חישוביות - שיעור 1

דוד קלוטה

22 במרץ 2026

הקדמה

אנחנו נמצאים במצב חריג ביותר (עם כמה שמצבים חריגים היו השגרה לאחרונה). כל אחד מאיתנו נמצא בסיטואציה שונה הרגע, גיא גר בתל אביב בלי ממ"ד (ועל כך יידרש לעזוב את הקורס במקרה של אזעקה), בעוד אני מסכם (הרגע) מנתניה, עם ממ"ד. גיא שלח קבוצת ווטסאפ ייעודית בה גיא יפרסם מתי השיעור יתחיל (כתוצאה מהמצב הביטחוני המוזכר דלעיל). להלן קבוצת הווטסאפ. [הערה של דוד: שניות לאחר הפצת הקישור, התקבלה התרעה על שיגור טילים מאיראן באזור המרכז. השיעור המשיך 20 דקות לאחר מכן בעת ירי הטילים לאזור המרכז.]

נהלי הקורס יפורסמו בזמן הקרוב במודל הקורס. האוניברסיטה מכירה בכך שתלמידים רבים במילואים - אך עקב חוסר הוודאות לא ניתן לפרסם הרגע מתווה סופי. יש לציין שב"מילואים" הכוונה היא לכל אדם אשר מושפע משירות מילואים (לרבות בני.ות זוג של משרתי מילואים). בעת ביצוע השיעורים בזום, יש לפתוח מצלמה בעת השיעורים. על אף שסיכומים אלה יפורסמו (בנוסף לסיכומים משנים קודמות) מומלץ בחום לסכם את הקורס - עם דגש על שימוש בכלי כתיבה.

לפני שנתחיל לדבר על החומר, יש לשים לב כי זה **קורס מתמטי** (עם טענות, הוכחות וכד') אך זוהי לא מתמטיקה כמו שאנחנו מכירים מקורסי מתמטיקה קודמים. פה יש מספר רב של הוכחות "משעממות" העוסקות במספר רב של מקרי קצה - יש לשים לב כי במהלך הקורס נקבל כתרגיל השלמה של הוכחות (אינדוקציות ומקרי קצה). בשיעורים הראשונים נציג מספר רב של מושגים חדשים - כל מושג מפני עצמו פשוט יחסית, אבל זה מצטבר. כתוצאה מכך, חשוב ביותר לעקוב, לזכור ולעכל את ההגדרות. כאשר מגדירים אובייקט זה מרגיש קל, אבל אז מגדירים קבוצה של האובייקט, ואף קבוצה של קבוצות של האובייקט! לפני כל שיעור, מומלץ בחום על ידי גיא להסתכל על הסיכומים של השיעור הקודם - זה עוזר לעקוב אחרי השיעור הנוכחי!

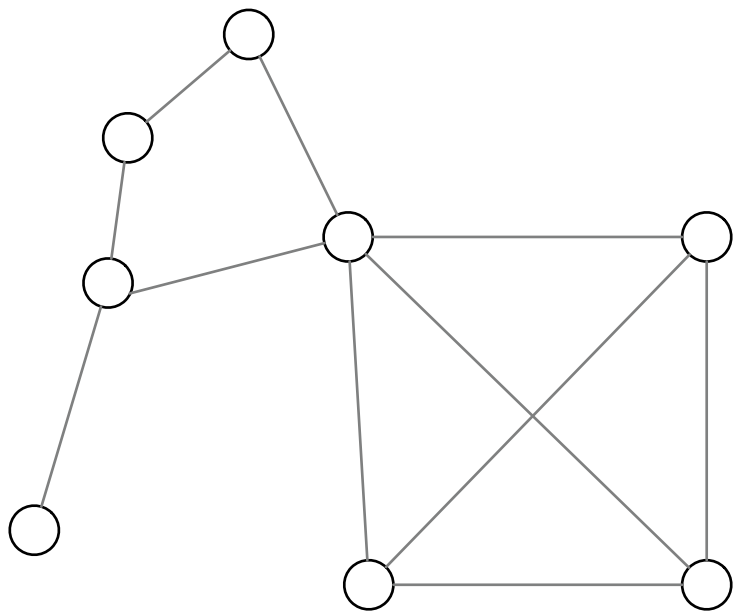
נושאי הקורס

הנושא הראשון של הקורס יהיה **בעיות חישוביות**. ראינו כבר כאלה - אבל לא הגדרנו אותן.

