ALGORITMUSOK ÉS BONYOLULTSÁGELMÉLET

Matematika MSc hallgatók számára

Kitekintő (olvasnivaló) előadás

Előadó: Hajnal Péter
Jegyzetelő: Hajnal Péter 2013.

1. Párhuzamos számítások, $\mathcal{NC}$

Az alábbi nyelvosztály bevezetésének motivációja, hogy a hatékonyan párhuzamosan kiszámítható/eldönthető problémák fogalmát szerették volna leírni. A definícióban a $\mathcal{P}$ nyelvosztálynak uniform hálózatsorozattal való leírásához adunk további feltételeket.

Definíció. Legyen $\mathcal{NC}$ azon L nyelvek osztálya, amihez létezik $L^{\text{bin}} \subset \{0,1\}^*$ bináris kódolás és $\{C_k\}_{k\in\mathbb{N}}$ hálózatsorozat, amelyre

- Adott ω input esetén, a méretéhez tartozó C_k hálózat $\mathcal{L}$-ben megkonstruálható,
- $\{C_k\}_{k\in\mathbb{N}}$ mérete polinomiális,
- $\{C_k\}_{k\in\mathbb{N}}$ mélysége polilogaritmikus,
- $\omega \in \{0,1\}^n$ akkor és csak akkor van L^{bin}-ben, ha $C_n(\omega) = 1$.

Az $\mathcal{NC}$ osztály finomítható, az alapján hogy a hálózatsorozat mélységsorozata $\log n$ milyen fokú polinomjával becsülhető.

Definíció. Legyen $\mathcal{NC}^i$ azon $L \in \mathcal{NC}$ nyelvek osztálya, amihez az $\mathcal{NC}$-hez tartozást bizonyító $L^{\text{bin}} \subset \{0,1\}^*$ bináris kódolás és $\{C_k\}_{k\in\mathbb{N}}$ hálózatsorozat olyan, hogy $\{C_k\}_{k\in\mathbb{N}}$ mélységére alkalmas α, β konstansokkal $\alpha \log^i k + \beta$ alakú felső becslés adható.

A bevezetett $\mathcal{NC}$ osztályról az első fontos eredményeket Nick Pippinger bizonyította. Ennek alapján javasolták, hogy az osztály legyen „Nick osztálya”, angolul Nick's Class. Innen ered az $\mathcal{NC}$ elnevezés.

Az új osztály viszonyát a korábbiakhoz az alábbi tételben foglaljuk össze.

1. Tétel.

$$\mathcal{NC}^1 \subset \mathcal{L} \subset \mathcal{NL} \subset \mathcal{NC}^2 \subset \mathcal{NC} \subset \mathcal{P}.$$

Bizonyítását az érdeklődő olvasóra bízzuk.

Természetesen, ha a hálózatainkat több output kapuval látjuk el, akkor a megfelelő kiszámítási feladatok osztályait is definiálhatnánk. Ezeket f-$\mathcal{NC}$, illetve f-$\mathcal{NC}^i$-vel jelöljük.

Néhány példával (esetünkben éppen kiszámítási feladatokkal) mutatjuk az osztályunk erejét.

Példa. Az összeadás, kivonás f-$\mathcal{NC}^1$-be esik.

Példa. Az összeadás, kivonás (számaink legyenek bináris számrendszerben kódolva) f-$\mathcal{NC}^1$-be esik.

Egy bizonyító hálózat tervezét az érdeklődő olvasóra bízzuk.

Példa. n darab n-bites szám összeadása f-$\mathcal{NC}^1$-be esik.

Számainkat csoportosítsuk hármas csoportokba. Az egy csoportba eső három szám adott helyiértékű három számjegyének összege $0, 1, 2$ vagy 3. Így két bit „hatása" lesz: a sajáthelyi értékére hat, illetve maradékot szolgáltat(hat) az eggyel nagyobb helyiértékhez. A hivatkozott bitek mindegyik három másiktól függ, konstans mélységben kiszámolható. A két hatást külön kezelve két $(n+1)$-bites szám összeadásával helyettesíthető az eredeti hármas összeg. Rekurzíven használva az ötelete (iterálva) egy logaritmikus méretű hálózat két szám összeadására vezeti vissza a kérdést, ami logaritmikus méretben megoldható.

