

שפות תכנות, 236319

פרופ' ד. לורנץ

אביב 2025



תרגיל בית 5

תאריך פרסום: 12.6.25

מועד אחרון להגשה: 13.7.25 בשעה 23:59.

מתרגל אחראי: ענבר מנצ'ל

מייל: inbarmenchel@campus.technion.ac.il

הערות:

- שאלות על התרגיל יענו [בפורום הקורס בפייסבוק בלבד](#).
- ניתן לפנות במייל במקרה של בעיות אדמיניסטרטיביות פרטניות, על כותרת המייל להתחיל ב-"236319 - תרגיל בית 5"
- 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 0

חלק א'

בהרצאה הוצגו לכם בנאים תיאורטיים שונים בעזרת שפת **Mock**. בשאלה זו נראה איך התיאוריה מתארת את המציאות.

- 1) בהרצאה ראיתם את הבנאים התיאורטיים. בחרו 4 בנאים, ועבור כל אחד, שפה אחת אשר מממשת אותו, ותנו דוגמא לקטע קוד אשר מציג שימוש בו בשפה.
 - עליכם להשתמש בשפה שונה לכל אחד. (השתמשו בדוגמאות שונות משנראו בהרצאה או בתרגול 😊)
 - הערה: לנוחיותכם, השתמשו בטבלה בסוף פרק 3.3 בחוברת השקפים (פרט ל-subrange).
 - בonus: ישנם 9 בנאים סך הכל בטבלה (פרט ל-subrange). פתרו את השאלה עבור כולם. השתמשו בשפות שונות לכל בנאי.

2) בחרו שני בנאים תיאורטיים אשר נלמדו בהרצאה.

- א) עבור אחד מהם, הסבירו מדוע הוא נחוץ, ומדוע הגדרה שלו בשפה יכולה לסייע. הדגימו באמצעות קוד.
- ב) עבור הבנאי השני, הסבירו מדוע אפשר לוותר עליו, ומדוע הוא אינו בהכרח נחוץ.

3) האם לדעתכם כדאי שלכל שפה יהיה מקביל לטיפוסים התיאורטיים של **top** ו-**bottom**? נמקו את תשובתכם.

חלק ב'

לאחר שסיים את תרגיל בית 4, החליט סטודנט בקורס לנוח קצת מהעבודה הקשה, וליצור משחק של [מבוכים ודרכונים](#) בשפת פייתון. מצורף לכם קובץ של אחד המודלים שהכין, [dnd.py](#), עבור שחקן מסוג מכשף. תבדקו סטטית את הקוד המסופק ב-[mypy Playground](#). הקוד אמור לעבור את הבדיקה בהצלחה. בשאלה זו ננסה לעזור לסטודנט להבין מה השגיאות שביצע, וגם לסייע לו להבין את היתרונות והחסרונות ב-[gradual typing](#).

הערה: שימו לב, הפונקציה `convert_spell_list_to_spell_set` המוגדרת בקובץ צריכה להיות `convert_spell_set_to_spell_list`.

1. הוסיפו [type annotations](#) לכותרת של כל פונקציה בקובץ. אין צורך להוסיף type annotation למילה `self`. צרפו את הקובץ המעודכן להגשה שלכם.
2. תריצו שוב את הבדיקות הסטטיות. הסבירו כל שגיאת טיפוסים אשר התגלה.
3. הסבירו מה הסיבה לכך שהשגיאות אלו לא התגלו קודם. מה הדבר מלמד אותנו על [gradual typing](#)?

