<u>הגדרה</u>: המחלקה $\mathbf{PCP}\left(r(n),q(n)\right)$ היא מחלקת כל השפות L עבורן יש פרוטוקול אינטראקטיבי בין מוכיח כל יכול לבודק הסתברותי.

עבור קלט $x\in\{0,1\}^n$, המוכיח כותב הוכחה $w\in\{0,1\}^{2^{r(n)}\cdot q(n)}$ ומכסה אותה, הבודק מטיל $r(n)$ מטבעות $y=y_1\cdots y_{r(n)}$, בוחר $q(n)$ ביטים שרוצה לקרוא $i_1,\ldots,i_{q(n)}$ וקורא אותם מההוכחה.

הוא מפעיל פרדיקט $V\left(\underbrace{x}_{\text{קלט}},\underbrace{y}_{\text{מטבעות}},w_{i_1},\ldots,w_{i_{q(n)}}\right)$ שמחליט אם לקבל או לדחות, כך ש:

$$x\in L \quad\Leftarrow\quad \underbrace{\exists w}_{\text{יש מוכיח}}\ .\ \underbrace{\forall y}_{\substack{\text{לכל בדיקה}\\ \text{של הבודק}}}\ .\ V(x,y,w)=1 .$$

$$x\notin L \quad\Leftarrow\quad \forall w.\Pr_y\left[V(x,y,w)=0\right]\geq 1/2$$ (כלומר לכל מוכיח, נתפוס אותו מרמה בהסתברות לפחות $1/2$).

משפט ה- PCP

$\mathrm{NP}\subseteq\mathrm{PCP}\left(O(\log n),O(1)\right)$ (ראינו בשיעור שעבר את הכיוון ההפוך).

נוכיח: $\mathrm{NP}\subseteq\mathrm{PCP}\left(\mathrm{Poly}(n),O(1)\right)$. בתוצאה הזאת, אורך ההוכחה $2^{\mathrm{Poly}(n)}$ אקספוננציאלי באורך הקלט.

1. מספיק להוכיח עבור שפה ספציפית שהיא NP-שלמה, שהיא גם שייכת למחלקה $\mathrm{PCP}\left(\mathrm{Poly}(n),O(1)\right)$.
2. נעבוד עם הבעיה QuadEq:

<u>הקלט</u>:

1. מטריצה בולאנית $A_{m\times n^2}$, נסמן ב-A_i את השורה ה-i של A, ונסמן $A_i=\left(a_{i1},\ldots,a_{in^2}\right)$.
2. וקטור $\vec{b}\in\{0,1\}^m$.

<u>הפלט</u>: הקלט $A,\vec{b}$ בשפה אם יש $u\in\{0,1\}^n$ כך ש-$A(u\otimes u)=\vec{b}$.

<u>קידוד</u>: $c:\{0,1\}^k\to\{0,1\}^{2^k}$.

התכונה של הקידוד תהיה: אם $u_1\neq u_2$, $u_1,u_2\in\{0,1\}^k$, אזי $c(u_1),c(u_2)$ שונים בלפחות חצי מהקואורדינטות.

הקוד: אם $u\in\{0,1\}^k$, $c(u)\in\{0,1\}^{2^k}$, ואז $\left(c(u)\right)_y=u\cdot y$ (כמו בשיעור הקודם).

<u>תכונה</u>: $\left(c(u)\right)_A+\left(c(u)\right)_B=\left(c(u)\right)_{A\oplus B}$.

למה? $\left(c(u)\right)_A+\left(c(u)\right)_B=u\times A+u\times B=\sum u_iA_i+\sum u_iB_i=$
$=\sum u_i(A_i+B_i)=u\cdot(A\oplus B)=\left(c(u)\right)_{A\oplus B}$.

לכן, אפשר להסיק כי $\left(c(u)\right)_A=\left(c(u)\right)_B+\left(c(u)\right)_{A\oplus B}$.

כזכור, יש 2^{2^k} מילים, ורק 2^k מילות קוד.

<u>טענה</u>: מילה אקראית, בהסתברות גבוהה, $1/4$-רחוקה מכל מילת קוד.

<u>משפט</u>: $\mathrm{QuadEq}\in\mathrm{PCP}\left(\mathrm{Poly}(n),O(1)\right)$.

