<u>תרגיל 1:</u>

נגדיר את מחלקת השפות DP באופן הבא: $L = L_1 \setminus L_2 \quad L \in DP \Leftrightarrow \exists L_1, L_2 \in NP$

(אוסף כל השפות שהן הפרש של שפות ב NP).

נגדיר את השפה $SATnoSAT$ באופן הבא: $SATnoSAT = \{(\varphi_1, \varphi_2) : \varphi_1 \in SAT \wedge \varphi_2 \notin SAT\}$

(אוסף כל הזוגות של פסוקי CNF, כך שהראשון ספיק והשני לא).

א. הוכח: $SATnoSAT \in DP$.

צ"ל: $SATnoSAT = L_1 \setminus L_2$ כך ש $L_1, L_2 \in NP$.

<u>הוכחה</u>: נבחר בשפות הבאות:

$$L_1 = \{(\varphi, x) \mid \varphi \in SAT, x \in \Sigma^*\}$$

$$L_2 = \{(x, \varphi) \mid \varphi \in SAT, x \in \Sigma^*\}$$

(נשים לב ששתי השפות הללו שייכות ל NP - פשוט מנחשים השמה ובודקים אם היא מספקת).

ב. הוכח: $SATnoSAT$ היא DP-קשה.

<u>הוכחה:</u>

תהי $L \in DP$. צ"ל: $L \leq_p SATnoSAT$. כלומר, $x \in L \Leftrightarrow f_P(x) = (\varphi_1, \varphi_2) \in SATnoSAT$

אם $L \in DP$ אז קיימות $L_1, L_2 \in NP$ כך ש $L = L_1 \setminus L_2$.

$SAT \in NPC$, $L_1, L_2 \in NP$ ולכן:

$L_1 \leq_P SAT$ באמצעות פונקצית הרדוקציה f_1

$L_2 \leq_P SAT$ באמצעות פונקצית הרדוקציה f_2

נגדיר: $f(x) = (f_1(x), f_2(x))$

$f(x)$ מלאה כי $f_1(x), f_2(x)$ מלאות.

$f(x)$ ניתנת לחישוב בזמן פולינומי כי $f_1(x), f_2(x)$ ניתנות לחישוב בזמן פולינומי.

תקפות:

אם $x \in L$ אז $x \in L_1$ וגם $x \notin L_2$ לכן (מתקפות f_1, f_2) $f_1(x) \in SAT$ ו $f_2(x) \notin SAT$.

לכן $f(x) = (f_1(x), f_2(x)) \in SATnoSAT$ כנדרש.

אם $x \notin L$ אז או ש $x \notin L_1$ או ש $x \in L_2$ לכן ((מתקפות f_1, f_2) $f_1(x) \notin SAT$ או $f_2(x) \in SAT$

לכן $(f_1(x), f_2(x)) \notin SATnoSAT$ כנדרש.

<u>תרגיל 2:</u>
עבור מ"ט א"ד M, נגדיר את הפונקציה $\#M:\Sigma^* \to \mathbb{N}$ באופן הבא:
מספר המסלולים של M המקבלים את x $\quad \#M(x) \triangleq$

נגדיר את מחלקת הפונקציות #P:
$\#P \triangleq \{f(x) \mid \#M(x) = f(x)$ כך ש: M קיימת מ"ט א"ד$\}$

בהנחה ש $P \neq NP$ הוכח או הפרך:
א. הפונקציה:
(אם x פסוק CNF, מספר ההשמות המספקות את x ואחרת 0) $= f(x)$ שייכת ל $\#P$.

<u>הוכחה</u>: המ"ט הרגילה (מנחשת השמה ומקבלת אמ"מ השמה זו מספקת את הפסוק)

ב. #P סגורה תחת חיבור, כלומר לכל $f_1, f_2 \notin \#P$ מתקיים $f_1 + f_2 \in \#P$
כאשר $(f_1 + f_2)(x) \triangleq f_1(x) + f_2(x)$.

<u>הוכחה</u>:
$f_1 \in \#P$ לכן קיימת מ"ט M_1 המחשבת אותה (כלומר M_1 על x - ישנם $f_1(x)$ מסלולים מקבלים)
$f_2 \in \#P$ לכן קיימת מ"ט M_2 המחשבת אותה.

נבנה M חדשה שמנחשת ביט.
אם בחרה ב 0, מריצה את M_1 על x ועונה כמוה.
אם בחרה ב 1, מריצה את M_2 על x ועונה כמוה.
לכן סה"כ מספר המסלולים שבו היא תקבל יהיה כמספר המסלולים ש M_1 מקבלת ועוד מספר המסלולים ש M_2 מקבלת.

