תוכן העניינים

1 מכונות טיורינג

הגדרה 1: מכונת טיורינג

מכונת טיורינג (מ"ט) היא שביעיה $M = (Q, \Sigma, \Gamma, \delta, q_0, \text{acc}, \text{rej})$.

Q	קבוצת מצבים סופיות	
Σ	א"ב קלט סופי	$\sqcup \notin \Sigma$
Γ	א"ב סרט סופי	$\Sigma \subseteq \Gamma$, $\sqcup \in \Gamma$
δ	פונקציית המעברים	$\delta : (Q \backslash \{\text{rej}, \text{acc}\} \times \Gamma \to Q \times \Gamma \times \{L, R\}$
q_0	מצב התחלתי	
acc	מצב מקבל	
rej	מצב דוחה	

הגדרה 2: קונפיגורציה

תהי $M = (Q, \Sigma, \Gamma, \delta, q_0, \text{acc}, \text{rej})$ מכונת טיורינג.
קונפיגורציה של M הינה מחרוזת

$$uq\sigma\text{v} \ , \quad u, \text{v} \in \Gamma^*, \ \sigma \in \Gamma, \ q \in Q \ .$$

משמעות:
- q מצב המכונה,
- σ הסימון במיקום הראש
- u תוכן הסרט משמאל לראש,
- v תוכן הסרט מימין לראש.

הגדרה 3: גרירה

תהי $M = (Q, \Sigma, \Gamma, \delta, q_0, \text{acc}, \text{rej})$ מכונת טיורינג, ותהיינה c_1 ו- c_2 קונפיגורציות של M.
נסמן

$$c_1 \vdash_M c_2$$

(במילים, c_1 גורר את c_2) אם כשנמצאים ב- c_1 עוברים ל- c_2 בצעד בודד.

נסמן

$$c_1 \vdash_M^* c_2$$

אם ניתן לעבור מ- c_1 ל- c_2 ב- 0 או יותר צעדים.

הגדרה 4: קבלה ודחייה של מחרוזת

תהי $M = (Q, \Sigma, \Gamma, \delta, q_0, \text{acc}, \text{rej})$ מכונת טיורינג, ו- $w \in \Sigma^*$ מחרוזת.
נאמר כי:

- M **מקבלת** את w אם $q_0 w \vdash_M^* u\ \text{acc}\ \sigma\ \text{v}$
- M **דוחה** את w אם $q_0 w \vdash_M^* u\ \text{rej}\ \sigma\ \text{v}$

כאשר $\text{v}, u \in \Gamma^*, \sigma \in \Gamma$ כלשהם.

הגדרה 5: הכרעה של שפה

תהי $M = (Q, \Sigma, \Gamma, \delta, q_0, \text{acc}, \text{rej})$ מכונת טיורינג, ו- $L \subseteq \Sigma^*$ שפה.
נאמר כי M **מכריעה** את L אם לכל $w \in \Sigma^*$ מתקיים

- $w \in L \Leftarrow$ M מקבלת את w.
- $w \notin L \Leftarrow$ M דוחה את w.

הגדרה 6: קבלה של שפה

תהי $M = (Q, \Sigma, \Gamma, \delta, q_0, \text{acc}, \text{rej})$ מכונת טיורינג, ו- $L \subseteq \Sigma^*$ שפה.
נאמר כי M **מקבלת** את L אם לכל $w \in \Sigma^*$ מתקיים

- אם $w \in L$ אז M מקבלת את w.
- אם $w \notin L$ אז M לא מקבלת את w.

במקרה כזה נכתוב ש- $L(M) = L$.

הגדרה 7: חישוב פונקציות

תהי $M = (Q, \Sigma, \Gamma, \delta, q_0, \text{acc}, \text{rej})$ מכונת טיורינג ותהי $f : \Sigma_1^* \to \Sigma_2^*$.
נאמר כי M **מחשבת** את f אם:

- $\Sigma = \Sigma_1$,$\Sigma_2 \subset \Gamma$,
- לכל $w \in \Sigma_1^*$ מתקיים $q_0 w \vdash_M^* \text{acc} f(w)$.

2 וריאציות של מכונות טיורינג

הגדרה 8: מודל חישוב

מודל חישובי = אוסף של מכונות שעבורם מוגדרים המושגים של הכרעה וקבלה של שפות.

הגדרה 9: מודלים שקולים חישובית

יהיו A, B מודלים חישוביים. נאמר כי A ו- B שקולים אם לכל שפה L:

- קיימת מכונה במודל A שמכריעה את L אם"ם קיימת מכונה כזו במודל B.
- קיימת מכונה במודל A שמקבלת את L אם"ם קיימת מכונה כזו במודל B.

