

שפות תכנות, 236319

פרופ' ד. לורנץ
אביב 2024



תרגיל בית 6

תאריך פרסום: 11.8.2024

מועד אחרון להגשה: 25.8.2024 בשעה 23:59.

מתרגל אחראי: אנדריי בבין

מייל: andrey.babyn@campus.technion.ac.il

הערות:

- שאלות על התרגיל יענו [בפורום הקורס בפיאצה בלבד](#).
- ניתן לפנות במייל במקרה של בעיות אדמיניסטרטביות 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 בִּרְרָה** 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

בשאלה זו נעסוק בחתכים (**cuts**) בפרולוג.

1. מה ההבדל בין חתך ירוק לחתך אדום?

2. עבור הפרדיקט **sign/2**, הכריעו האם כל אחד מהחתכים הוא אדום או ירוק.

עבור כל חתך ירוק, הסבירו את המוטיבציה להוסיף אותו לפרדיקט.

עבור חתך אדום, תנו דוגמה לשאילתה עבורה הפרדיקט יתן תוצאה שונה אם מסירים את החתך.

הניחו כי הפרמטר הראשון תמיד קשור.

sign(X, negative) :- X < 0, !.

sign(X, zero) :- X = 0, !.

sign(X, positive).

3. הסבירו את ההתנהגות הבאה של הפרדיקט:

?- sign(-5, X).

X = negative.

?- sign(-5, positive).

true.

כלומר, מה הסיבה לכך שפרולוג לא מוצא את האטום **positive** כהשמה אפשרית

ל-X כאשר הוא אכן מקבל את **sign(-5, positive)**.

שאלה 3

בתרגול למדנו על שפת התכנות TypeScript כדוגמה לשפה עם טיפוסיות הדרגתית (Gradual Typing).

1. ראינו שב-TypeScript ניתן להגדיר טיפוס `Record` באופן הבא:

```
type Record = {x: number, y: number};
```

האם תאימות בין טיפוסים כאלו הוא `Nominal` או `Structural`? מה לגבי תאימות בין טיפוסים המוגדרים על ידי מחלקות? הסבירו והביאו דוגמאות קוד התומכות בטענה שלכם. ✨

2. בתרגול ראינו שמערכת הטיפוסים של השפה אינה בטוחה (safe), מכיוון שניתן להשתמש ב-`any` כדי לבטל בדיקות טיפוסים והטיפוסים אינם נבדקים בזמן ריצה. בנוסף לכך, מערכת הטיפוסים של השפה אינה בטוחה גם ללא שימוש ב-`any`. הביאו דוגמת קוד בה **לכל** הערכים יש טיפוס **סטטי קונקרטי** מוגדר, אך בזמן ריצה עדיין תתרחש שגיאת טיפוס.

הערות:

- תוכלו להשתמש ב-[TypeScript Playground](#) כדי לנסות את הדוגמאות שלכם.
- אין להשתמש ב-`type assertions` (המקביל ב-TypeScript להמרות טיפוסים).

רמזים לסעיף 2:

a. בהינתן טיפוס B היורש מ-A, הקומפילר מתייחס למערך של הראשון כתת-טיפוס של מערך של השני, כלומר קטע הקוד הבא חוקי:

```
let bArr: B[] = [new B(), new B()];  
let aArr: A[] = bArr;
```

b. כאשר B תת-טיפוס של A, ניתן גם לבצע השמות של B למשתנים מסוג A וגם להעביר ערך מסוג B לפונקציה המצפה לקבל ארגומנט מסוג A.

c. עבור טיפוס נוסף C היורש מ-A (ולא מ-B), נסו לגרום לכך שיכנס איבר מסוג C למערך של איברים מסוג B.

חלק רטוב

- קבצים המסופקים לחלק זה תוכלו למצוא בגיטהאב של הקורס - [Homework 6](#).
- 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-VSCode. [מדריך התקנה ושימוש](#).
- יש לבדוק, לקמפל ולהריץ את פתרונותיכם לתרגיל זה בסביבה הנ"ל.

