a belső csúcsokon.

Proof: Ismert, hogy minden Boole-függvény felírható konjunktív normál forma alakba. Tehát, a Boole-hálózat legalsó szintjén az inputokat csoportosítjuk klózokba $\vee$ jelekkel, ahol kell beillesztünk egy $\neg$ jelet, ha $\neg x_i$ szerepel a klózban. Az alkozott klózokat meg behúzzuk az egyetlen output csúcsba, amiben egy $\wedge$ jel szerepel, tehát *össze vagyoljuk a klózokat*. □

Tétel 5.4

Adott $\{0,1\}$ fölötti L nyelvre $L \in P/\text{poly} \iff \exists \mathcal{C}_1, ..., \mathcal{C}_n$ Boole-hálók, hogy $\mathcal{C}_i$ az i inputon *működik* és $\exists c > 0 : \mathcal{C}_i$ mérete legfeljebb i^c és $w \in L$-et eldönti a $\mathcal{C}_{|w|}$ háló.

Proof:

$\Longleftarrow$: $\forall n : |w| = n$ a súgás legyen C_n. Mivel C_n mérete polinomiál ezért C_n leírása is polinomiális és tudunk alkotni egy Turing-gépet, amely polinomiális futásidő alatt tudja szimulálni C_n Boole-háló működését.

$\Longrightarrow$: Ha $L \in P$, akkor létezik olyan T Turing-gép, ami L-et eldönti polinomiális időben. Feltehető, hogy T egy egyszalagos Turing-gép, mert tudjuk, hogy minden Turing-gép szimulálható egy egyszalagossal, persze négyzetese lassítással, de ez még belefér a polinomiális futásidőbe.

Csináljunk úgy, mintha a T egyetlen szalagjára írt input olyan mint C_n

Tehát minden mezőre egy lépésről a másikre egy $O(1)$ bemenetű Boole-függvény dönte el az új állapotot.

Mivel T polinomiálisan hosszú ideig fut, ezért legfeljebb polinomiális mélységű az így kapott háló.

Ha $L \in \text{P/poly}$, akkor az első lépésben a szalag úgy néz ki, hogy van a szokásos input a szalagon balról jobbra, és ettől az inputtól balra odaírjuk még a polinomiális súgást. □

Következmény 5.5

Ha kiderülne, hogy G3C-re nincs őt felismeről polinomiális méretű Boole-háló, akkor $\text{P} \neq \text{NP}$. Mivel így $\text{P/poly} \neq \text{NP}$ és $\text{P} \subseteq \text{P/poly}$.

Definíció 5.6

$$\text{PARITY}(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} x_i \pmod 2$$

$$\text{MAJORITY}(x_1, x_2, ..., x_n) = \begin{cases} 1 & \text{ha } \sum x_i > n/2 \\ 0 & \text{különben} \end{cases}$$

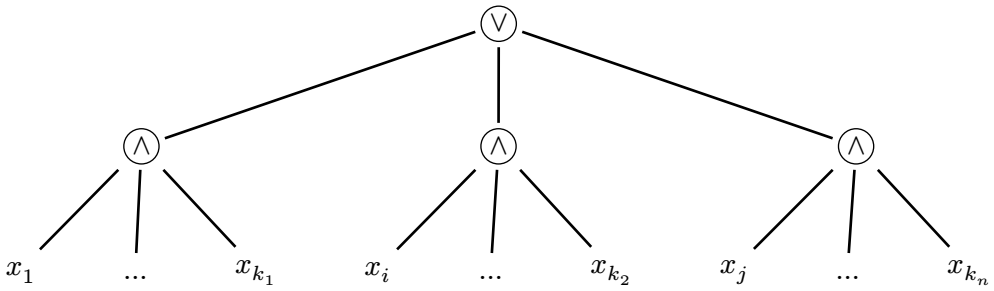
$$\text{THRESHOLD}_k(x_1, x_2, ..., x_n) = \begin{cases} 1 & \text{ha } \sum x_i > k \\ 0 & \text{különben} \end{cases}$$

$$\text{MOD}_m(x_1, x_2, ..., x_n) = \begin{cases} 1 & \text{ha } \sum x_i \not\equiv 0 \pmod m \\ 0 & \text{különben} \end{cases}$$

Megjegyzés

2-mély Boole-hálókra exponenciális méret kell a PARITY kiszámolására.