נראה פרוטוקול $\mathrm{PCP}\left(\mathrm{Poly}(n),O(1)\right)$ לבעיה:

<u>קלט</u>: $A_{m\times n^2}$, $\vec{b}$.

מוכיח הוגן צריך לעשות:

אם $\langle A,\vec{b}\rangle\in\mathrm{QuadEq}$, אז יש $u\in\{0,1\}^n$ כך ש-$A(u\otimes u)=\vec{b}$.

לכן, המוכיח נותן $T_1=c(u)$, $T_2=c(u\otimes u)$.

הבודק:

בודק ש-T_1 היא 99/100-קרובה למילת קוד.

בודק ש-T_2 היא 99/100-קרובה למילת קוד.

בוחר $r_1,r_2\in\{0,1\}^n$, ובודק ש-$\left(T_2\right)_{r_1\otimes r_2}=\left(T_1\right)_{r_1}\cdot\left(T_1\right)_{r_2}$, $O\left(\log\frac{1}{\varepsilon}\right)$ פעמים.

בוחר $\alpha_1,\ldots,\alpha_m\in\{0,1\}$ באקראי, מחשב $C=\sum\alpha_i A_i$, בוחר $B\in_R\{0,1\}^{n^2}$, ובודק ש- $(T_2)_B+(T_2)_{B\oplus C}=\sum\alpha_i\vec{b}_i$. על השלב הזה חוזרים $O\left(\log\frac{1}{\varepsilon}\right)$ פעמים.

הבדיקה ש- $T\in\{0,1\}^{2^t}$ היא ε-קרובה למילת קוד:

נחזור על התהליך הבא $O\left(\log\frac{1}{\varepsilon}\right)$ פעמים:

נבחר $A,B\in\{0,1\}^t$, ונוודא ש- $T_A+T_B=T_{A\oplus B}$.

אם T באמת מילת קוד, אז יש u כך ש- $T=c(u)$, והבדיקה תמיד עוברת.

<u>משפט</u>: אם T δ-רחוקה ממילת קוד, אזי המבחן ייכשל בהסתברות לפחות δ.

<u>תכונה 1</u>: זהו באמת פרוטוקול $\mathrm{PCP}(\mathrm{Poly}(n),O(1))$.

הבודק מטיל $O(n^2)$ מטבעות, קורא $O\left(\log\frac{1}{\varepsilon}\right)=O(1)$ ביטים מההוכחה (את ε אנחנו קובעים, ונגיד ש- $\varepsilon=1/100$). אורך ההוכחה $2^{n^2}+2^n$.

שלמות:

נניח $(A,\vec{b})\in\mathrm{QuadEq}$, ונניח שהמוכיח הוגן. אזי יש $u\in\{0,1\}^n$ כך ש- $A(u\otimes u)=\vec{b}$.

המוכיח ייתן $T_1=c(u)$, $T_2=c(u\otimes u)$.

הבדיקה הראשונה תצליח.

הבדיקה השנייה: $(T_2)_{r_1\otimes r_2}\overset{?}{=}(T_1)_{r_1}\cdot(T_2)_{r_2}$.

$$(T_2)_{r_1\otimes r_2}=\left(c(u\otimes u)\right)_{r_1\otimes r_2}=(u\otimes u)\cdot(r_1\otimes r_2)=$$
$$=\sum_{1\le i,j\le n}(u\otimes u)_{i,j}\cdot(r_1\otimes r_2)_{i,j}=\sum_{i,j}u_i\cdot u_j\cdot(r_1)_i\cdot(r_2)_j=$$
$$=\left(\sum_i u_i\cdot(r_1)_i\right)\cdot\left(\sum_j u_j\cdot(r_2)_j\right)=(u\cdot r_1)\cdot(u\cdot r_2)=(c(u))_{r_1}\cdot(c(u))_{r_2}=(T_1)_{r_1}\cdot(T_1)_{r_2}$$

הבדיקה השלישית:

u מקיים $A(u\otimes u)=\vec{b}$. לכן, לכל $1\le i\le m$, $A_i\cdot(u\otimes u)=\vec{b}_i$.