Példa. Egy $n \times n$ méretű n-bites számokat tartalmazó mátrix nyomának (főátlójára eső elemeinek összege) kiszámolása f-$\mathcal{NC}^1$-be esik.

Megjegyezzük, hogy a nyom egy fontos paramétere a mátrixoknak. Egyik alternatív leírása, hogy a sajátértékeinek összege. Általában a sajátértékek komplex számok és kiszámításuk sok kérdést vet fel. Összegük kiszámolása azonban triviális.

Példa. Két darab n-bites szám szorzása f-$\mathcal{NC}^1$-be esik.

Az előző példa és a standard szorzási eljárás alapján nyilvánvaló.

Példa. Két darab $n \times n$ méretű n-bites számokat tartalmazó mátrix szorzása f-$\mathcal{NC}^1$-be esik.

n^3 darab $a_{ij}a_{jk}$ alakú szorzatot kell kiszámolni. Ezt hálózatunk ugyanazon mélységében (az input felett logaritmikus magasságban) megtörténik. Azaz párhuzamosan megtehető. Majd n tagú összegeket számolunk ki, ami a korábbi trükkel szintén logaritmikus mélységben megtehető.

2. Lemma. *Legyen M egy $n \times n$ méretű mátrix. Ekkor az $M, M^2, M^3, \ldots, M^n$ mátrix hatványok f-$\mathcal{NC}^2$-ben kiszámolhatók.*

Bizonyítás. A hatványozást több fázisban végezzük el. M^2, majd M^3, M^4, majd M^5, M^6, M^7, M^8 kiszámítása történik és így tovább. Minden fázisban az egyes mátrixhatványok párhuzamosan (a hálózat ugyanazon mélységében) történnek és csak két korábban már kiszámolt mátrixhatvány összeszorzását kivánják. Az n-edik hatvány eléréséhez a fázisok száma logaritmikus, egy fázis megvalósításához is logaritmikus mélység kell. Így hálózatunk mélysége $\log^2$ nagyságrendű. ■

Ezen egyszerű párhuzamos mátrixhatványozási trükknek két következményét is megemlítjük.

3. Következmény. *Legyen M egy alsó trianguláris mátrix. tegyük fel, hogy a főátlón nem-nulla elemek vannak, azaz a diagobális elemeknek létezik inverze. Ekkor az M^{-1} inverzmátrix f-$\mathcal{NC}^2$-ben kiszámolható.*

Bizonyítás. Egyszerű meggondolni, hogy M

$$\begin{pmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \mu_{2,1} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{n,n} & \mu_{n,2} & \cdots & 1 \end{pmatrix}$$

alakban írható fel és így elég meggondolni, hogy az inverz hogyan számolható ki, ha a főátlón csupa 1-esek szerepelnek, azaz $M = I - M_0$. Ez a korábbiak és a következő képlet alapján könnyen megtehető:

$$(I - M_0)^{-1} = I + M_0 + M_0^2 + \ldots + M_0^{n-1}.$$

(Ne feljtsük el, hogy M_0 főátlóján és felette már csak 0-k szerepelnek. Így a M_0^i mátrixban a főátló mellett, az alatta következő $i-1$ darab átlón is csak 0-k lesznek. Speciálisan azaz $M_0^n = M_0^{n+1} = \ldots = 0$.) ■

4. Következmény. *Legyen M egy természetes számokat tartalmazó $n \times n$ méretű mátrix. Sajátértékei legyenek $\lambda_1, \lambda_2, \ldots, \lambda_n$. Ekkor ezen számok hatványösszegei, Newton-szimmetrikuspolinomjai:*

$$N_i(\lambda_1, \lambda_2, \ldots, \lambda_n) = \lambda_1^i + \lambda_2^i + \ldots + \lambda_n^i$$

f-$\mathcal{NC}^2$-ben kiszámolhatók.