הגדרה 10: מכונות שקולות חישובית

שתי מכונות הן שקולות חישובית אם הן מקבלות ודוחות בדיוק את אותן המילים.

משפט 1: מכונת טיורינג עם סרט ימינה בלבד

מודל מ"ט טם סרט אינסופי לכיוון אחד בלבד (מודל O) שקול למודל אינסופי בשני הכיוונים (מודל T). כלומר, לכל שפה L:

- יש מ"ט ממודל O שמקבלת את L אם"ם יש מ"ט במודל T שמקבלת את L.
- יש מ"ט ממודל O שמכריעה את L אם"ם יש מ"ט במודל T שמכריעה את L.

משפט 2: מכונת טיורינג מרובת סרטים

במכונת טיורינג מרובת סרטים:

- יתכנו מספר סטרים.

  מספר הסרטים סופי וקבוע מראש בזמן בניית המ"ט, ואינו תלוי בקלט או במהלך החישוב.
- לכל סרט יש ראש נפרד.
- הפעילות (תנועה וכתיבה) בכל סרט נעשית בנפרד.

  בפרט, הראשים יכולים לזוז בכיוונים שונים בסרטים שונים.
- ישנו בקר מרכזי יחיד, שקובע את הפעילות בכל אחד מהסרטים, על סמך המידע שמתקבל מכל הסרטים.

  לכן, תוכן סרט אחד יכול להשפיע על הפעילות בשאר הסרטים.

- בתחילת החישוב, הקלט נמצא בסרט הראשון ושאר הסרטים ריקים.

משפט 3:

לכל k, המודל של מ"ט עם k סטרים שקול חישובי למודל של מ"ט עם סרט אחד.

משפט 4:

קבלה ודחייה של מחרוזות:
עבור מ"ט לא דטרמיניסטית N ומחרוזת w:

- N **מקבלת** את w אם קיים חישוב של N על w שמגיע למצב מקבל.
- N **דוחה** את w אם כל החישובים של N על w עוצרים במצב דוחה.

הכרעה וקבלה של שפות:
עבור מ"ט לא דטרמיניסטית N ושפה L:

- N **מכריעה** את L אם N מקבלת אץ כל המילים ב- L ודוחה את כל המילים שאינן ב- L.
- N **מקבלת** את L אם N מקבלת אץ כל המילים ב- L ולא מקבלת את כל המילים שאינן ב- L.

משפט 5:

לכל מ"ט לא דטרמיניסטית קיימת מ"ט דטרמיניסטית שקולה.

3 התזה של צ'רץ'-טיורינג

4 התזה של צ'רץ'-טיורינג

שמות נרדפים לשפות כריעות ושפות קבילות

שפות כריעות	**Decideable languages**	**שפות קבילות**	**Acceptable languages**
שפות רקורסיביות	Recursive languages	שפות ניתנות לזיהוי	recognizable languages
		שפות כריעות למחצה	Semi-deidable languages
			Partially-deidable languages
		שפות הניתנות למנייה רקורסיביות	Recursively enumerable languages

משפט 6: סגירות שפות כריעות

השפות הכריעות סגורות תחת:

- איחוד

- חיתוך
- משלים
- שרשור

- סגור קלין

משפט 7: סגירות שפות קבילות

- איחוד
- חיתוך
- שרשור
- סגור קלין

משפט 8: היחס בין הכרעה לקבלה

אם שפה הינה כריעה אז היא קבילה.
אם שפה והמשלים שלה קבילות אז היא כריעה.

הגדרה 11: שפת סימפל

<u>משתנים</u>

- טבעיים:

```
i,j,k,...
```

  מקבלים כערך מספר טבעי.

- מערכים:

```
A[],B[],C[],...
```

  בכל תא ערך מתוך א"ב Γ אין סופיים.

- אתחול: הקלט נמצא בתאים הראשונים של

```
A[]
```

  .

  כל שאר המשתנים מאותחלים ל-

```
0
```

  .

<u>פעולות</u>

- השמה בקבוע:

```
i=3, B[i]="#"
```

- השמה בין משתנים:

```
i=k, A[k]=B[i]
```

- פעולות חשבון:

```
x = y + z , x = y - z , x = y.z
```

תנאים

- `B[i]==A[j]` (מערכים).
- `x >= y` (משתנים טבעיים).

כל משתנה מופיע רק פעם אחת בכל פעולה או תנאי.