חלק רטוב

- קבצים המסופקים לחלק זה תוכלו למצוא בגיטהאב של הקורס [Homework5](#).
- פתרון לשאלה 1 יש לכתוב בקובץ ששמו [q1.sml](#).
- פתרון לשאלה 2 יש לכתוב בקובץ ששמו [q2.pl](#).
- בנוסף לכל שאלה מסופקת לכם דוגמת קלט ופלט רצוי. לצורך העניין את הפתרון שלכם תוכלו לבדוק בעזרת הפקודות הבאות:

```
smlnj q1.sml < q1.in | sed '1,/===TEST START===/d' > q1.out
diff q1.out q1.expected
swipl -q q2.pl < q2.in | sed '1,/===TEST START===/d' > q2.out
diff q2.out q2.expected
```

- סיפקנו עבורכם סביבת פיתוח מוכנה ל SML ב-[מדריך התקנה ושימוש](#)
- סיפקנו עבורכם סביבת פיתוח מוכנה ל Prolog ב-[מדריך התקנה ושימוש](#)
- שימו לב, יש לשנות את הפקודה:

```
git clone https://github.com/Whitomtiti/VSCoDe-Prolog.git
```

ל-

```
git clone https://github.com/Technion-Programming-Languages/VSCoDe-Prolog.git
```

- יש לבדוק, לקמפל ולהריץ את פתרונותיכם לתרגיל זה בסביבות הנ"ל.

שאלה 1

לאחר שהוולגנים הרסו את כדור הארץ על מנת לבנות במקומו כביש מהיר בין-כוכבי, יצא ארתור, בין האנשים הבודדים ששרדו, למצוא את מחשב העל "הרהור עמוק" כדי שיוכל לעזור לו להביס את הוולגנים.

בשאלה זו נעזור לארתור למצוא את מחשב העל "הרהור עמוק" לפני שהוולגנים ימצאו אותו.

ארתור גילה שמחשב העל נמצא בכוכב מגרתיאה, אשר אנו נייצג כמטריצה עצלה. כלומר, בדומה לרשימות עצלות שראינו בתרגול:

רשימה עצלה רגילה:

```
datatype 'a Seq = Nil | Cons of 'a * (unit -> 'a Seq)
```

מטריצה עצלה:

```
datatype 'a DSeq = DNil | DCons of 'a * (unit -> 'a DSeq)
* (unit -> 'a DSeq);
```

כל מקום בכוכב מגרתיאה מכיל tile אשר יכול להיות מאוכלס באחת מבין האפשרויות הבאות:

- חור שחור (BLACKHOLE): אם הגענו אליו, נשאבים אליו ואין דרך לצאת. ולכן אסור להיכנס למשבצת עם חור שחור.
- אקדח לייזר (LASER_GUN): אם הגענו אליו, אז נאסוף אותו. אין הגבלה על מספר אקדחי לייזר שמותר להחזיק.
- מרווין (MARVIN): מרווין הוא רובוט דכאוני, ולכן אם לא נפגע בו עם אקדח לייזר, הוא יגרום לארתור להיות מדוכדך, והוא כבר לא ירצה להמשיך לחפש את מחשב העל. לכן, מותר להיכנס למשבצת בה מרווין נמצא, רק אם יש בידינו אקדח לייזר (או 42, כפי שיוסבר בהמשך). במידה וכן פגשנו את מרווין, עלינו לזרוק אקדח לייזר יחיד.
- 42 (FOURTY_TWO): במידה ומצאנו את 42, התשובה לחיים, ליקום, ולכל השאר, אז נוכל להיכנס לכל משבצת בה מרווין נמצא ולהביס אותו, גם ללא אקדחי לייזר.
- "הרהור עמוק" (DEEPTHOUGHT): זו משבצת הסיום, ואליה יש להגיע. ניתן להניח כי יש לכל היותר מחשב יחיד בכוכב.

tile הוא datatype אשר מוגדר כך:

```
datatype tile = BLACKHOLE | LASER_GUN | MARVIN |
FOURTY_TWO | DEEPTHOUGHT;
```

בנוסף, עקב חוקי הכבידה על הכוכב, ארתור יכול לזוז אך ורק ימינה או למטה.

```
datatype direction = DOWN | RIGHT;
```

הערה חשובה:

