

# שפות תכנות, 236319

# פרופ' ד. לורנץ

אביב 2025



## תרגיל בית 6

### תאריך פרסום: 30.6.25

**מועד אחרון להגשה: 13.7.25 בשעה 23:59.**

## מתרגל אחראי: ענבר מנצ'ל

**מייל:** [inbarmenchel@campus.technion.ac.il](mailto:inbarmenchel@campus.technion.ac.il)

## הערות:

- שאלות על התרגיל יענו [בפורום הקורס בפיאצה בלבד](#).
- ניתן לפנות במייל במקרה של בעיות אדמיניסטרטיביות פרטניות, על כותרת המייל להתחיל ב-"236319 - תרגיל בית 6"
- הבהרות ותיקונים לתרגיל יפורסמו בגיליון זה **ויסומנו בצהוב**.
- לא תתאפשר הגשה לאחר מועד ההגשה כלל (אין הגשות באיחור)
- הסיבה היחידה שבגינה ניתן לקבל דחייה במועד הגשת תרגילי הבית היא מילואים. כדי לקבל דחיה עבור שירות מילואים, יש לפנות למתרגל האחראי על התרגיל במייל.
- הגשה בזוגות. לפני ההגשה, ודאו כי ההגשה שלכם תואמת את הנחיות ההגשה בסוף התרגיל. **הגשות שלא יעמדו בהנחיות יפסלו על הסף.**

# חלק יבש

לפניכם מספר שאלות, חלקן על דברים שראינו בכיתה וחלקן לא. על מנת לענות על השאלות ניתן להיעזר בחיפוש באינטרנט ובבינה מלאכותית ✨. שימו לב, איפה שאין סימן ל-AI, מומלץ מאוד לפתור לבד למען הלמידה. כתבו את הפתרונות אליהן בקובץ **dry.pdf**. בשאלות של כתיבת קוד, **יש לכתוב את הקוד** במסמך ההגשה ולא לצרף תמונות.

## שאלה 1 ✨

המתרגלים בקורס שפות תכנות רצו לכתוב שאלות למבחן בקורס, אך מרוב שהשקיעו בשאלות שיעורי הבית נגמרו להם הרעיונות הטובים לשאלות. הציעו 2 שאלות בִּיָּרָה למבחן (שאלות אמריקאית) על אודות חומר הלימוד שהוצג בהרצאות או בשקפים. על כל שאלת בררה להציע בדיוק 4 תשובות אפשריות, שמתוכן על הנבחן לברור תשובה אחת, הנכונה ביותר. לכל שאלה ציינו מה התשובה הנכונה ונמקו.

טיפים - ראו הנחיות לכתיבת שאלות בררה מוצלחות (חלק ג' במסמך [הזה](#)).

## שאלה 2

לאור הצלחת הסדרה "משחקי הדיונון", חשב סגל הקורס לייצור משחק בסגנון משחקי הדיונון עבור הסטודנטים בקורס. אך לאחר שנתקל בקשיים טכניים ליישם את המשחק במציאות, החליט להתפשר ולייצר גרסה דומה (ובטוחה יותר).

עבור האתגר הראשון, החליט סגל הקורס לייצור משחק בסגנון "Red light, green light" אשר מוצג בסדרה, אך כעת נקרא למשחק red cut, green cut, ובו על הסטודנטים להסביר עבור כל חתך (cut) בפרולוג, האם הוא אדום או ירוק. כאשר המנצח ממשיך לשלב הבא, והמפסיד צריך לכתוב מחדש את תרגיל 4.

- (1) הסבירו מהו ההבדל בין חתך אדום לחתך ירוק בפרולוג.
  - (2) עבור כל אחד מהפריטים הבאים, הסבירו האם החתך הוא אדום או ירוק. נמקו היטב.
- הערה: ניתן להניח כי המשתנה הראשון קשור, ובמקרה של min\_squid גם שהשני קשור.