זרימה

- סדרה פקודות ממוספרות.
- `goto` : מותנה ולא מותנה.
- `stop` עצירה עם ערך חזרה.

```
one = 1
zero = 0
B[zero] = "0"
i=0
j=i
if A[i] == B[zero] goto 9
i=j + one
goto 3
C[one] = A[j]
if C[one] ==  A[zero] goto 12
stop(0)
stop(1)
```

הגדרה 12: קבלה ודחייה של מחרוזת בשפה SIMPLE

עבור קלט w ותוכנית P בשפה SIMPLE. נאמר כי

- P **מקבלת** את w אם הריצה של P

על

w

עוצרת עם ערך חזרה

1

.

- P

דוחה את

w

אם הריצה של

P

על

w

עוצרת עם ערך חזרה

0

.

הגדרה 13: הכרעה וקבלה של שפות

עבור שפה

L

ותוכנית

P

בשפת SIMPLE. נאמר כי

- P

מכריעה את

L

אם היא מקבלת את המילים שב-

L

ודוחה את אלה שלא ב-

L

.

- P

מקבלת את

L

אם היא מקבלת את כל ורק המילים ב-

L

.

משפט 9:

המודלים של מכונת טיורינג ותוכניות SIMPLE שקולים.

משפט 10: מ"ט ותוכניות מחשב

מ"ט חזקה לפחות כמו תוכנית משחב.
כל תוכנית מחשב ניתנת למימוש במ"ט.
לכן, כל שפה שהינה כריעה ע"י מחשב היא כם כריעה ע"י מ"ט.
וכמו כן, שפה שהינה קבילה ע"י מחשב היא גם קבילה ע"י מ"ט.

הגדרה 14: דקדוקים כלליים

בדקדוק כללי, בצד שמאל של כלל יצירה יכולה להופיעה מחרוזת (לא ריקה) כלשהי.
פורמלית, כלל יצירה בדקדוק כללי הוא מהצורה

$$\gamma \rightarrow u$$

כאשר $\gamma \in (V \cup \Sigma)^+$, $u \in (V \cup \Sigma)^*$.

משפט 11:

תהי L שפה. L קבילה אם"ם קיים דקדוק כללי G כך ש- $L(G) = L$.

משפחת שפות	דקדוק	מודל חישובי
קבילות	כללי	מכונת טיורינג
חסרות הקשר	חסר הקשר	אוטומט מחסנית
רגולריות	רגולרי	אוטומט סופי

משפט 12:

כל שפה חסרת הקשר הינה כריעה.

משפט 13: התזה של צ'רץ' טיורינג

התזה של צ'רץ' טיורינג מודל מ"ט מגלם את המושג האבסטרקרטי של "אלגוריתם".
כלומר, כל אלגוריתם שניתן לתיאור כתהליך מכניסטי שבו:

- התהליך מתבצע כסדרה של צעדים.
- כל צעד מצריך כמות סופית של "עבודה".

ניתן גם לתיאור כמ"ט.
בפרט, אין מודל מכניסטי / אוטומטי יותר ממ"ט.

5 אי-כריעות

הגדרה 15: השפה ATM

$$ATM = \left\{(P,w) \;\middle|\; P(w)=1\right\} \ .$$

השפה ATM כוללת את כל הזוגות של מחרוזות P,w כך ש:

- P היא קוד (תקין) של תוכנית.
- w מחרוזת.
- מתקיים שאם מריצים את התוכנית P על הקלט w אז התוכנית עוצרת עם ערך חזרה 1.

הגדרה חלופית:

$$A_{TM} = \left\{\langle M,w\rangle \;\middle|\; \text{מכונת טיורינג שמקבלת את } w \;\; M\right\}$$

השפה A_{TM} כוללת את כל הזוגות של מחרוזות $\langle M,w\rangle$ של כל מכונת טיורינג M וכל קלט w כך ש- M מקבלת את w.

סיכום 1: התוכנה U

התוכנה U היא תוכנה שמקבלת כקלט זוג מחרוזות P,w ופועלת כך:

- U מחזירה את ערך החזרה שהתקבל מהריצה של P על w.
- מריצה את התוכנה P על קלט w (במקרה שבו P אינה תוכנית מחשב תקינה אז U מחזירה ערך 0).

נשים לב שאם P לא עוצרת על w אז גם U לא עוצרת על הזוג P,w.
התוכנה U פועלת באופן דומה לאופן שבה מערכת ההפעלה מפעילה תוכנות אחרות.

התוכנה U נקראת גם "תוכנה אוניברסלית" (או, בעולם מכונות טיורינג: "מכונת טיורינג אוניברסלית") כיוון שהיא תוכנה אחת שמדמה כל תוכנה אחרת.