ג. $\#P$ סגורה תחת כפל. כלומר לכל $f_1, f_2 \notin \#P$ מתקיים $f_1 \cdot f_2 \in \#P$
כאשר $(f_1 \cdot f_2)(x) \triangleq f_1(x) \cdot f_2(x)$.

<u>הוכחה</u>:
באופן דומה להוכחה לגבי חיבור:
$f_1 \in \#P$ לכן קיימת מ"ט M_1 המחשבת אותה (כלומר M_1 על x - ישנם $f_1(x)$ מסלולים מקבלים)
$f_2 \in \#P$ לכן קיימת מ"ט M_2 המחשבת אותה.
נבנה M חדשה שמריצה את M_1 על x ואם היא מקבלת אז מריצה את M_2 על x ועונה כמוה.

ד. אם $f:\Sigma^* \to \mathbb{N}$ ניתנת לחישוב בזמן פולינומי אז $f \in \#P$.

<u>הוכחה:</u> $f:\Sigma^* \to \mathbb{N}$ ניתנת לחישוב בזמן פולינומי, לכן קיימת מכונה M_f שמחשבת אותה תוך $P(x)$ צעדים.
נתאר מכונה M, א"ד מתאימה:
M על x:
1. באמצעות M_f תחשב את $f(x)$.
2. תנחש מספר k בן $p(|x|)$ ביטים, כלומר $0 \le k \le 2^{|x|}$
3. תקבל אם $k < f(x)$ ואחרת תדחה.

המכונה שתיארנו היא א"ד פולינומית כי כל הצעדים שלה פולינומים. התקפות ברורה.

ה. אם $f:\Sigma^* \to \mathbb{N}$ מקיימת $f \in \#P$ אז היא ניתנת לחישוב בזמן פולינומי.

<u>הפרכה:</u> הפונקציה מסעיף א' - אם זה היה נכון אז היא הייתה ניתנת לחישוב בזמן פולינומי ואז היה ניתן להכריע את SAT בזמן פולינומי בסתירה להנחה ש P שונה מ NP.

ו. כל פונקציה ב $\#P$ ניתנת לקירוב עד כדי קבוע חיבורי 10.
<u>הפרכה:</u> נראה כיצד להכריע את SAT בזמן פולינומי באמצעות אלגוריתם הקירוב:
בהינתן פסוק CNF, φ, נגדיר את הפסוק $\varphi' = \varphi \wedge (y_1 \vee y_2 \vee y_2 \vee y_4 \vee y_5)$ כאשר $y_1 ... y_5$ הם משתנים חדשים.
על כל השמה שמספקת את φ, יצרנו כעת $2^5 = 32$ השמות חדשות, שמלבד אחת (שנותנת ל $y_1 ... y_5$ את הערך F), כל ההשמות האחרות מספקות את φ'. לכן מספר ההשמות שמספקות את φ' הוא פי 31 ממספר ההשמות שמספקות את φ.

נניח בשלילה שקיים אלגוריתם קירוב כזה.
נבנה את φ' ונריץ עליו את אלגוריתם הקירוב ונשווה את התשובה ל 21:
אם התשובה גדולה או שווה ל 21 אז נקבל ואחרת נדחה.

<u>בעיית החלוקה PARTITION:</u> נתונה קבוצה $\left(X_1,...,X_n\right)$ ורוצים לדעת האם ניתן לחלק אותה לשני חלקים שסכום אבריהם שווים. ידוע שזו בעיה קשה (NPC)

בעיית L_b:

$$L_b=\left\{\left(X_1,...,X_n\right)\mid \exists I\subseteq\{1,...,n\}:\sum_{i\in I}x_i-\sum_{i\notin I}x_i=b\right\}$$

(האם קיימת חלוקה כך שההפרש בין שני החלקים הוא b)
נראה שגם זו בעיה קשה:

נראה רדוקציה מ PARTITION:

$$f\left(X_1,...,X_n\right)=\left(3bX_1,3bX_2,...,3bX_n,b\right)$$

אם $\left(X_1,...,X_n\right)$ היה ניתן לחלק לשני חלקים שווים, אז נשתמש באותה חלוקה כאשר את ה b הנותר נצרף לקבוצה הראשונה וכך ההפרש יהיה בדיוק b, כנדרש.

אם $\left(X_1,...,X_n\right)\notin Partition$ אז כל חלוקה מקיימת: $\sum_{i\in I}x_i-\sum_{i\notin I}x_i\geq 1$

לכן: $\sum_{i\in I}3bx_i-\sum_{i\notin I}3bx_i\geq 3b$ לכן לא משנה באיזה צד נשים את ה b נקבל שההפרש הוא לפחות $2b$,
ולכן $f\left(X_1,...,X_n\right)=\left(3bX_1,3bX_2,...,3bX_n,b\right)\notin L_b$