Proof: Mivel a PARITY egy Boole-függvény ezért fel lehet írni konjunktív normál formába, így ezt a konjunktív normál formát ki tudjuk számolni Boole-hálóval. A következő ábra illusztálja hogy hogyan nézhet ki egy CNF hálója.



Mivel azon n hosszú bemenetek száma melyekre 1-et ad a PARITY-t kiszámoló háló 2^{n-1}, azaz

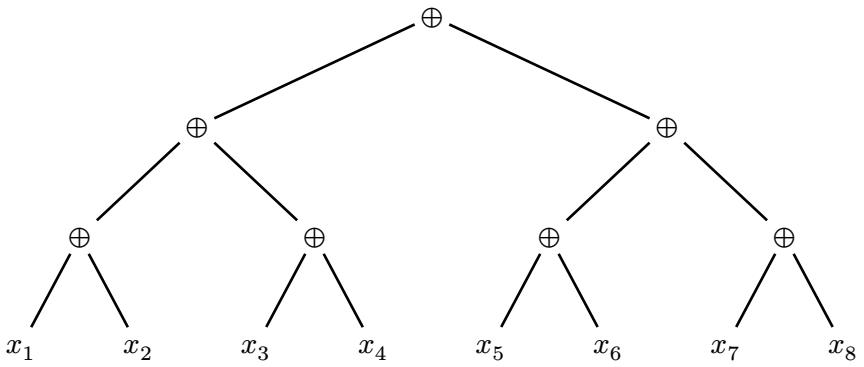
$$\left|\text{PARITY}_n^{-1}(1)\right| = 2^{n-1},$$

ezért nem lehet kevesebb mint 2^n csúcsa a hálónak. □

Megjegyzés

A PARITY nyelvre $\log n$-nél mélyebb Boole-hálóra nem megy a szuper-polinomiális alsó becslés.

Proof: Jelöljük a $\bmod 2$ összeadást $\oplus$-al, azaz a bináris XOR műveletet. Ekkor, ezekkel a $\oplus$ kapukkal tudunk alkotni egy $\log n$ mély Boole-hálót, aminek a mérete $O(n)$ a következő módon:



□

Megjegyzés

A PARITY nyelvet $\log n / \log\log n$ mélységben is ki lehet számolni a $\wedge, \vee, \neg$ jelekkel.

Proof: TODO □

Állítás 5.7

$$\log_{\log n} n = \frac{\log n}{\log\log n}$$

Proof: A logaritmus alap átváltás formula révén

$$\log_a b = \frac{\log_c a}{\log_c b} \Longrightarrow \log_{\log n} n = \frac{\log n}{\log\log n}.$$

□

Tétel 5.8 **Yao–Håstad–Smolensky**

A PARITY-t kisszámoló $\wedge, \vee, \neg$ kapukat használó k-mély Boole-háló mérete legalább

$$2^{\frac{1}{2}n^{\frac{1}{2k}}}.$$

Megjegyzés

A bizonyítás a következő cikk alapján hangzott el előadáson: https://dl.acm.org/doi/pdf/10.1145/28395.28404

6 Párhuzamos algoritmusok

Példa

Feladat: **input:** $a_1, ..., a_n \in \mathbb{N}$, **output:** $\max a_i$

n^2 processzorral meg tudunk adni egy $O(1)$ algoritmust.

Csinálunk egy b auxillary tömböt. Minden $i \neq j$ index párosra egy processzorral kiszámoljuk $a_i \overset{?}{<} a_j$-t. Ha $a_i < a_j$, akkor $b[i] = *$. Egy lépés alatt az n^2 processzor elvégzi az összes összehasonlítást. A végén csak a maximális elemnél nem lesz $*$.