• **הטסטים עודכנו ב-15.08.2024.**

שאלה 1

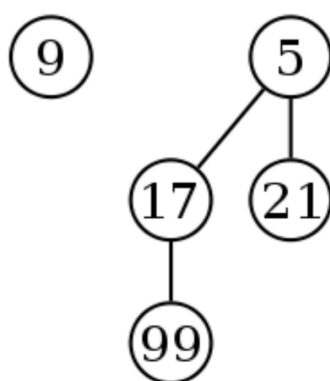
ד"ר סדר מהפקולטה למתמטיקה התגאה מהעבודה המוצלחת שלכם עם רצפים בתרגיל הבית הקודם והחליט להמשיך שיתוף פעולה עם הקורס שלנו.

הוא חקר את שפת תכנות פרולוג ומאוד התלהב ממנה, על כן ד"ר סדר רוצה שתעזרו לו לבנות מיון של מספרים שלמים בה כחלק של המחקר שלו ב-"**מבנים בינומיאלי**".

נתחיל מכמה הגדרות שסופקו ע"י ד"ר סדר:

- עץ מקיים את תכונת **הערימה** אם המספר שרשום בכל צומת פנימית קטן יותר מכל המספרים שמתחתיו. לשם פשטות נניח שכל הערכים שבעץ הם **שונים**.
- **עץ בינומיאלי** הוא עץ שמקיים את תכונת הערימה:
 - עץ בינומיאלי מסדר 0 מכיל צומת אחת בלבד.
 - עץ בינומיאלי מסדר k מכיל בסה"כ 2^k צמתים. צומת אחד בשורש, ועוד $2^k - 1$ צמתים המסודרים באופן הבא: **הבן הראשון הוא עץ בינומיאלי מסדר $k-1$, הבן השני הוא עץ בינומיאלי מסדר $k-2$, ואילו הבן ה- k הוא עץ בינומיאלי מסדר 0.**

דוגמה לעץ מסדר 0 אשר מכיל מספר 9 ולעץ מסדר 2:



סעיף א'

ממשו פרדיקט $bt/2$ אשר מייצג **עץ בינומיאלי** ומורכב משני פרמטרים:

1. ערך הצומת בשורש.
2. רשימה (שיכולה להיות ריקה) של העצים הבינומיאליים שהם בנו, כלומר כל פריט ברשימה הוא פרדיקט bt בעצמו.

ניתן להניח כי שני הפרמטרים קשורים.

הקפידו לבדוק כי כל ההגדרות של עצים בינומיאליים מתקיימים עבור bt .

דוגמה:

```
?- bt(5, [bt(7, [])]).
true.
?- bt(7, [bt(5, [])]).
false.
```

סעיף ב'

לפי ד"ר סדר, בהינתן שני עצים בינומיאליים, מסדר k , קל ליצור מהם עץ בינומיאלי מסדר $k+1$. הוספת עץ אחד כבן אחרון לעץ האחר, תעשה זאת. כדי לשמור על תכונת הערימה, נבחר תמיד להוסיף את העץ שהערך בשורש שלו גדול יותר, כבן נוסף לצומת שהערך בשורש שלו קטן יותר.

ממשו פרדיקט `merge_bt/3` אשר נכון אמ"מ שני הפרמטרים הראשוניים הם עצים בינומיאליים מאותו סדר והפרמטר השלישי הוא תוצאת המיזוג של שני העצים לפי השיטה הנ"ל.

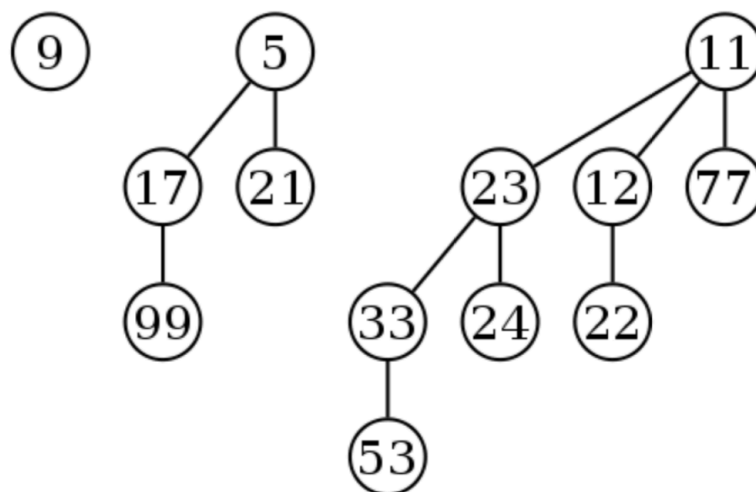
ניתן להניח כי שני הפרמטרים הראשוניים הם קשורים.