דוגמה **1.0.1.** 1. בהינתן מערך של n מספרים, יש למיין אותו. ראינו בקורס DAST כי אפשר לבצע את המשימה הזו בסיבוכיות זמן ריצה של $\mathcal{O}(n \log n)$.

2. בהינתן גרף $G = (V, E)$ ושני צמתים $s, t \in V$ ב־G. יש למצוא מסלול קצר ביותר בין s ל־t כאשר $|V| = n$. ניתן לפתור בעיה זו בזמן ריצה של $\mathcal{O}(|E| + |V|) = \mathcal{O}(n^2)$

3. הנה דוגמה (חדשה) למשימה חישובית: אם בהינתן גרף $G = (V, E)$ עם n צמתים יש למצוא את גודל הקליקה המקסימלית בו.



איור 1: גרף המכיל קליקה מקסימלית בת ארבע צמתים

יש לנו מספר אפשרויות לנסות למצוא את גודל הקליקה המקסימלי.

(א) ננסה את כל הקבוצות של הצמתים - ייקח לנו $\mathcal{O}(n^2 \cdot 2^n)$

(ב) לעבור על כל צומת ולחשב את הקליקה המקסימלית שמכילה אותו. בגישה זו אנחנו עוברים על כל אחד מ־n הצמתים ועושים עליהם משהו. נקבל נוסחת רקורסיה כזו:

$$T(n) = nT(n-1) = \mathcal{O}(n!)$$

נבחין כי שני האלגוריתמים שהבאנו לא יעילים בעליל, אך לא ידוע לבעיה זו זמן ריצה טוב יותר.

4. בהינתן תכנית בשפת Python (מחרוזת של תווים), יש להגיד אם היא תעתור כשנריץ אותה - זוהי **בעיית העצירה** Halting problem ולא ניתן לפתור אותה! (ספציפית ידוע כי לא קיים אלגוריתם אשר פותר את הבעיה).

סך הכל ראינו בעיות אשר קיים עבורן פתרון יעיל (כמו מיון מערך), בעיות אשר לא ידוע לנו (אך יש מצב כלשהו שקיים) פתרון יעיל אליהן (כמו מציאת גודל קליקה מקסימלית), ובעיות שאין להן פתרון כלל (כמו בעיית העצירה). על מנת להבין את האמירה "לא קיים מחשב אשר יכול לפתור את בעיית העצירה", נידרש קודם להגדיר בעיה חישובית, להגדיר אלגוריתם ולהגדיר מחשב, חישוב ומודל חישובי.

הצלחנו להגיע לפתרון של $\mathcal{O}(n \log n)$ בהינתן מחשב אשר קיימות בו השוואות. האם ניתן להגיע לפתרון יעיל יותר בהינתן מודל חישובי אחר? זוהי שאלה מורכבת ביותר, אשר תדרוש מאיתנו להיכנס למקומות אשר לא העזנו לגעת בהם עד קורס זה.

כאשר כל הדברים האלה מוגדרים, נוכל לדון במחלקות של בעיות חישוביות:

1. בעיות אשר ניתן לפתור בזמן פולינומי
2. בעיות אשר ניתן לפתור אך לא בזמן פולינומי
3. בעיות אשר לא ניתן לפתור כלל

דבר נוסף שנדבר עליו בקורס הוא **סיבוכיות**. אחרי שכבר בחרנו מודל חישובי, נוכל לשאול בהינתן בעיה חישובית, כמה משאבים צריך על מנת לפתור אותה (סיבוכיות זמן, מקום, אקראיות, ומגוון סוגי משאבים אחרים)?