הדביקו את הקוד הבא בתחילת קובץ הפתרון שלכם:

```
datatype tile = BLACKHOLE | LASER_GUN | MARVIN |  
FOURTY_TWO | DEEPTHOUGHT;  
datatype direction = DOWN | RIGHT;  
datatype 'a Seq = Nil | Cons of 'a * (unit -> 'a  
Seq);  
datatype 'a DSeq = DNil | DCons of 'a * (unit -> 'a  
DSeq) * (unit -> 'a DSeq);
```

```
exception NoDeepThought;
```

שימו לב: אם בשאלה זו אתם ממשים פונקציות עזר (כפי שמומלץ), עליהן להיות ב-local כפי שלמדנו בתרגולים.

הערה: בפתרון השאלה, אתם יכולים להניח כי תמיד המבנה יהיה מטריצה (ולא עץ למשל). בנוסף, ניתן להניח כי המקום ה- i,j במטריצה מוגדר היטב. כלומר, ללא תלות בדרך בו הגעתם למקום.

למשל, אם ביצעתם צעד ימינה ואז למטה, או למטה ואז ימינה, בהכרח תגיעו לאותו המקום.

(1) ממשו את הפונקציה:

```
val good_path = fn : tile Seq -> bool;
```

אשר מקבלת מסלול של משבצות ברשימה עצלה, ומחזירה האם המסלול חוקי. כלומר:

- מסתיים במשבצת בה קיים המחשב "הרהור עמוק".
- מקיים את כל ההגבלות אשר הוגדרו לגבי כניסה למשבצות שונות.

(2) ממשו את הפונקציה:

```
val get_flat_path = fn : 'a DSeq -> direction Seq ->  
'a Seq;
```

אשר מקבלת את המשבצות אשר מייצגות את הכוכב מגרתיאה, ורשימה עצלה של כיווני תזוזה, ומחזירה רשימה עצלה של המקומות בהם נעבור לאורך המסלול בכוכב מגרתיאה.

שימו לב, המטריצה העצלה בנויה כך האיבר הראשון הוא צעד למטה, ואילו השני הוא צעד ימינה.

- ניתן להניח כי המסלול אשר יוחזר הוא חוקי, וכי מימדי הכוכב גדולים מספיק.

(3) ממשו את הפונקציה:

```
val find_all_paths = fn : int * int -> direction Seq  
list;
```

אשר זוג של שורה וטור, ומחזירה רשימה של כל הרשימות העצלות אשר מייצגות דרך להגיע לאותה המשבצת מההתחלה (המקום (0,0), תחילת המטריצה העצלה), ללא תלות בתוכן המשבצת.
לדוגמא:

עבור (1,1), יוחזר: [[DOWN,RIGHT],[RIGHT,DOWN]].

(4) לאחר שסיימנו להגדיר את פונקציות העזר, נסייע לארתור להגיע ראשון אל "הרהור עמוק". ממשו את הפונקציה:

```
val minimal_distance_for_deep_thought = fn : tile  
DSeq -> int;
```

הפונקציה מקבלת את משבצות הכוכב מגרתיאה, ומחזירה את אורך המסלול החוקי הקצר ביותר מההתחלה (המקום (0,0), תחילת המטריצה העצלה), אל המחשב "הרהור עמוק".

במידה והמחשב אינו קיים כלל, יש לזרוק את החריגה NoDeepThought אשר מוגדרת לעיל.

הערות כלליות:

- מומלץ להשתמש בפונקציות אשר מימשותם בסעיפים 1,2,3.
- נסו להבין (או להרהר) קודם האם המחשב "הרהור עמוק" קיים בכלל, ורק אז להתחיל לבדוק מסלולים קצרים ביותר.

- אם אין מסלול חוקי אל המחשב "הרהור עמוק", יש להתייחס למצב כאילו המחשב אינו קיים כלל בכוכב.

שאלה 2

בשאלה זו נממש מספר פרדיקטים ב-Prolog עבור סוגים שונים של סדרות מתמטיות.