**min\_squid(A,B,A):- A<=B,!.**

**min\_squid(A,B,B):- A>B.**

**signed\_squid(S,low\_squid):-S<0,!.**

**signed\_squid(S,no\_squid):-S=0,!.**

**signed\_squid(S,high\_squid).**

(3) שימו לב, **סעיף 3 בלבד!** הוא סעיף בונים.  
לאחר שהפסיד במשחק, והבין שעליו לעשות מחדש את כל תרגיל 4, אחד הסטודנטים החליט לנסות לנצח במשחק בכך שישנה את הפרדיקטים כך שיתאימו לתשובה שבחר. עזרו לסטודנט לבצע את השינוי הזה. כלומר, בחרו את אחד הפרדיקטים **min\_squid** או **signed\_squid**, בהתאם לתשובה שלכם לסעיף 2, ושנו את הקוד כך שכעת התשובה תהיה הפוכה. למשל, אם בסעיף 2 קבעתם כי **min\_squid** הוא חתך אדום, אז כתבו את **min\_squid** מחדש, כך שכעת החתך בו ירוק.

(4) סעיף זה הוא סעיף **חובה!**  
לצערו של הסטודנט, סגל הקורס גילה את תחבולתו והוא נאלץ לעשות מחדש את כל תרגיל 4. מרוב כעס, הצהיר כי הוא לא מבין מה היתרון בחתכים בכלל, וכי עדיף לכתוב קוד בלעדיהם. האם אתם מסכימים עם הסטודנט? אם כן, הסבירו מה החסרונות של חתכים. אם לא, הסבירו מהם היתרונות של חתכים.

## שאלה 3

**בשאלה זו ישנם 3 סעיפים שאינם תלויים זה בזה. עליכם לבחור 2 מתוך 3 הסעיפים ולענות עליהם. (אם תענו על כולם, יבדקו רק השניים הראשונים).**

(1) עבור כל אחת מהשפות הבאות, הסבירו האם היא **gradually typed**. נמקו היטב. עליכם לתת דוגמא לפחות לאחת מהשפות אשר מדגישה את היותה **gradually typed** או לא **gradually typed**.

א) Cython.

ב) C#.

ג) C.

(2

א) הסבירו מהו המושג duck typing, ותנו דוגמא בשפה לבחירתכם אשר משתמשת בסגנון זה.

ב) סטודנט מבולבל מעוניין לממש duck typing בשפה חדשה שהוא מתכנן, אבל הוא אינו יודע כיצד להתייחס לתאימות של טיפוסים בשפה, כך שתתאים ל-duck typing. האם עליו להשתמש בתאימות **nominal** או **structural**? נמקו.

(3 בתרגולים ראינו את מנגנון ה-RTTI.

א) הסבירו מדוע המנגנון נחוץ בשפות בעלות טיפוסיות דינמית.

ב) האם יש יתרונות בשימוש במנגנון גם בשפות בעלות טיפוסיות סטטית?

## חלק רטוב

- קבצים המסופקים לחלק זה תוכלו למצוא בגיטהאב של הקורס [homework6](#).
- פתרון לשאלה 1 יש לכתוב בקובץ ששמו [q1.pl](#).
- בנוסף לכל שאלה מסופקת לכם דוגמת קלט ופלט רצוי. לצורך העניין את הפתרון שלכם תוכלו לבדוק בעזרת הפקודות הבאות:

```
swipl -q q1.pl < q1.in | sed '1,/===TEST START===/d' > q1.out  
diff q1.out q1.expected
```

- סיפקנו עבורכם סביבת פיתוח מוכנה לProlog ב-[מדריך התקנה ושימוש](#)
- יש לבדוק, לקמפל ולהריץ את פתרונותיכם לתרגיל זה בסביבה הנ"ל.

## שאלה 1

דריה, סטודנטית בוגרת הקורס בשפות תכנות, רצתה לדעת האם אכן השיר: "never gonna give you up" של Rick Astley הוא נכון. כלומר, האם באמת הוא לעולם לא יוותר על אהובתו. לשם כך, היא רצתה לתכנן מערכת בפרולוג אשר תוכל לבדוק האם תכונות כלשהן מתקיימות תמיד או אף פעם.

לאחר מחשבה רבה, היא נזכרה כי למדה בקורס אחר בפקולטה על לוגיקה טמפורלית, אשר יכולה לסייע בפתרון הבעיה. לוגיקה טמפורלית היא הרחבה של הלוגיקה הפשוטה המוכרת לכם (and,or,not) אשר מוסיפים לה אופרטורים נוספים על מנת לייצג זמן.