U היא תוכנית שמקבלת את ATM. כלומר:

$$L(U) = ATM \ .$$

מסקנה: ATM קבילה.

שיטת ההוכחה:
אם שפה הינה קבילה אבל לא כריעה, אז המשלים שלה בהכרח אינה קבילה.
לכן, בשביל להוכיח ששפה אינה קבילה, די להוכיח שהשפה לא כריעה, והמשלים שלה כן קבילה.

הגדרה 16: השפה HALT

$$HALT = \left\{(P,w) \mid P(w)\downarrow\right\} \ .$$

השפה HALT כוללת את כל הזוגות של מחרוזות P, w כך ש:

- P היא קוד (תקין) של תוכנית.
- w מחרוזת.
- מתקיים שאם מריצים את התוכנית P על הקלט w אז התוכנית עוצרת (הסימון $\downarrow$ מסמן עצירה).)

בעיית העצירה קבילה אבל לא כריעה.
כיוון שכך, המשלים שלה לא כריעה ולא קבילה.

הגדרה חלופית:

$$HALT_{TM} = \left\{\langle M,w\rangle \mid \text{מכונת טיורינג שעוצרת על } w \ M\right\}$$

השפה $HALT_{TM}$ כוללת את כל הזוגות של מחרוזות $\langle M,w\rangle$ של כל מכונת טיורינג M וכל קלט w כך ש- M עוצרת על w.

הגדרה 17: השפה E

$$E = \left\{P \mid L(P) = \emptyset\right\}$$

השפה E כוללת את כל המחרוזות P כך ש-

- P היא קוד (תקין) של תוכנית.
- השפה של P ריקה.

כלומר, לכל קלט w, הריצה של P על w לא מחזיריה 1.

הגדרה חלופית:

$$E_{TM} = \left\{\langle M\rangle \mid \text{מכונת טיורינג שעומדת בתנאי } L(M)=\emptyset \ M\right\}$$

השפה E_{TM} כוללת את כל מחרוזות $\langle M\rangle$ של כל מכונת טיורינג M כך ש- M לא מקבלת אף מילה.
במילים אחרות, השפה של M ריקה: $L(M) = \emptyset$.

הגדרה 18: השפה EQ

$$EQ = \left\{(P_1,P_2) \mid L(P_1) = L(P_2)\right\} \ .$$

השפה EQ כוללת את כל זוגות המחרוזות P_1, P_2 כך ש:

- P_1, P_2 הינן קודים (תרינים) של תוכניות.
- השפות של P_1, P_2 זהות.

כלומר, P_1, P_2 מקבלות בדיוק את אותן המילים.

הגדרה חלופית:

$$EQ_{TM} = \left\{ \langle M_1, M_2 \rangle \;\middle|\; \text{מכונות טיורינג } M_1, M_2 \quad L(M_1) = L(M_2) \right\}$$

השפה EQ_{TM} כוללת את כל זוגות של מכונות טיורינג $\langle M_1, M_2 \rangle$ שמקבלות בדיוק אותן המילים. במילים אחרות, השפות של M_1 ו- M_2 זהות: $L(M_1) = L(M_2)$.

	כריעה	קבילה
ATM	$\times$	$\checkmark$
$\overline{ATM}$	$\times$	$\times$
$HALT$	$\times$	$\checkmark$
$\overline{HALT}$	$\times$	$\times$
E	$\times$	$\times$
$\overline{E}$	$\times$	$\checkmark$
EQ	$\times$	$\times$
$\overline{EQ}$	$\times$	$\times$

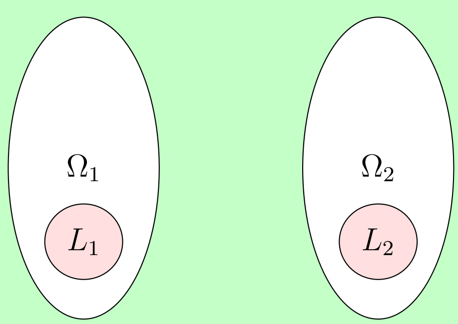
הגדרה 19: הרדוקציה

רדוקציית התאמה (many to one reduction) מקבוצה $L_1 \subseteq \Omega_1$ לקבוצה $L_2 \subseteq \Omega_2$ הינה **פונקציה**

$$R : \Omega_1 \to \Omega_2$$

כך שלכל $x \in \Omega_1$ מתקיים:

$$x \in L_1 \quad \Leftrightarrow \quad R(x) \in L_2 \ .$$



סימון: $L_1 \leq_m L_2$ רדוקציה התאמה ניתנת לחישוב מ- L_1 ל- L_2.