Bizonyítás. M^i sajátértékei $\lambda_1^i, \lambda_2^i, \ldots, \lambda_n^i$. Így a sajátertékek i-edik hatványösszegének kiszámításához M^i nyomának kiszámítása szükséges. Ez különböző i-kre párhuzamosan megtehető (miután M hatványai kiszámítottak). ■

Következményeinknek érdekes következményei vannak, ha a következő szimmetrikus polinomokra vonatkozó alaperedményt ismerjük. Ehhez egy definícióra van szükségünk.

Definíció. Legyen E_i az i-edik ($i = 1, 2, \ldots, n$) elemi szimmetrikus polinom az $x_1, x_2, \ldots, x_n$ változókkal:

$$E_i(x_1, x_2, \ldots, x_n) = \sum_{R \in \binom{[n]}{i}} \prod_{j \in R} x_j.$$

A következő Newton—Girard-formulaként ismert tételt bizonyítás nélkül ismertetjük.

5. Tétel. *Az $\{E_i(x_1, x_2, \ldots, x_n)\}_{i=1}^n$ polinomok kifejezhetők az $\{N_i(x_1, x_2, \ldots, x_n)\}_{i=1}^n$ polinomokkal az alábbi képletek szerint:*

$$\begin{aligned}
E_1 &= N_1, \\
2E_2 &= E_1 N_1 - N_2, \\
3E_3 &= E_2 N_1 - E_1 N_2 + N_3, \\
4E_4 &= E_3 N_1 - E_2 N_2 + E_1 N_3 - N_4, \\
&\vdots
\end{aligned}$$

A tétel alkalmazásaként aláhúzzuk, hogy a Newton és elemi szimmetrikuspolinomok közötti kapcsolatot egy alsó trianguláris mátrix-szal írhatjuk le. Ennek inverzét kiszámolhatjuk $\mathcal{NC}^2$-ben. Így az elemi szimmetrikuspolinomokat is ki tudjuk értékelni a sajátértékeken. Így ki tudjuk számolni a mátrix karakterisztikus polinomját, speciálisan determinánsát. Cramer-szabály alapján a mátrix inverze is adódik (amennyiben invertálható). Kapjuk a következő tételt.

6. Tétel. *Adott egy M négyzetesmátrix. Ekkor a következő problémák f-$\mathcal{NC}^2$-be esnek.*

(i) *M karakterisztikus polinomjának kiszámítása,*

(ii) $\det M$ *kiszámítása,*

(iii) *M^{-1} kiszámítása (feltéve, hogy M invertálható),*

(iv) *Adott b vektor esetén az $M \cdot \vec{x} = b$ egyenletrendszer megoldása (feltéve, hogy M invertálható).*

A fenti tételnek sok független bizonítása született, általában a párhuzamos algoritmusok vizsgálata előtt. Például P.A. Samuelson Nobel-díjas közgazdász egyik lineáris algebrai módszere is alkalmas a párhuzamosításra. A fenti megoldás U. Le Verreir módszerének párhuzamosítása. Ezt a lehetőséget L. Csanky vette észre. (U. Le Verrier számításai vezettek a Neptunusz felfedezéséhez, neve egyik az Eiffeltoronyba gravirozott 72 névnek.)

A fentieknek ezen alaptétel egy gráfelméleti következményét említjük meg. Ehhez emlékeztetünk egy problémára.

Emlékeztető. PÁROS-GRÁF-TELJES-PÁROSÍTÁS-TESZT azon páros gráfokat tartalmazó nyelv, amelyekben van teljes párosítás.

Diszkrét matematika előadáson szerepelt egy véletlen algoritmus ennek megoldására: Írjuk fel az alsó-felső csúcs illeszkedési mátrixot és minden 1-est helyettesítsünk $\{1, 2, \ldots, N\}$ egy uniform eloszlású véletlen elemével. Ha a kapott mátrix determinánsa nem-nulla, akkor biztosak lehetünk, hogy gráfunkban van teljes párosítás. Ha a determináns értéke nulla, akkor nagy bizonyossággal állíthatjuk, hogy gráfunkban nincs teljes párosítás (feltéve, hogy N-et alkalmasan nagynak választottuk).