לבסוף נעסוק בתחום **החישוביות**: בהינתן בעיה חישובית, האם בכלל ניתן לפתור את הבעיה במודל חישובי כלשהו?

התחלנו!

1.1 בעיות חישוביות

הגדרה 1.1.1 (מילה word או מחרוזת string). מילה או מחרוזת מעל א"ב Σ היא מחרוזת של תווים מתוך Σ (שעבורנו תמיד תהיה קבוצה סופית ולא ריקה). להלן מספר דוגמאות:

$$\Sigma_1 = \{0, 1, \ldots, 9\}, \quad \Sigma_2 = \{a, b, \ldots, z\}$$

המילה היא $\{\sigma_i\}_{i=0}^{n}$ היא סדורה בה לכל $i \in [n] \cup \{0\}$ מתקיים $\sigma_i \in \Sigma$.

הגדרה 1.1.2 (המילה הריקה). המילה ε הינה מילה בעלת אורך של 0 תווים.

הגדרה 1.1.3 (בעיה חישובית computational task / problem). בעיה חישובית עם קלטים מעל א"ב Σ ופלטים מעל א"ב Γ היא פונקציה המתאימה לכל מילה מעל Σ קבוצה של מילים מעל Γ (קבוצת הפלטים החוקיים).

יש סוג מאוד ספציפי (אך מאוד חשוב) של בעיות חישוביות אשר נגדיר כעת

הגדרה 1.1.4 (בעיות הכרעה decision problems). אלה בעיות חישוביות שבהן:

1. הא"ב של הפלט הוא $\{\text{True}, \text{False}\}$
2. לכל קלט יש רק פלט אחד חוקי אשר הוא מילה באורך תו אחד

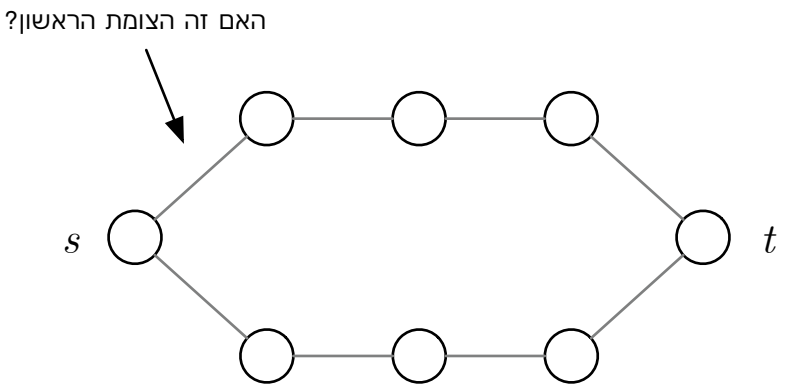
דוגמה 1.1.1 (לבעית הכרעה). בהינתן גרף $G = (V, E)$ יש להכריע אם הגרף קשר.

הקלט של בעיה זו הוא מחרוזת המתארת גרף, והפלט צריך להיות True במידה והגרף קשיר, ו־False אחרת.

בעיות אלה חשובות משתי סיבות:

1. אלה בעיות שהן פשוטות ביותר לתיאור
2. בעיות אלה כמעט ולא מורידות את הכלליות! הכוונה לכך היא שרוב הבעיות אשר נדון בהן שקולות לבעיית הכרעה כלשהי.

דוגמה 1.1.2. בעיית מציאת המסלול המינימלי בין שתי קשתות בגרף שקולה לסדרת הבעיות אשר מקבלת קשת ומכריעה אם קשת כלשהי היא הקשת הראשונה, השנייה, השלישית וכד' במסלול הקצר.