דוגמה:

```
?- merge_bt(bt(3, [bt(4, [])]), bt(5, [bt(7, [])]), X).  
X = bt(3, [bt(5, [bt(7, [])]), bt(4, [])]).
```

סעיף ג'

נגדיר מבנה בינומיאלי חדש. ערימה בינומיאלית היא רשימה של פריטים, כך שבמקום ה- i יכול להופיע עץ בינומיאלי מסדר i או פריט ריק. בפרט, מבנה ערימה בינומיאלית שבה n מספרים בסה"כ נקבע בדיוק על פי הייצוג בביטים של n . הנה לדוגמא בערימה שיש בה 13 מספרים תהיה רשימה שבה העצים הבינומיאליים הבאים:



עצים אלו יופיעו במקומות 0, 2 ו-3 ברשימה. המקום ה-1 ברשימה הוא ריק. שימו לב לכך שהייצוג של 13 בבינארי הוא 0b1101. עוד שימו לב לכך שהמבנה הטופולוגי של כל הערימות בגודל 13 הוא זהה. רק התוכן שלהן שונה. חיבור של שתי ערימות בינומיאליות נעשה בדרך שבה מחברים מספרים שלמים הכתובים בייצוג בינארי.

בפרולוג, נייצג **ערימה בינומיאלית** כרשימה של `bt/2` או אטומים `empty/0` **כאשר האיבר האחרון ברשימה הוא לא `empty`**.

ממשו פרדיקט `add_bt/3` אשר נכון אמ"מ הפרמטר הראשון הוא עץ בינומיאלי ושני הפרמטרים האחרים הם ערימות בינומיאליות כאשר הפרמטר השלישי הוא כמו הערימה בפרמטר השני, אך עם הוספת העץ הבינומיאלי.

ניתן להניח כי שני הפרמטרים הראשוניים הם קשורים. אין צורך לבדוק כי הפרמטר השני ערימה בינומיאלית.

דוגמה:

```
?- add_bt(bt(5, []), [bt(4, [])], X).  
X = [empty, bt(4, [bt(5, [])])].
```

סעיף ד'

ממשו פרדיקט `add/3` אשר נכון אמ"מ הפרמטר הראשון הוא מספר ושני הפרמטרים האחרים הם ערימות בינומיאליות כאשר הפרמטר השלישי הוא כמו הערימה בפרמטר השני, אך עם הוספת המספר.

ניתן להניח כי שני הפרמטרים הראשוניים הם קשורים. אין צורך לבדוק כי הפרמטר השני ערימה בינומיאלית.

דוגמה:

```
?- add(3, [bt(4, [])], X).  
X = [empty, bt(3, [bt(4, [])])].
```

סעיף ה'

ממשו פרדיקט `pop_min/3` אשר נכון אמ"מ הפרמטר הראשון הוא מספר ושני הפרמטרים האחרים הם ערימות בינומיאליות כאשר המספר הוא האיבר הקטן ביותר בערימה הראשונה והערימה השנייה היא כמו הערימה הראשונה שהוציאו ממנה את האיבר הקטן ביותר.

ניתן להניח כי הפרמטר השני קשור. אין צורך לבדוק כי הפרמטר השני ערימה בינומיאלית.

דוגמה:

```
?- fetch_min(X, [empty, bt(3, [bt(4, [])])], Y).  
X = 3,  
Y = [bt(4, [])].
```

סעיף ו'

הגענו לחלק המרתק לפי מילי ד"ר סדר! בסעיף זה אתם מתבקשים לממש מיון של מספרים שלמים [בעזרת ערימות בינומיליות](#). פתרון אשר משתמש בשיטות מיון אחרות לא יתקבל.

ממשו פרדיקט `sort_me/2` אשר נכון אמ"מ שני הפרמטרים הם רשימות של מספרים שלמים כאשר הרשימה השנייה היא הפרמוטציה הממוינת של הרשימה הראשונה. ניתן להניח כי הפרמטר הראשון קשור וכי אין מספרים חוזרים ברשימות.

דוגמה:

```
?- sort_me([5, -2, 10, 14, 1, 4, 0], X).  
X = [-2, 0, 1, 4, 5, 10, 14].
```


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!