A fenti (Lovász Lászlóhoz fűzhető) algoritmus jelentősége abban rejlik, hogy a determinisztikus része minden probléma nélkül párhuzamosítható. Az érdeklődő olvasó definiálhatja az $\mathcal{NC}$ osztály véletlen változatát ($\mathcal{RNC}$). Mi csak megemlítjük a következő tételt.

7. Tétel (Lovász László). *PÁROS-GRÁF-TELJES-PÁROSÍTÁS-TESZT$\in \mathcal{RNC}$.*

Hogy a fenti nyelv $\mathcal{NC}$-hez tartozik vagy nem, az a párhuzamos számítások elméletének egy fontos megoldatlan problémája.

2. Interaktív bizonyítások

$\mathcal{NP}$ egyik értelmezésében egy bizonyító és egy ellenőrző szereplő van. A bizonyítás egy üzenet váltás: a bizonyító elküld egy bizonyítékot/tanút, amit az ellenőrző determiszintikusan elbírál. Mi történik, ha az interakció nem egyetlen egy üzenet cseréje, illetve, ha az ellenőrző kezébe véletlen számokhoz való hozzáférést biztosítunk? Ez a kérdés vezet el az interaktív bizonyítások fogalmához.

Definíció. Az L nyelv pontosan akkor tartozik az $\mathcal{IP}$ nyelvosztályhoz, ha van két (véletlen) Turing-gép B (bizonyító) és E (ellenőrző) Turing-gép, amely a következőket „tudja”:

(o) B és E (munkaszalagjuk és véletlen bitejeiket tartalmazó szalag mellett) tartalmaz egy közös csak írható, nem törölhető kérdésszalagot, illetve válaszszalagot. E (ellenőrző) a kérdésszalagot írhatja, a válaszszalagot csak olvashatja. Továbbá van egy speciális „?" állapota. B a kérdésszalagot csak olvashatja, a válaszszalagot írhatja. B „érzékeli", ha E állpota ? lesz.

E futása természetes módon definiálható a ? állapot szerepének tisztázása után. Ekkor B a kérdésszalag (előző kérdés után írt) karaktereit kérdésnek fogja fel és a válaszszalagra ír egy karaktersorozatot. E ezt olvashatja, „válaszként fogja fel" és futását folytatja.

(i) Az ω input és a kérdésszalag κ tartalma függvényében a válaszszalagra B által leírt választ egy $B : (\omega, \kappa) \mapsto \nu \in \Sigma^*$ függvény írja le. A futásba B komplexitása nam számít be (ahogy $\mathcal{NP}$-hez sem kellett egy jó tanú generálásának bonyolultság'aval fogalkoznunk), csupán a válasz leírása (ami „könyvelhető" az olvasásra is).

(ii) E polinomiális lépés után leáll ELFOGAD vagy ELVET állapottal.

(ii) E elfogadja az L nyelvet, azaz

- $\omega \in L$ esetén van olyan B, ami amire $\mathbb{P}[E(\omega, \rho) = ELFOGAD] \geq 2/3$,
- $\omega \notin L$ esetén tetszőleges B-re $\mathbb{P}[E(\omega, \rho) = ELVET] \geq 2/3$.

Ha algoritmusunknak több szereplője van, akkor gyakran protokolként hivatkozunk rá.

A következő fogalom az egyik alapprotokol példája.

Definíció. Legyen IZOMORFIZMUS az a nyelv, ami azon (G, H) gráfpárokat tartalmazza, amelyekre G és H izomorfak, azaz $G \simeq H$.

Legyen NEM-IZOMORFIZMUS a komplementer nyelv, azaz az amely azon (G, H) gráfpárokat tartalmazza, amelyekre G és H nem izomorfak, azaz $G \not\simeq H$.