לכן גם $\underbrace{\left(\sum\alpha_i A_i\right)}_{C}\cdot(u\otimes u)=\sum\alpha_i(A_i\cdot(u\otimes u))=\sum\alpha_i\vec{b}_i$.

מכיוון שהמוכיח הוגן, $(T_2)_C=\sum\alpha_i\vec{b}_i$.

לכן: $(T_2)_B+(T_2)_{B\oplus C}=(T_2)_C=\sum\alpha_i\vec{b}_i$.

לכן, המבחן תמיד יעבור. נקבל:

<u>תכונה</u>: אם הקלט בשפה והמוכיח הוגן, אז כל הבדיקות תעבורנה.

תקפות:

נניח כי $(A,\vec{b})\notin\mathrm{QuadEq}$, ונקבע מוכיח מרושע כלשהו, שנותן לנו $T_1\in\{0,1\}^{2^n}$, $T_2\in\{0,1\}^{2^{n^2}}$.

אם T_1 היא 1/100-רחוקה מכל מילת קוד, נתפוס אותו בבדיקה הראשונה בהסתברות $1-\varepsilon$ וסיימנו. בה"כ ניתן להניח כי T_1 היא 99/100-קרובה לאיזו מילת קוד $c(u)$.

אם T_2 היא 1/100-רחוקה מכל מילת קוד, נתפוס אותו בבדיקה הראשונה בהסתברות $1-\varepsilon$ וסיימנו. בה"כ ניתן להניח כי T_2 היא 99/100-קרובה לאיזו מילת קוד $c(v)$.

בא המבחן השני. המטרה שלנו היא לבדוק ש- $v=u\otimes u$. אם המוכיח בחר לתת לנו $v\neq u\otimes u$, נגלה את זה.

מה למעשה עושה הבדיקה השנייה? $(T_2)_{r_1\otimes r_2}\overset{?}{=}(T_1)_{r_1}\cdot(T_2)_{r_2}$. קוראים 3 ביטים: ביט אחד מ- T_2 ו-2 ביטים מ- T_1. כל מקום שאנחנו קוראים בפני עצמו מיקום אקראי.

יש סיכוי של לכל היותר 0.03 שאחד מהביטים שקראנו שונה מ- $c(u)$, $c(v)$ בהתאמה. חוץ מזה, מה שאנחנו רואים:

$$(T_2)_{r_1\otimes r_2}\overset{?}{=}(T_1)_{r_1}\cdot(T_1)_{r_2}$$
$$(c(v))_{r_1\otimes r_2}\overset{?}{=}(c(u))_{r_1}\cdot(c(u))_{r_2}$$
$$v\cdot(r_1\otimes r_2)\overset{?}{=}(u\cdot r_1)\cdot(u\cdot r_2)=(u\otimes u)\cdot(r_1\otimes r_2)$$

מכיוון ש- $v\neq u\otimes u$, לא נעבור את המבחן.
אנחנו משווים 2 מילות קוד שונות במיקום אקראי. מכיוון שכל 2 מילות קוד שונות בחצי מהמיקומים, בהסתברות 0.5 נתפוס את השגיאה.
עכשיו, בה"כ T_1 קרובה ל- $c(u)$, T_2 קרובה ל- $c(v)$ והקלט לא בשפה. לכן, בהכרח $A(u\otimes u)\neq\vec{b}$, ולכן יש לפחות $1\le i\le m$ אחד כך ש- $A_i\cdot(u\otimes u)\neq\vec{b}_i$.
בבדיקה השלישית, בוחרים $\alpha_1,\ldots,\alpha_m$ באקראי. נסמן: $C=\sum\alpha_i A_i$. בודקים ש- $(T_2)_B+(T_2)_{B\oplus C}\overset{?}{=}\sum\alpha_i\vec{b}_i$. קוראים 2 ביטים מ- T_2 .
בהסתברות לכל היותר 0.02 יש ביט שקראנו שלא משקף את $c(u\otimes u)$. חוץ מזה, אנחנו רואים $\underbrace{(c(u\otimes u))_B+(c(u\otimes u))_{B\oplus(\sum\alpha_i A_i)}}_{(c(u\otimes u))_{\sum\alpha_i A}=(u\otimes u)\cdot\sum\alpha_i A_i}\overset{?}{=}\sum\alpha_i\vec{b}_i$.