איור 2

נחזור שנייה לבעיית העצירה (שהיא דוגמה לבעיית הכרעה אשר כבר נידונה בשיעור זה), וננסה להבין מה קורה אם הקלט הוא לא תכנית חוקית? יש לשאלה זו כמה תשובות:

1. נגדיר את הקלט לפלט כזה להיות False
2. נתעלם מהאפשרות...

הגדרה 1.1.5 (שפה). שפה מעל א"ב Σ היא קבוצת המילים מעל Σ. נסמן את השפה בתור Σ^*. שפה יכולה להיות ריקה, סופית, או אינסופית.

דוגמה **1.1.3**. להלן דוגמאות לשפות:

1. דוגמה לשפה היא אוסף המילים מעל $\Sigma = \{0,1\}$ שאורכן אי־זוגי. נקבל כי 011 בשפה אך 0110 לא בשפה.
2. אוסף התכניות החוקיות בשפת Python שעוצרות כשמריצים אותן. המילה

```
print("Hello!")
```

היא בשפה, ואילו המילה

```
while True:
 print("Hello!")
```

לא בשפה משום שהיא לא עוצרת, והמילה

```
while (7 > 7:
```

לא בשפה עקב שגיאת קומפילציה.

בעצם, נוכל להתאים לכל בעיית הכרעה שפה שהיא אוסף המילים עבורן הקלט הוא True - ונוכל להתאים לכל שפה בעיית הכרעה אשר מחזירה True אמ"ם מילה היא בשפה. ההתאמות האלה הן הפוכות זו לזו!

דוגמה **1.1.4**. בעבור השפה שהיא שמות העצם בשפה העברית, מתאימה בעיית ההכרעה שמחזירה True אמ"ם מילה היא שם עצם.

1.2 מודלים חישוביים

דוגמה **1.2.1** (מודל חישובי, ניסיון מספר 1). תכניות Python או C וכד' - מתארת חישוב ומריצים אותה. הבעיה בשפות אלה היא שהן לאו דווקא מוגדרות, ויש מספר רב של גרסאות - והשפות עצמן מסובכות ביותר.

דוגמה **1.2.2**. נקבע א"ב $\Sigma = \{0,1\}$. נרצה להבין אם כל בעיית הכרעה מעל Σ ניתנת לפתרון בעזרת "מחשב" (כלומר: יישום של המודל החישובי)?

התשובה לכך היא שלא! יש בעיות הכרעה שאינן פתירות במודל החישובי שלנו: עוצמת קבוצת תכניות ה־Python היא $\aleph_0$, כי כל תכנית כזו היא מילה באורך סופי מעל א"ב Unicode שהוא גם סופי. מצד שני, עוצמת קבוצת השפות (כלומר בעיות ההכרעה) מעל Σ היא כמות תתי־הקבוצות של אוסף המילים מעל Σ, כלומר $\aleph_0$ ולכן היא $\mathfrak{c} = 2^{\aleph_0} > \aleph_0$. לכן עוצתמ קבוצת השפות גדולה ממש מעוצמת התכניות, ולכן קיימות שפות שאין להם תכנית מתאימה.

בהמשך הקורס נוכל לומר כי יש שפה ספציפית (בעיית העצירה) שאותה אי אפשר להכריע.

דוגמה **1.2.3** (מודל חישובי, ניסיון 2). אלן טיורינג המציא מודל חישובי אשר נגיע אליו בהמשך - מכונת טיורינג, אך היא מסובכת מדיי, אז לא נשתמש בו בינתיים

דוגמה **1.2.4** (מודל חישובי, ניסיון 3). האוטומטים הסופיים הדטרמיניסטיים Deterministic finite automata - DFA הם המודל החישובי בו נשתמש בחלק זה של השיעור.