בשאלה זו, עליכם לבנות את הפרדיקט satisfy אשר מקבל מצב ובנוסף נוסחה בלוגיקה טמפורלית, ומחזיר האם היא ספיקה (לגבי כל אופרטור חדש, מוגדר בסעיף המתאים מתי הוא מסתפק). ובעזרתו, תוכל דריה לבדוק את נכונות השיר.

### הגדרות חשובות:

- אנו נייחס את כל המצבים האפשריים בגרף מכוון. עבור כל צומת, הפרדיקט  $vertex/2$  מסתפק אמ"ם List היא רשימת התכונות שהמצב מקיים. למשל

**`vertex(1,['hello'])`**

אומר כי צומת 1, בעלת התכונה 'hello' בלבד.

- אם יש קשת בין שני מצבים, מסומנת ע"י הפרדיקט  $edge/2$ , אז נאמר כי Y בא לאחר X ישירות.

- אם יש מסלול בין שני מצבים X ו-Y, אז נאמר כי Y בא לאחר X (לא בהכרח ישירות).

- בכל נוסחה, כאשר נרצה לייצג תווית נשתמש ב-atom. למשל, הנוסחה:

**`atom('hello')`**

תסתפק עבור צמתים בעלי התכונה 'hello'.

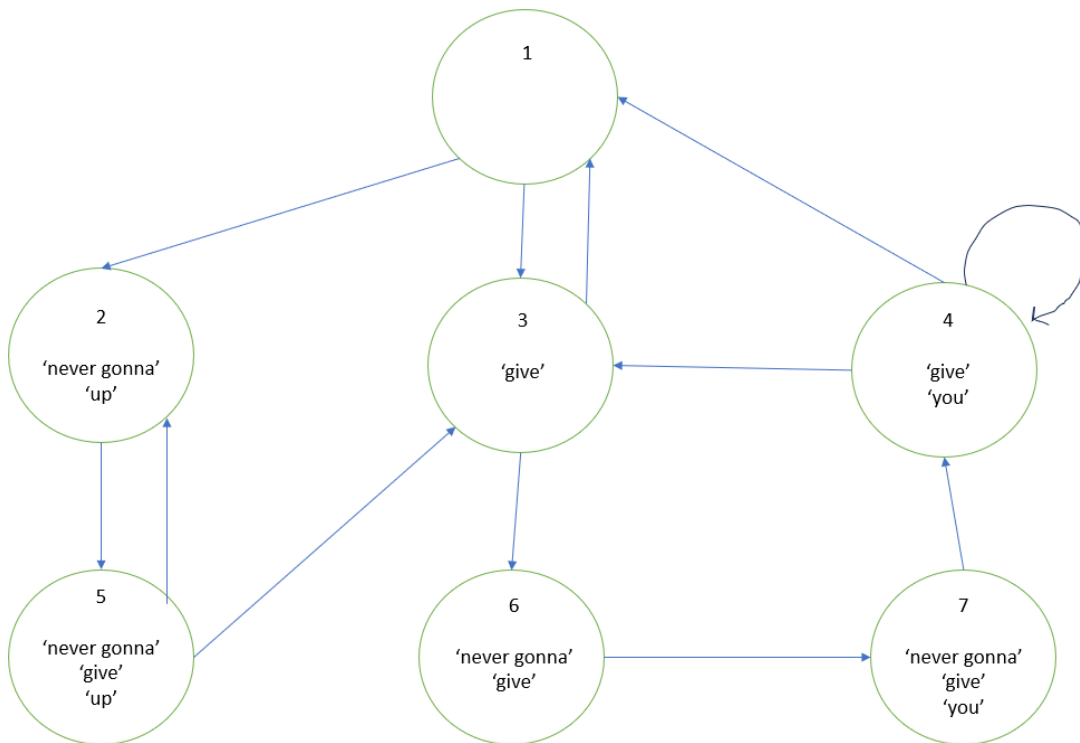
### הערות:

- ניתן להניח שמספר הצמתים בגרף הוא סופי.

- לא ניתן להניח כי הגרף חסר מעגלים.

- אפשר, ואף מומלץ, לממש פרדיקטים לעזר.