משפט 14: משפט הרדוקציה

טענה:
אם:

- L_2 כריעה
- $L_1 \leq L_2$

אז L_1 כריעה.

מסקנה:
אם:

- L_1 לא כריעה
- $L_1 \leq L_2$

אז L_2 לא כריעה.

מתכון להוכחה ששפה L_2 לא כריעה:

1. בחר שפה L_1 לא כריעה.
2. מצא רדוקציית התאמה ניתנת לחישוב מ- L_1 ל- L_2.

טענה:
אם:

- L_2 קבילה
- $L_1 \leq L_2$

אז L_1 קבילה.

מסקנה:
אם:

- L_1 לא קבילה
- $L_1 \leq L_2$

אז L_2 לא קבילה.

מתכון להוכחה ששפה L_2 לא קבילה:

1. בחר שפה L_1 לא קבילה.
2. מצא רדוקציית התאמה ניתנת לחישוב מ- L_1 ל- L_2.

משפט 15: תכונות של רדוקציות

A	$\leq_m$	B
כריעה	$\Leftarrow$	כריעה
לא כריעה	$\Rightarrow$	לא כריעה

A	$\leq_m$	B
קבילה	$\Leftarrow$	קבילה
לא קבילה	$\Rightarrow$	לא קבילה

משפט 16: לכל שפה קיימת רדוקציה ל- A_{TM}

מכל שפה כריעה A קיימת רדוקציה חישובית ל- A_{TM}.

כלומר

$$A \leq_m A_{TM} \ .$$

משפט 17: רדוקציה משפות כריעות

מכל שפה כריעה קיימת רדוקציה חשיבה לכל שפה אחרת שאינה $\emptyset$ או Σ^*.

הגדרה 20:

$$NOTREG = \left\{ P \;\middle|\; \text{לא רגולרית } L(P) \right\} \ .$$

השפה NOT-REG כוללת את כל המחרוזות P כך ש:

- P הינה קוד (תקין) של תוכנית.
- השפה של P לא רגולרית.

הגדרה חלופית:

$$NOTREG_{TM} = \left\{ \langle M \rangle \;\middle|\; \text{לא רגולרית } L(M) \right\} \ .$$

השפה $NOTREG_{TM}$ כוללת את כל המחרוזות $\langle M \rangle$ של מ"ט M כך שהפשה של M לא רגולרית.

משפט 18: השפה $NOT-REG$ אינה קבילה.

השפה $NOT-REG$ אינה קבילה.

6 סיבוכיות זמן

הגדרה 21: זמן הריצה

זמן הריצה של מכונת טיורינג M על קלט w הוא מספר צעדי החישוב ש- M מבצעת על w.

הגדרה 22: סיבוכיות זמן ריצה

תהי M מ"ט דטרמיניסטית אשר עוצרת על כל קלט. **הזמן הריצה** או **הסיבוכיות זמן** של M היא פונקציה $f : \mathbb{N} \to \mathbb{N}$, כאשר $f(n)$ המספר צעדי חישוב המקסימלי ש- M מבצעת על קלט w של אורך n.

אם $f(n)$ זמן הריצה של M, אומרים כי M רץ בזמן $f(n)$ וש- M היא $f(n)$ זמן מכונת טיורינג

הגדרה 23: מחלקה של סיבוכיות זמן

המחלקת הסיבוכיות זמן מסומנת $\text{TIME}\left(t(n)\right)$ ומוגדרת להיות אוסף של כל השפות אשר ניתנות להכרעה על ידי מכונת טיורינג בזמן $O\left(t(n)\right)$.

משפט 19:

הגדרת זמן הריצה שנתנו היא תלויה במודל של מכונת הטיורינג שאיתו אנחנו עובדים.

משפט 20:

תהי $t : \mathbb{N} \to \mathbb{R}^+$ פונקציה $t(n)$.
אם מתקיים

$$t(n) \geq n$$

אז לכל מכונת טיורינג $O\left(t(n)\right)$ רב-סרטי קיימת מ"ט $O\left(t^2(n)\right)$ עם סרט אחד.

הגדרה 24: זמן הריצה של מ"ט לא דטרמיניסטית

תהי N מכונת טיורינג לא דטרמיניסטית.
הזמן הריצה של N מוגדרת להיות הפונקציה $f : \mathbb{N} \to \mathbb{N}$ כאשר $f(n)$ הוא המספר הצעדים המקסימלי אשר N מתוך כל הענפים של החישוב שלה על קלט של אורך n.