בודקים: $\sum\alpha_i(A\cdot(u\otimes u))=\sum\alpha_i\vec{v}_i$. כלומר: $\sum_{i=1}^{m}\alpha_i\cdot(A_i\cdot(u\otimes u)-b_i)\overset{?}{=}0$.

$\underbrace{\vec{\alpha}}_{\text{וקטור באורך } n}\cdot\underbrace{(A(u\otimes u)-\vec{b})}_{\text{וקטור באורך } n}\overset{?}{=}0$. $(c(A(u\otimes u)-\vec{b}))_{\vec{\alpha}}\overset{?}{=}0$, ולכן נתפוס את המוכיח מרמה בהסתברות 0.5 .
בסך הכל: $\Pr\left[\begin{matrix}\text{נטעה}\\ \text{ונגיד בסדר}\end{matrix}\right]\le\frac{2}{100}+\frac{1}{2}=0.6$.

משפט: $\mathrm{NP}=\mathrm{PCP}(O(\log n),O(1))$.

<u>הגדרה</u>: $Y,N\subseteq\{0,1\}^*$ הוא **בעיית הבטחה** (*promise problem*) אם $Y\cap N=\emptyset$.
<u>הגדרה</u>: $L\subseteq\{0,1\}^*$ הוא **שפה** אם בעיית ההבטחה המתאימה: $\langle L,\{0,1\}^*\setminus L\rangle$.
<u>הגדרה</u>: אלגוריתם A פותר בעיית הבטחה $\langle Y,N\rangle$ אם לכל $x\in Y$, $A(x)=1$ ולכל $x\in N$, $A(x)=0$.
מה קורה עם הקלטים $x\notin Y\cup N$? לא אכפת לנו.
<u>דוגמה</u>: $\mathrm{Gap}_{[\alpha,\beta]}\mathrm{Max3SAT}$, $\alpha<\beta$.
היא בעיית הבטחה שמקבלת פסוק 3SAT .
Y - כל פסוק 3SAT $\varphi=\bigwedge_{i=1}^{m}c_i$ כך שיש השמה שמספקת לפחות βm פסוקיות.
N - כל פסוק 3SAT $\varphi=\bigwedge_{i=1}^{m}c_i$ כך שאין השמה שמספקת αm פסוקיות.
<u>מטרה</u>: לטעון ישי $\beta<1$ כך ש- $\mathrm{Gap}_{[\beta,1]}\mathrm{Max3SAT}$ היא NP -קשה.
<u>הגדרה</u>: יהיו $\langle Y_1,N_1\rangle$, $\langle Y_2,N_2\rangle$ בעיות הבטחה. נגיד $\langle Y_2,N_2\rangle\le\langle Y_1,N_1\rangle$ אם יש רדוקציה $\varphi:\{0,1\}^*\to\{0,1\}^*$ שמקיימת $x\in Y_2\Leftrightarrow\varphi(x)\in Y_1$, $x\in N_2\Leftrightarrow\varphi(x)\in N_1$.
כמו במקרה הכללי, לרדוקציה יש סיבוכיות. ניתן להגדיר רדוקציות במחלקות, למשל $\le_{\mathrm{L}}$, $\le_{\mathrm{P}}$.
<u>הגדרה</u>: בעיית הבטחה $\langle Y,N\rangle$ היא NP -קשה אם לכל $A\in\mathrm{NP}$ מתקיים $\langle A,A^c\rangle\le_{\mathrm{P}}\langle Y,N\rangle$.
<u>משפט</u>: יש קבוע $\beta<1$ כך ש- $\mathrm{Gap}_{[\beta,1]}\mathrm{Max3SAT}$ היא NP -קשה. נניח שאנחנו יודעים $\mathrm{NP}=\mathrm{PCP}(O(\log n),O(1))$, ונוכיח את המשפט.
תהי $A\in\mathrm{NP}$. לכן, $A\in\mathrm{PCP}(O(\log n),O(1))$, ולכן יש לה פרוטוקול PCP כנ"ל.
לכל $x\in\{0,1\}^n$ קלט של A , נבנה פסוק 3SAT φ_x כך ש- $x\in A\Rightarrow\varphi_x$ ספיק (כלומר יש השמה המספקת את כל הפסוקיות), ו- $x\notin A\Rightarrow$ כל השמה תספק לכל היותר β מהפסוקיות.