1. הגדירו את הפרדיקט $\text{fib}/2$ המתקיים אם הפרמטר הראשון הוא האיבר ה-N בסדרת פיבונאצ', כאשר N הוא הפרמטר השני.
ניתן להניח כי הפרמטר השני קשור, כלומר בעל ערך קונקרטי בזמן הפעלת הפרדיקט.
לדוגמה:

?- fib(X,7).

X=8.

?- fib(1,2).

true.

הערה: הניחו כי האיבר הראשון בסדרה הוא 0, וכי האיבר השני בסדרה הוא 1.

2. סדרת פאדובן (padovan) היא סדרה אשר

- שלושת האיברים הראשונים הם 1.
- כל איבר אחר הוא סכום שני האיברים הקודמים לו שאינם האיבר אשר קודם לו ישירות. כלומר, אם למשל מדובר ב-1,2,3, אז האיבר הבא הוא 3, שכן הוא הסכום של 1 ועוד 2.

הגדירו את הפרדיקט $\text{padovan}/2$ המתקיים אם הפרמטר הראשון הוא האיבר ה-N בסדרת פאדובן, כאשר N הוא הפרמטר השני.
ניתן להניח כי הפרמטר השני קשור, כלומר בעל ערך קונקרטי בזמן הפעלת הפרדיקט.
לדוגמה:

?- padovan(X,7).

X=4.

?- padovan(1,3).

true.

3. מספרי קטלן הם סדרה של מספרים אשר מופיעים במגוון של בעיות שונות בקומבינטוריקה.
לדוגמא:

- מספר ביטויי סוגריים חוקיים.
 - מספר הטריאנגולציות של מצולע קמור.
 - מספר מסלולי סריג אשר אינם עוברים את האלכסון הראשי.
- לסדרת מספרי קטלן יש מספר נוסחאות (סגורות או רקורסיביות) לחישוב האיברים בה. בסעיף זה, ניתן להיעזר בכל אחת מהדרכים הללו, או להמציא דרך נוספת (שעובדת) בעצמם.
- הגדירו את הפרדיקט $catalan/2$ המתקיים אם הפרמטר הראשון הוא האיבר ה-N בסדרת קטלן, כאשר N הוא הפרמטר השני.
- ניתן להניח כי הפרמטר השני קשור, כלומר בעל ערך קונקרטי בזמן הפעלת הפרדיקט.
- לדוגמא:

?- $catalan(X,7)$.

$X=429$.

?- $catalan(1,1)$.

true.

הבהרה בנוגע לספירה של מספרי קטלן:

בסעיפים 3 ו-4 זה הספירה של מספרי קטלן היא החל מ- $n=1$. כלומר, נתייחס למספרי קטלן כסדרה:

1,2,5,14...

שאלת בונוס

4. הגדירו את הפרדיקט $multisum/4$ המתקיים אם

- הפרמטר הראשון הוא N.
 - הפרמטר השני הוא סכום N מספרי פיבונאצ'י הראשונים.
 - הפרמטר השלישי הוא סכום N מספרי פאדובן הראשונים.
 - הפרמטר הרביעי הוא סכום N מספרי קטלן הראשונים.
- ניתן להניח כי הפרמטר הראשון קשור, כלומר בעל ערך קונקרטי בזמן הפעלת הפרדיקט.

לדוגמא:

?- $multisum(7,X,Y,Z)$.

$X=20$.

```
Y=14.
```

```
Z=625.
```

```
?- multisum(2,1,2,3).
```

```
true.
```

הנחיות הגשה

- את הפתרון לתרגיל הבית יש להגיש בקובץ zip המכיל את הקבצים הבאים בלבד:
q1.sml, q2.pl, dnd.py, dry.pdf.
- על החלק היבש להיות מוקלד, אין להגיש סריקה או צילום של התשובות לחלק זה.
- שם קובץ ההגשה יהיה EX5_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!