משפט 21:

תהי $t(n)$ פונקציה המקיימת $t(n) \geq n$. כל מ"ט $O\left(t(n)\right)$ לא דטרמיניסטית N סרט אחד, שקולה למכונת טיורינג $2^{O(t(n))}$ דטרמיניסטית סרט אחד.

הגדרה 25: מכונת טיורינג פולינומית

מכונת טיורינג M תיקרא **פולינומית** או **יעילה** אם קיים $c \in \mathbb{N}$ כך ש- M פועלת בסיבוכיות זמן ריצה $O\left(n^c\right)$.

הגדרה 26: המחלקה P

המחלקה P היא אוסף השפות שקיימת מכונת טיורינג פולינומיאלית M המכריעה אותן. כלומר:

$$P = \bigcup_k \text{TIME}\left(n^k\right) \ .$$

הגדרה 27: אלגוריתם אימות

אלגוריתם אימות של שפה A הוא אלגוריתם V כך ש-:

$$A = \left\{w \mid \ V \text{ מקבל } \langle w, c\rangle \text{ על פי } c\right\}$$

במילים, **אלגוריתם אימות** הוא אלגוריתם V אשר מאמת כי הקלט w שייך לשפה A על פי התנאי, c שנקרא **אישור** (certification).

אנחנו מגדירים את זמן הריצה של V על פי האורך של w. לכן **אלגוריתם אימות זמן-פולינומיאלי** רץ בזמן פולינומיאלי $O\left(n^k\right)$ כאשר n האורך של w.

הגדרה 28: מחלקת הסיבוכיות NP

- המחלקה NP היא מחלקת השפות שניתן לאמתן באמצעות אלגוריתם זמן-פולינומיאלי.

הגדרה חלופית למחלקה NP הינה:

- המחלקה NP היא מחלקת השפות שניתן להכרעה באמצעות מ"ט אי-דטרמיניסטית זמן-פולינומיאלית.

למטה במשפט 22.

משפט 22: $A \in NP$ אם"ם A ניתנת לאימות ע"י N_{TM}

שפה A כלשהי שייכת למחלקה NP אם ורק אם A ניתנת להכרעה על ידי מכונות טיורינג אי-דטרמיניסטית זמן-פולינומיאלית.

הגדרה 29: פונקיצה הניתנת לחישוב זמן-פולינומיאלי

פונקציה $f:\Sigma^* \to \Sigma^*$ ניתנת לחישוב זמן-פולינומיאלי אם קיימת מ"ט זמן-פולינומיאלית M, עבורה על הקלט w, M עוצרת עם $f(w)$ על הסרט שלה.

הגדרה 30: פונקציה שניתנת לרדוקציה זמן-פולינומיאלית

השפה A ניתנת לרדוקציה זמן-פולינומיאלית לשפה B, שנסמן $A \leq_P B$, אם קיימת פונקציה שנתנת לחישוב זמן-פולינומיאלית $f:\Sigma^* \to \Sigma^*$ כך שלכל

$$w \in A \Leftrightarrow f(w) \in B\ .$$

הפונקציה f נקראת **הרדוקציה זמן-פולינומיאלית** של A ל- B.

משפט 23: אם $A \leq_P B$ ו- $B \in P$ אז $A \in P$

אם $A \leq_P B$ ו- $B \in P$ אז $A \in P$.

משפט 24: 3-SAT ניתנת לרדוקציה זמן-פולינומיאלית ל- CLIQUE

בהביית 3-SAT ניתנת לרדוקציה זמן-פולינומיאלית לבעיית CLIQUE:

$$3SAT \leq_p CLIQUE\ .$$

מסקנה 1: $3SAT \in P \quad \Rightarrow \quad CLIQUE \in P$

לפי משפט 23 ומשפט 24:

אם $CLIQUE \in P$ אז $3SAT \in P$.

הגדרה 31: NP-שלמות

שפה B היא **NP- שלמה** או **שלמה ב- NP** (NP-complete) אם היא מקיימת את השני התנאים הבאים:

1) $B \in NP$ וגם

2) $A \leq_p B$ עבור כל $A \in NP$.

במילים פשוטות: כל A ב- NP ניתנת לרדוקציה זמן-פולינומיאלית ל- B.

הגדרה 32: NP קשה

אם שפה B מקיימת את תכונה 2) אולם לא בהכרח את תכונה 1) בהגדרה 31 אז אומרים כי B **NP- קשה** או **קשה ב- NP** (NP-hard).

משפט 25:

אם B NP- שלמה ו- $B \in P$ אז $P = NP$.

משפט 26: אסוציאטיביות של NP שלמות

אם השני תנאים הבאים מתקיימים:

1) B היא שפה NP- שלמה.