IZOMORFIZMUS nyilván $\mathcal{NP}$-beli. NEM-IZOMORFIZMUS nyelvhez tartozás igazolására nem ismert $\mathcal{NP}$-beli eljárás. A következő protokol egy $\mathcal{IP}$ igazolási eljárást mutat.

Példa. Legyen G, H két nem izomorf gráf. A Bizonyító szeretné erről az Ellenőrző-t meggyőzni. Az Ellenőző veszi a (G, H) gráfpár kódolását két szomszédsági mátrixszal. Az alábbiakban leírjuk E-t.

A kódot „megcsavarja" az eredetihez (inputszalagon lévőhöz) képest a sorok (vele összehangoltan az oszlopok) véletlen permutálásával. Majd a gráfpár sorrendjét 1/2 valószínűséggel megtartja, 1/2 valószínűséggel megcseréli. Az így kapott gráfpárt odaadja a bizonyítónak: „Mondja meg melyik G és melyik H." Azaz mondja meg, hogy az utolsó véletlen bit megcseréltette-e vele az eredeti sorrendet vagy nem. Ha az ellenőrző nem találja el cselekedetét, akkor nem fogadja el a nyelvhez tartozást. Ha az ellenőrző eltalálja cselekedetét, akkor megismétli üzenetét (új, azaz független véletlen csavarással és sorrenddel). Ha most is eltalálja, hogy sorrendet cserélt vagy nem, akkor elfogadja, hogy a két gráf nem izomorf.

A fenti protokol nyilván a NEM-IZOMORFIZMUS $\mathcal{IP}$-beliségét bizonyítja. Ha a (G, H) a nyelvhez tartozik és a Bizonyító jogkövető, akkor nem izomorfizmus esetén a protokol biztos elfogadtatja azt az Ellenőrzővel. Ha a (G, H) nem tartozik a nyelvhez, akkor a Bizonyító bármit tesz, akkor el kell találni a két forduló utolsó véletlen bitjét. A siker valószínűsége legfeljebb 1/4.

A következő példa már komolyabb eszköztárat igényel.

Definíció. Legyen MULTILINEAR-VALUE-SUM azon (p, k) párok által alkotott nyelv, ahol $p(x_1, x_2, \ldots, x_n)$ egy multilineáris polinom egy $\mathbb{F}$ test felett és $k = \sum_{(e_1,e_2,\ldots,e_n)\in\{0,1\}^n} p(e_1, e_2, \ldots, e_n)$. Azaz k a bináris behelyettesítéseknél kapott értékek összege. p-ről azt tesszük fel, hogy az ellenőrző ki tudja számolni helyettesítési értékeit (azaz nem szükségszerűen monomok összegeként van felírva).

Példa. Legyen

$$p_i(x_1, x_2, \ldots, x_i) = \sum_{(e_{i+1},e_{i+2},\ldots,e_n)\in\{0,1\}^{n-i}} p(x_1, x_2, \ldots, x_i, e_{i+1}, \ldots, e_n).$$

p_0 egy szám, amely értékét az input tartalmazza (azt gondoljuk, hogy a Bizonyító szolgáltatta), ennek korrekt értékéről szeretne az Ellenőrző megbizonyosodni. Lássuk a protokolt.

A Bizonyítótól elkéri a $p_1(x_1)$ polinomot. Ha $p_1(0) + p_1(1) = k$, akkor azt mondja hogy „p_1 konzisztens p_0-lal". (Konzisztenciateszt a tétel és az első válasz között.) Ezek után vesz egy r_1 uniform eloszlású véletlen elemét $\mathbb{F}$-nek és elkéri a $p_2(r_1, x_2)$ polinomot. Ismét egy teszt következik: $p_1(r_1) = p_2(r_1, 0) + p_2(r_1, 1)$ egyenlőség tesztelése ismét egy konzisztenciatesz (az első két válasz között). Majd generál egy r_2 véletlen elemet $\mathbb{F}$-ből és elkéri a $p_3(r_1, r_2, x_3)$ polinomot. Ismét egy teszt következik: $p_2(r_1, r_2) = p_3(r_1, r_2, 0) + p_3(r_1, r_2, 1)$ egyenlőség tesztelése ismét egy konzisztenciateszt (a második és harmadik válasz között). A protokol így megy az utolsó $p_n(r_1, r_2, \ldots, r_{n-1}, r_n)$ kérés konzisztenciájának ellenőrzéséig az előző válasszal. Ekkor azonban az Ellenőrző már maga ki tudja számolni a kért számot és megbizonyosodhat a Bizonyító korrektségéről. Az ellenőrző akkor és csak akkor fogad el, ha minden konzisztenciateszten és az utolsó ellenőrzésen is átment a „beszélgetés".