1) ראשית, נרצה ליצור את המצבים והקשרים ביניהם.  
כתבו בפרולוג את הקוד אשר יוצר את הגרף הבא:



הערה: עליכם להשתמש בפרדיקטים vertex ו-edge אשר הוגדרו לכם.

**הערה חשובה: עליכם לדאוג כי בקובץ הפתרון, הפתרון של סעיף 1 יהיה מעל הפתרונות של שאר הסעיפים!**

2) הוסיפו לפרדיקט satisfy אפשרות לבדיקה עבור האופרטורים הלוגיים הבסיסיים, כך שהקוד הבא יעבוד. כלומר, יחזיר true או false בהתאם לתכונות אותן מקיים או לא מקיים המצב.

**satisfy(1,atom('never gonna')).**  
**satisfy(2,not(atom('give'))).**  
**satisfy(3,and(atom('give'),atom('you'))).**  
**satisfy(4,or(atom('up'),atom('!'))).**

3) הוסיפו לפרדיקט satisfy אפשרות לבדיקה עבור האופרטור next. אופרטור next מסתפק רק כאשר קיים מצב שבא לאחר מצב הנבדק ישירות, והוא מקיים את הנוסחה אשר ב-next.  
לדוגמא:

**satisfy(1,next(atom('never gonna'))).**

יוחזר true רק כאשר למצב 1, קיים מצב X אשר קיים אחריו ישירות, כאשר ל-X יש את התכונה 'never gonna'.

(4) הוסיפו לפרדיקט satisfy אפשרות לבדיקה עבור האופרטור eventually. אופרטור eventually מסתפק רק כאשר קיים מצב שמגיע לאחר מצב הנבדק, והוא מקיים את הנוסחה אשר ב-eventually. לדוגמא:

**satisfy(1,eventually(atom('give'))).**

יוחזר true רק כאשר למצב 1, קיים מצב X אשר קיים אחריו, כאשר ל-X יש את התכונה 'give'.

(5) הוסיפו לפרדיקט satisfy אפשרות לבדיקה עבור האופרטור until. אופרטור until מסתפק רק כאשר קיים מסלול בו עד אשר מתקיים תנאי הסיום, תמיד מתקיים תנאי ההתחלה. לדוגמא:

**satisfy(1,until(atom('you'),atom('finish'))).**

במקרה זה, יוחזר true רק אם קיים מסלול אשר מתחיל מצומת 1, בו עד אשר נמצא צומת בעלת התכונה 'finish', תתקיים התכונה 'you'. הערות:

- לאחר שהתקיים תנאי הסיום, אין דרישה על המשך המסלול.
- תנאי ההתחלה חייב להתקיים לפחות פעם אחת. כלומר, בדוגמא שלנו, אם vertex(1,['finish']) אז יוחזר false.

(6) הוסיפו לפרדיקט satisfy אפשרות לבדיקה עבור האופרטור globally. אופרטור globally מסתפק רק כאשר בכל מסלול ממנו, התכונה מתקיימת לכל אורכו. לדוגמא:

**satisfy(1,globally(atom('up'))).**

במקרה זה, יוחזר true רק אם כל מסלול אשר יוצא מהצומת 1, תמיד מקיים לכל אורכו את התכונה 'up'.

שימו לב: ייתכנו מעגלים בגרף, ולכן ייתכנו גם מסלולים באורך אינסופי. רמז: נסו לחשוב אילו צמתים רלוונטים לבדיקה הזו.

## הנחיות הגשה

- את הפתרון לתרגיל הבית יש להגיש בקובץ zip המכיל את הקבצים הבאים בלבד:

q1.pl, dry.pdf.

- על החלק היבש להיות מוקלד, אין להגיש סריקה או צילום של התשובות לחלק זה.
- שם קובץ ההגשה יהיה EX6\_ID1\_ID2.zip כאשר ID1, ID2 הם מספרי ת.ז. של המגישים.
- בודקי התרגילים **מאוד** אוהבים Memes. שתפו את תחושותיכם במהלך פתירת התרגיל באמצעות Memes מתאימים ב-pdf של החלק היבש. Memes מצחיקים בצורה יוצאת דופן יזכו את היוצרים בבונוס.

**בהצלחה!**