2) קיימת $C \in NP$ עבורה $B \leq_p C$.

אז C שפה NP- שלמה.

משפט 27: משפט קוק לוין

הבעיית SAT היא NP - שלמה.

משפט 28: 3-SAT היא NP שלמה.

3-SAT היא NP שלמה.

7 נוסחאות נוספות

הגדרה 33: הבעיית הספיקות SAT

$$SAT = \{\langle\phi\rangle \mid \text{היא נוסחה בוליאנית ספיקה } \phi\}$$

במילים, בעיית SAT שואלת את השאלה: בהינתן נוסחה שמכילה רק קשרים $\wedge$, $\vee$ ו- $\neg$ ("גם", "או" ו"לא"), האם קיימת השמה של ערכי אמת למשתנים כך שהנוסחה ϕ תקבל ערך אמת?
אם קיימת השמה כזו אז נאמר כי הנוסחה ϕ ספיקה.

הגדרה 34: הבעיית 3-SAT

$$3SAT = \{\langle\phi\rangle \mid \text{היא נוסחת בוליאנית בצורה 3CNF ספיקה. } \phi\}$$

במילים, $3SAT$ היא הבעיית SAT שמוגדר בהגדרה 33 במקרה הנוסחה שהנוסחה היא בצורה 3CNF.
דוגמה של נוסחה בצורה 3CNF היא:

$$(x_1 \vee \bar{x}_1 \vee \bar{x}_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$$

הגדרה 35: הביית PATH

בהינתן גרף מכוון $G=(V,E)$.
הבעיית PATH שואלת את השאלה הבאה: בהינתן גרף מכוון $G=(V,E)$, וקדקודים s ו- t.
האם הגרף G מסלול בין קדקוד s לבין קדקוד t.

$$PATH = \{\langle G,s,t\rangle \mid \text{גרף מכוון שמכיל מסלול מכוון מ- } s \text{ ל- } t \ \ G\} \ .$$

הגדרה 36: מסלול המילטוני

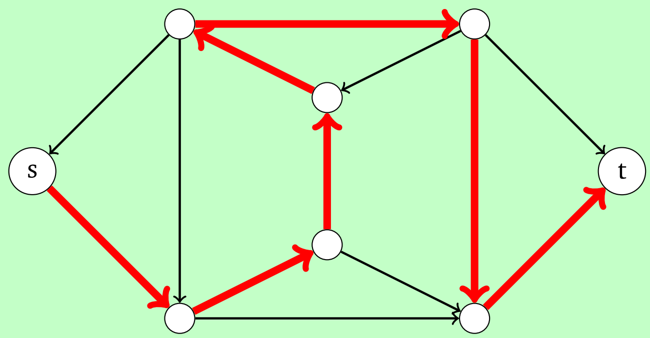
נתון גרף מכוון $G=(V,E)$.
מסלול המילטוני מקדקוד s לקדקוד t הוא מסלול אשר עובר דרך כל קדקוד בדיוק פעם אחת.

הגדרה 37: הבעיית מסלול המילטוני HAMPATH

בהינתן גרף מכוון $G=(V,E)$ וקדקודים s ו- t.
הבעיית המסלול ההמילטוני שואלת את השאלה: האם קיים מסלול המילטוני מקדקוד s לקדקוד t?

$$HAMPATH = \{\langle G,s,t\rangle \mid \text{גרף מכוון המכיל מסלול המילטוני מ- } s \text{ ל- } t. \ \ G\}$$

התרשים למטה מראה דוגמה של מסלול המילטוני בגרף מכוון.



הגדרה 38:

בהינתן שלמים x, y.
הבעייה RELPRIME שואלת את השאלה: האם x, y זרים.

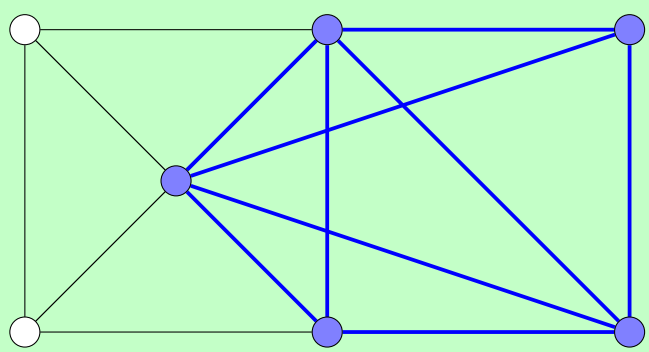
$$RELPRIME = \Big\{ \{x, y\} \in \mathbb{N} \;\Big|\; \gcd(x, y) = 1 \Big\}$$

הגדרה 39: קליקה

נתון גרף לא מכוון.