Ha $\omega \in L$, akkor egy jogkövető bizonyító esetén az ellenőrző biztos elfogad. A kérdés, hogy milyen valószínűséggel fogad el az Ellenőrző egy hamis tételt. Ehhez az utolsó kérdésre persze jót kellett válaszolnia a Bizonyítónak. Tehát volt egy i, hogy az i-edik kérdésre egy $\widetilde{p_i}(r_1, \ldots, r_{i-1}, x_i) \neq p_i(r_1, \ldots, r_{i-1}, x_i)$ polinomot felelt, de $i + 1$-edik válasza már a korrekt $p_{i+1}(r_1, \ldots, r_i, x_{i+1})$ volt. $\widetilde{p_i}$ bejelentése után generálta az ellenőrző az r_i véletlen testelemet. $\widetilde{p_i}(r_1, \ldots, r_{i-1}, x_i) - p_i(r_1, \ldots, r_{i-1}, x_i)$ egy nem-nulla, egy határozatlanú lineáris polinom. Így legfeljebb egy helyen vesz fel 0 értéket. Azaz legfeljebb egy helyen egyezik meg a p_i és $\widetilde{p_i}$ polinom. Az Ellenőrző az aktuális konzisztenciateszben $p_{i+1}(r_1, \ldots, r_i, 0) + p_{i+1}(r_1, \ldots, r_i, 1) = p_i(r_1, \ldots, r_i)$-t számolta ki és $\widetilde{p_i}$ értékével vetette össze. Legfeljebb $1/|\mathbb{F}|$ a valószínűsége, hogy a bizonyító „nem bukik le". A hibás elfogadás valószínűségét $n/|\mathbb{F}|$-fel becsülhetjük. (A „hibázás" eseményt felülről becsli a „valamelyik i-re bekövetkezik a szerencsétlen r_i választás" esemény.) Azaz, ha $|\mathbb{F}| \geq \frac{3}{2}n$, akkor MULTILINEAR-VALUE-SUM$\in \mathcal{IP}$.

Megjegyzés. A multilinearitás feltételét könnyű relaxálni. Ha azt tesszük fel, hogy polinomunk minden monomjában minden változó kitevője legfeljebb d, akkor is igaz a megfelelő nyelv $\mathcal{IP}$-be esése elég nagy $\mathbb{F}$ test esetén. (Csak az r_i választásánál legfeljebb d szerencsétlen érték lesz, a hibázás valószínűsége legfeljebb $nd/|\mathbb{F}|$.)

Definíció. $\sharp$-3CNF az a nyelv, ami azon (φ, k) párokat tartalmazza, amelyekre φ egy 3CNF formula és k a kielégítő kiértékelések száma.

Példa. $\sharp$-3CNF $\in \mathcal{IP}$.

Ennek igazolásához egy alkalmas p_φ korlátos fokú polinomot és $\mathbb{F}_q$ testet adunk, amire $\sum_{(e_1,e_2,\ldots,e_n)\in\{0,1\}^n} p_\varphi(e_1, e_2, \ldots, \ldots, e_n) = k \pmod q$. q egy 2^n-nél nagyobb prímszám (ezt az Ellenőrző ellenőrizheti). Ha a polinom konstrukcióját megadtuk, akkor a fenti protokolból következik az állítás.