הרדוקציה:

נתון $x\in\{0,1\}^n$.

בפרוטוקול ה- PCP , המוכיח נותן לנו הוכחה $w=w_1\ldots w_k$, $k=\text{Poly}(n)$.

הבודק מטיל $y\in\{0,1\}^{r(n)}$ מטבעות ($r(n)=O(\log n)$). לכל $y_1,\ldots,y_{r(n)}$ אפשרי, מסתכל על $q(n)=O(1)$ ביטים $w_{i_1},\ldots,w_{i_{q(n)}}$ ומחשב $V\left(x,y,w_{i_1},\ldots,w_{i_{q(n)}}\right)$ שמחליט אם לקבל או לדחות.

לכל $y_1,\ldots,y_{r(n)}$ אפשרי, נבנה פסוק 3SAT שהמשתנים שלו הם $w_{i_1},\ldots,w_{i_{q(n)}}$ והערך שלו $V\left(x,y,w_{i_1},\ldots,w_{i_{q(n)}}\right)$.

הפסוק $\varphi_{x,y_1,\ldots,y_{r(n)}}$ הוא פסוק 3SAT באורך קבוע $2^{O(q(n))}$.

הפלט של הרדוקציה: $\bigwedge_{y_1,\ldots,y_{r(n)}}\varphi_{x,\bar{y}}$.

הרדוקציה רצה ב- P , כי יש $2^{O(\log n)}=\text{Poly}(n)$ אפשרויות ל- y.

שלמות:

אם $x\in A$, יש מוכיח שנותן $w=w_1\cdots w_k$ כך שלכל $y\in\{0,1\}^n$ מתקיים $V\left(x,y,w_{i_1},\ldots,w_{i_{q(n)}}\right)=1$, ולכן עבור ה- w הזה, לכל y , $\varphi_{x,y}$ ספיק, ולכן w מספק את φ_x.

תקפות:

אם $x\notin A$, אז לכל $w=w_1\cdots w_k$ (נחשוב עליו כהוכחה שמוכיח רשע מספר למערכת ה- PCP), $\Pr_y\left[V\left(x,y,w_{i_1},\ldots,w_{i_{q(n)}}\right)=0\right]\geq 1-\beta$. לכן, $\Pr_y\left[\begin{matrix}v\text{ לא מספק}\\ \varphi_{x,y}\text{ את}\end{matrix}\right]\geq 1-\beta$.

ב- $\varphi_{x,y}$ יש מספר קבוע, B , של פסוקיות.

$\varphi_x=\bigwedge_{y_1,\ldots,y_{r(n)}}\underbrace{\varphi_{x,y}}_{\substack{\text{פסוק 3SAT}\\ \text{באורך קבוע}}}$, ולכן בסך הכל ב- φ_x יש $B\cdot 2^{r(n)}$ פסוקיות, ולכן ב- w שנקבל, לפחות $(1-\beta)\cdot 2^{r(n)}$ פסוקים $\varphi_{x,y}$ לא מסתפקים עם w , ולכן לפחות $(1-\beta)\cdot 2^{r(n)}$ פסוקיות של φ_x לא מסתפקות, ולכן אחז הפסוקיות שלא מסתפקות הוא לפחות $\gamma=\frac{1-\beta}{B}=\frac{(1-\beta)\cdot 2^{r(n)}}{B\cdot 2^{r(n)}}$, ו- $\gamma>0$ קבוע. זה נכון עם $\beta=\frac{1}{2}$.

מש"ל.

<u>הגדרה</u>: $\text{Gap}_{[\alpha,\beta]}\text{maxClique}$:

- Y - גרפים על n קודקודים בהם יש קליקה בגודל לפחות β.
- N - גרפים על n קודקודים בהם הקליקה המקסימלית היא בגודל לכל היותר α.

<u>טענה</u>: נניח כי $A\in\text{Gap}_{[a,b]}$ היא NP -קשה. אם יש אלגוריתם P שמקרב את A עד כדי פקטור $\frac{b}{a}$, אזי $\text{P}=\text{NP}$.