- קליקה בגרף לא מכוון הוא תת-גרף שבו כל זוג קדקודים קשורים על ידי קשת.
- k- קליקה היא קליקה שבו יש בדיוק k קדקודים.

התרשים למטה מראה דוגמה של 5- קליקה.



הגדרה 40: בעיית הקליקה

נתון גרף לא מכוון $G = (V, E)$. בעיית הליקה שואלת את השאלה:
האם הגרף G מכיל קליקה בגודל k.
בשפה פרומלית:

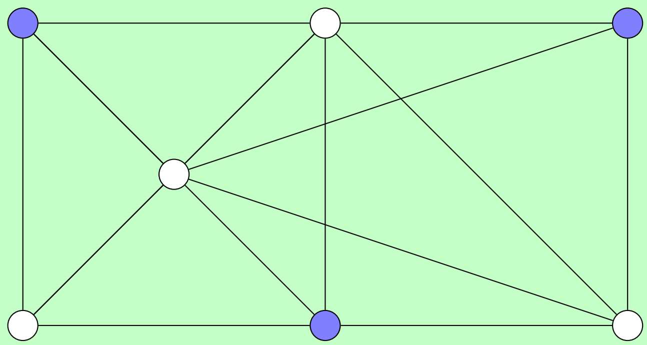
$$CLQ = \Big\{ \langle G, k \rangle \;\Big|\; \text{גרף לא מכוון שמכיל קליקה בגודל } k \text{ לפחות} \; G \Big\}$$

הגדרה 41: קבוצה בלתי תלויה

נתון גרף לא מכוון $G=(V,E)$.
קבוצה בלתי תלויה ב- G היא תת-קבוצה של קדקודים $S\subseteq V$ כך שלכל שני קדקודים $u_1,u_2\in S$ מתקיים ש-

$$(u_1,u_2)\notin E\ .$$

התרשים למטה מראה דוגמה של גרף לא מכוון G שמיל קבוצה בלתי תלויה בגודל 3.



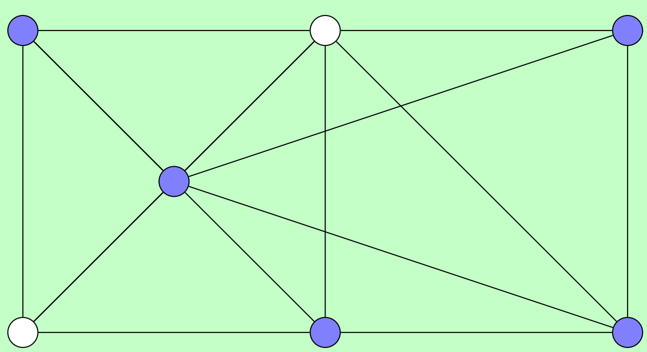
הגדרה 42: בעיית בקבוצה הבלתי תלוייה (IS) Independent Set

בהינתן גרף לא מכוון $G=(V,E)$ ומספר טבעי k.
הבעייה IS שואלת את השאלה: האם קיימת קבוצה בלתי תלוייה ב- G בגודל k לפחות.
בשפה פורמלית:

$$IS=\left\{\langle G,k\rangle\ \middle|\ \text{מכיל קבוצה בלתי תלוייה בגודל } k \text{ לפחות.}\ G\right\}$$

הגדרה 43: כיסוי קדקודים

בהינתן גרף לא מכוון $G=(V,E)$.
כיסוי קדקודים ב- G הוא תת-קבוצה של קדקודים $C\subseteq V$ כך שלכל צלע $(u_1,u_2)\in E$ מתקיים:
$u_1\in C$ או $u_2\in C$.
הגרף למטה מכיל כיסוי קדקודים בגודל 5.



הגדרה 44: הבעיית כיסוי קדקודים (VC) Vertex Cover

בהינתן גרף לא מכוון $G=(V,E)$ ומספר טבעי k.
הבעיית כיסוי קדקודים שואלת את השאלה הבאה:
האם קיים כיסוי בקדקודים ב- G בגודל k?
בשפה פורמלית:

$$VC = \left\{ \langle G,k \rangle \;\middle|\; G \text{ מכיל כיסוי בקדקודים בגודל } k \right\} .$$

משפט 29: שפות NP- שלמות

SAT	NP- שלמה.	(משפט קוק לוין)
3SAT	NP- שלמה.	
HAMPATH	NP- שלמה.	
CLIQUE	NP- שלמה.	
INDEPENDENT-SET	NP- שלמה.	
VERTEX-COVER	NP- שלמה.	