Tehát feladatunk a φ logikai formula „aritmetizálása". Példákkal szemléltetjük a konstrukciót. Először egy-egy klózt aritmetizálunk. $x \vee y \vee \neg z$-t helyettesítsük $1-(1-x)(1-y)z$-vel. $\neg x \vee w \vee \neg a$-t helyettesítsük $1-x(1-w)a$-val. A klózok aritmetizálása után p_φ legyen a klózoknak megfelelő polinomok szorzata. Minden változóban a fok a klózok számával becsülhető.

A következő tétel alapvető jelentőségű. Bizonyítására már nincs időnk.

8. Tétel.

$$\mathcal{IP} = \mathcal{PSPACE}$$

3. Véletlen belepillantással ellenőrizhető bizonyítások, $\mathcal{PCP}$

Definíció. Vegyünk egy tanúszalagos Turing-gépet, aminek egy véletlen bitszalagja is van. A tanúszalagja különleges: egyes karakterei felett elhaladva a gép nem látja azt csak ha speciális tanú-olvasó állpotban van.

Mikor számít ki egy ilyen gép egy L nyelvet?

Definíció. A fenti gép akkor és csak akkor fogad el L nylevet, ha

(i) $\omega \in L$ esetén alkalmas τ tanúszalag-tartalomra a gép biztos ELFOGAD.

(ii) $\omega \notin L$ esetén minden τ tanúszalag-tartalomra a gép legalább 1/2 valószínűséggel ELVET.

A számítási modell több paraméterét is használhatjuk bonyolultsági osztályok bevezetésére.

Definíció. $\mathcal{PCP}[r(n), q(n)]$ azon L nyelveket tartalmazza, amelyhez van olyan polinomiális idejű elfogadó gép, ami n hosszú inputok esetén $r(n)$ véletlen bitet használ és a tanúszalagot legfeljebb $q(n)$-szer olvassa el.

Paraméterek alkalmas választásával korábban bevezetett osztályok alternatív leírásait kapjuk.

Feladat. (i) $\mathcal{PCP}[0,0] = \mathcal{P}$,

(ii) $\mathcal{PCP}[O(log(n)), 0] = \mathcal{P}$,

(iii) $\mathcal{PCP}[0, O(log(n))] = \mathcal{P}$,

(iv) $\mathcal{PCP}[poly(n), 0] = \mathrm{co}\mathcal{RP}$,

(v) $\mathcal{PCP}[0, poly(n)] = \mathcal{NP}$.

Különösen fontos az $r(n) = \mathcal{O}(\log n)$, $q(n) = \mathcal{O}(1)$ paraméter választás. Az erre vonatkozó tétel a bonyolultságelmélet egyik eddigi legnagyobb eredménye.

9. Tétel (S. Arora, C. Lund, R. Motwani, M. Sudan, Szegedy Márió). $\mathcal{PCP}[\mathcal{O}(\log n), \mathcal{O}(1)] = NP$.

A tétel nagyon meglepő. A gép/ellenőrző csupán konstans karaktert olvas el a tanúszalagról (természetesen ezek véletlen pozíciók). Ezek után jelenti be nagy bizonyossággal az input viszonyát az L nyelvhez. A bizonyítása hosszú, évtizedes munka csúcseredménye (amelyben résztvevők kutatókból csak egy rövid válogatást adunk U. Feige, S. Goldwasser, Lovász László, S. Safra, S. Micali, C. Rackoff, J. Kilian).

A tételnek messzemutató következményei vannak. $\mathcal{NP}$ eredeti leírása elvezetett a HÁLÓZAT-SAT, SAT, illetve 3SAT $\mathcal{NP}$-teljességéhez. Azaz egy $\mathcal{NP}$-beli nyelvhez tartozást áttranszformáltunk egy megfelelő CNF formula kielégíthetőségének eldöntésébe. Azaz arra a kérdésre vezettük vissza, hogy van-e minden klózát kielégítő kiértékelés vagy legfeljebb klózainak számánál eggyel kevesebb klóza elégíthető ki egyszerre. Az új értelmezés szinte ugyanazon az úton új $\mathcal{NP}$-teljességi eredményeket kapunk. Azaz egy $\mathcal{NP}$-beli nyelvhez tartozást áttranszformálhatunk egy megfelelő CNF formulára, amely igen speciális lesz: vagy minden klóza igazzá tehető (kielégíthető), vagy klózainak legfeljebb 90%-a lesz kielégíthető. Az eredeti nyelvhez tartozás ekvivalens a két lehetőség közötti megkülönböztetéssel. Ezáltal új típusú $\mathcal{NP}$-teljes problémákhoz jutunk.