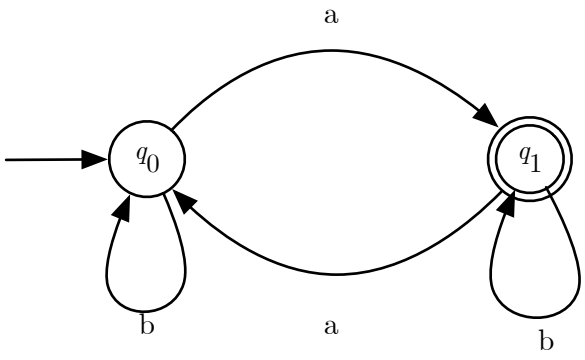
באוטומט זה יש מספר אלמנטים:

מצבים באוטומט הבא יהיו שני מצבים, q_0 ו־q_1

מצב התחלתי נסמנו לרוב q_0 ונסמנו עם חץ אשר נכנס אליו

כלל מעבר חץ העובר בין שני מצבים בקבלת קלט כלשהו

מצב קבלה בעת סיום על מצב q_1, האוטומט החזיר True, אחרת False ונסמנו עם שני עיגולים



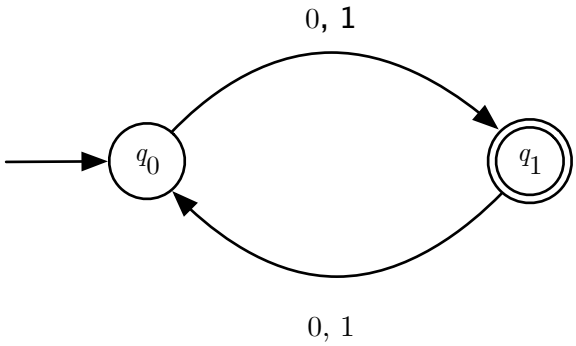
איור 3: אוטומט סופי דטרמיניסטי A

דוגמה **1.2.5.** אוסף המילים שאוטומט A מקבל הוא אוסף המילים בהם התו a מופיע מספר אי־זוגי של פעמים - נקרא לזה שפה L.

$$L = \{w \in \{a,b\}^* \mid \text{the number of } a\text{'s in } w \text{ is odd}\}$$

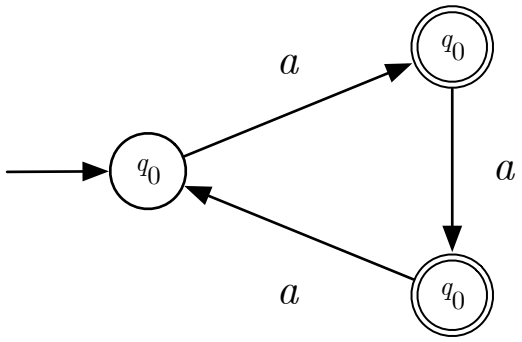
אוטומט A מכריע / מקבל את שפה L!

להלן דוגמה לאוטומט סופי דטרמיניסטי נוסף, המתאים לדוגמה 1 שנתנו לשפה.



איור 4: אוטומט סופי דטרמיניסטי המתאים לדוגמה 1 לשפה

דוגמה **1.2.6.** להלן דוגמה נוספת - ניקח את השפה $\Sigma = \{a\}$ ונגדיר את L להיות שפת המילים מעל Σ שאורכן לא מתחלק ב־3.



איור 5: אוטומט סופי דטרמיניסטי לשפה אשר ניסחנו הרגע

אם סיימו ב־q_0 אז מילה מתחלקת ב־3 והיא לא בשפה, ואילו במצבים האחרים אורך המילה לא תתחלק ב־3 ולכן המילה תהיה בשפה.

דוגמה **1.2.7** (שלא נסיים). נגדיר $\Sigma = \{0,1\}$ ונגדיר את M להיות שפת הייצוגים הבינאריים של מספרים שאינם מתחלקים ב־3. להלן תזכורת לייצוג בינארי:

$$(110)_2 = 2 + 4 = 6$$
$$(100)_2 = 4$$
$$(1111)_2 = 1 + 2 + 4 + 8 = 15$$

לבינתיים - שאלה פתוחה: איך נבנה אוטומט שכזה?