Csak akkor szeretne ugyanoda írni két processzor ha ugyanazt akarják írni, így ez nem lesz probléma.

Példa

Feladat: **input:** $a_1, ..., a_n \in \mathbb{N}$, **output:** $\max a_i$

n processzorral meg tudunk adni egy $O(\log \log n)$ algoritmust.

Bontsuk fel az n elemet $\sqrt{n}$ darab $\sqrt{n}$ méretű részre. Ha kiszámultuk mindegyik rész maximumját, akkor már csak a kapott $\sqrt{n}$ maximum maximumját kell kiszámolnunk, amit n processzorral tudunk $O(1)$ időben.

Legyen $t(n)$, hogy mennyi idő n elem maximumát kiszámolni n processzorral. Most láttuk, hogy $t(n) = t(\sqrt{n}) + 1$. Ezt a rekurziót kifejtve azt kapjuk, hogy $t(n) = t\left(n^{\frac{1}{2^i}}\right) + i$.

Ha $i = \log \log n$, akkor $n^{\frac{1}{2^i}} = 1$. Tehát $t(n) = O(\log \log n)$.

Definíció 6.1 **PRAM**

A PRAM-nek van n processzora és mindegyik processzor egy külön RAM, amit processzornak hívunk, és van egy közös (globális) memóriája. A közös memóriát egész számokkal indexelünk és M-nek jelöljük, tehát a memória cellái: $..., M[-2], M[-1], M[0], M[1]1, M[2], ...$

A közös memóriába tetszőleges egész számok írhatóak, tehát $\forall i M[i] \in \mathbb{Z}$.

Egy számítási lépés a következő három szakaszból áll:
1. Mindegyik processzor olvashat legfeljebb 1 cellát M-ből.
2. Mindegyik processzor számolhat.
3. Mindegyik processzor írhat legfeljebb 1 cellát m-be.

A közös memóriába konkurrens írásokat hogyan oldunk fel?
- Illegális egyszerre több processzornak ugyanabba a globális memória címbe írnia.
- Csak akkor megengedett ha a konkurrens írások megegyeznek.
- Mindegyik processzornak van egy prioritása és a konkurrens írásoknál a legmagasabb prioritású processzor írása marad.

A PRAM inputja M-ben adott és ide is várjuk az output-ot.

A PRAM egy lépését annak számoljuk, hogy a processzorok párhuzamosan olvasnak, számítanak, és írnak egyet.

Megjegyzés

Jelölje egy adott feladat megoldására szükséges időt p processzorral t_p. Ekkor $\frac{t_1}{p} \leq t_p$.

Proof: Tegyük fel, hogy $pt_p < t_1$, ez azt jelenti, hogy ha egy processzorral szimuláljuk p processzor párhuzamos működését az gyorsabb lenne mint t_1, de pont úgy definiáltuk t_1-et, hogy ez a minimális idő hogy 1 processzorral megoldjuk a feladatot. □

Következmény 6.2

Az előző miatt egy exponenciális algoritmushoz legalább exponenciálisan sok processzor kell, hogy polinomiális idejű algoritmust kapjunk.

Definíció 6.3 **NC**

Ha egy L feladatot $n^{O(1)}$ processzorral $\log^{O(1)} n$ időben ki tudok számolni, akkor az L feladat NC-ben van (Nick's Class).

Példa

Két n hosszú vektor skaláris szorzata n processzorral $\log n$ időben.

$a, b \in \mathbb{N}^n$, kell $a \cdot b \in \mathbb{N}$.

Proof: $a_i b_i$ szorzatok $O(1)$. Összeadások mint egy teljes bináris fa -> $\log n$ □

Példa

Mátrix-mátrix szorzás

$A \in \mathbb{N}^{n \times m}$, $B \in \mathbb{N}^{m \times k}$, kell $AB = C \in \mathbb{N}^{n \times k}$.