Definíció. Legyen ϵ-APPROX-MAX-SAT az a probléma ami elfogadja a kielégíthető CNF-eket és elveti azokat, amelyek klózainak legfeljebb $1 - \epsilon$ része kielégíthető bármely kiértékelés által. A többi CNF-en tetszőleges eredményt elfogadunk.

10. Tétel. *ϵ-APPROX-MAX-SAT $\mathcal{NP}$-teljes, ha ϵ elég kicsi.*

Bizonyítás. Legyen L egy tetszőleges $\mathcal{NP}$, azaz $\mathcal{PCP}[\mathcal{O}(\log n), \mathcal{O}(1)]$-beli nyelv. Legyen ρ egy lehetséges tartalma a véletlen bitek szalagjának. Ezt ismerve tudjuk, hogy gépünk a tanúszalag τ tartalmának mely c (konstans) karakterét olvasva jelenti be döntését. Az ω-tól függő döntést leírhatjuk legfeljebb 2^c klózt tartalmazó φ_ρ CNF-fel (változók az elolvasott tanú-karakterek). ρ értéke polinomiális sok lehet. A polinomiális sok φ_ρ formula ÉS-sel összekötve adja az ω-hoz rendelt formulát (melynek változóhalmaza már a tanúszalag teljes karaktersorozatát felöleli). Ha $\omega \in L$, akkor mindegyik φ_ρ-t kielégíti a tanúszalag megfelelő tartalma (ennek persze csak c értékétől függően). A teljes tanúszalag tartalma egy olyan kiértékelést ad, ami minden klózt kielégít. Ha $\omega \notin L$, akkor a φ_ρ formulák legalább felében a 2^c klóz közül legfeljebb $2^c - 1$-et elégíthet ki bármely értékelés. Ez azt jelenti, hogy az összes klóz közül legalább $1/2^{c+1}$-ed rész nem lesz kielégítve. Így L visszavezethető $(1 - \epsilon_L)$-APPROX-MAX-SAT-ra.

Ha L-et először SAT-ra vezetjük vissza, majd megismételjük a fenti gondolatmenetet, akkor $(1 - \epsilon)$-APPROX-MAX-SAT-ra való visszavezetést kapunk, ahol ϵ abszolút konstans (ϵ_{SAT}). ■

Definíció. MAX-SAT az a probléma egy CNF esetén annak a maximális klózszámnak a meghatározását kéri, ami egy kiértékeléssel kielégíthető.

11. Következmény. *Ha a MAX-SAT problémára létezik δ-közelítő polinomiális algoritmus (ahol δ elég kicsi), akkor $\mathcal{NP} = \mathcal{P}$.*

Bizonyítás. Tegyük fel, hogy létezik ilyen közelítő algoritmus. SAT-ot redukáljuk ϵ-APPROX-MAX-SAT problémára. Tegyük fel, hogy m a klózok számát jelöli. Azt kellene eldönteni, hogy az az értékelés, amley a lehető legtöbb klózt kielégíti (ezek számát jelöljük μ-vel) az m vagy pedig $(1-\epsilon)m$-nél nem nagyobb. Ha lenne olyan polinomiális algoritmus, ami egy $1/\Delta\cdot\mu$ és $\Delta\cdot\mu$ közötti számot ad ki, ahol $\Delta = \sqrt{\frac{1}{1-\epsilon}}$, akkor a SAT problémára kapnánk polinomiális algoritmust. Így az állítás teljesül $\delta < \Delta - 1 = \sqrt{\frac{1}{1-\epsilon}} - 1$ értékkel, ahol ϵ az előző tételben szereplő (ki nem számolt) konstans. ■