שפות תכנות, 236319

פרופ' ד. לורנץ

אביב 2023

תרגיל בית 4

תאריך פרסום: 22.5.2023

מועד אחרון להגשה: 7.6.2023 בשעה 23:55.

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 אדמיניסטרטיביות פרטניות, על כותרת המייל להתחיל ב-"*236319 - תרגיל בית 4*"
- 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 בכתה וחלקן לא. כתבו את הפתרונות אליהן בקובץ **dry.pdf**.

שאלה 1

בשאלה זו נדון בייצוגים אפשריים של מספרים טבעיים (כולל אפס) ב-SML וב-Lisp.

1. נציע את הייצוג הבינארי הבא למספרים הטבעיים:
   - בייצוג יהיו שני סימבולים - "0" ו-"1".
   - המספר אפס מיוצג על ידי רשימה המכילה את "0" בלבד, המספר אחד מיוצג על ידי רשימה המכילה את "1" בלבד.
   - לכל n>0 - המספר 2n מיוצג על ידי הוספת "0" בראש הרשימה המייצגת את n, המספר 2n+1 מיוצג על ידי הוספת "1" בראש הרשימה המייצגת את n.

   נרצה לממש את הפונקציה add שמקבלת ייצוגים של המספרים n ו-m ומחזירה את הייצוג של n+m. האם ניתן לממש זאת בשפת Lisp ובשפת SML? אם ניתן - הסבירו בקצרה איך. אם לא ניתן, נמקו למה.

2. נציע את הייצוג האונארי הבא למספרים הטבעיים:
   - המספר אפס מיוצג על ידי רשימה ריקה.
   - לכל n>0 - המספר n מיוצג על ידי רשימה המכילה איבר בודד. האיבר הבודד הוא n-1 לפי אותו אופן ייצוג.

   נרצה לממש את הפונקציה add שמקבלת ייצוגים של המספרים n ו-m ומחזירה את הייצוג של n+m. האם ניתן לממש זאת בשפת Lisp ובשפת SML? אם ניתן - הסבירו בקצרה איך. אם לא ניתן, נמקו למה.

3. נייצג ב-SML את סדרת המספרים הטבעיים כסדרה של פונקציות. לפונקציה המייצגת את המספר n נקרא nat_n. הפונקציה nat_n מקבלת כפרמטר פונקציה f ומחזירה פונקציה שהיא הרכבה n פעמים של f:

$$nat\_n\ f = \overbrace{f \circ f \circ \cdots \circ f}^{n\ times}$$

למשל:

```
fun nat_0 _ = fn x => x;
fun nat_1 f = fn x => f x;
fun nat_2 f = fn x => f (f x);
fun nat_3 f = fn x => f (f (f x));
fun nat_4 f = fn x => f (f (f (f x)));
```

(א) הגדירו את הפונקציה nat2int שמקבלת פונקציה שמייצגת את המספר n ומחזירה את המספר n.

```
- nat2int nat_2;
val it = 2 : int
```

(ב) הגדירו את הפונקציה nat2str שמקבלת פונקציה שמייצגת את המספר n ומחזירה מחרוזת של כוכביות באורך n.

```
- nat2str nat_3;
val it = "***" : string
```

(ג) איזה ערך יוחזר מהקריאה הבאה? הסבירו בקצרה למה.

```
- nat2str (nat_2 nat_3);
```

(ד) הפונקציה foo מקבלת שתי פונקציות שמייצגות את המספרים n ו-m. איזה מספר מייצגת הפונקציה שהיא מחזירה? הסבירו בקצרה למה.

```
fun foo n m = fn f => fn x => m (n f) x;
```

(ה) הפונקציה bar מקבלת שתי פונקציות שמייצגות את המספרים n ו-m. איזה מספר מייצגת הפונקציה שהיא מחזירה? הסבירו בקצרה למה.

```
fun bar n m = fn f => fn x => m n f x;
```

(ו) השלימו את הגדרת הפונקציה succ המקבלת כפרמטר פונקציה המייצגת את המספר n ומחזירה פונקציה המייצגת את המספר העוקב n+1.

```
fun succ n = fn f => fn x => ___________________________
```

לדוגמה:

```
- nat2int (succ nat_2);
val it = 3 : int
```

(ז) השלימו את הגדרת הפונקציה add המקבלת שתי פונקציות המייצגות את המספרים n ו-m ומחזירה פונקציה המייצגת את n+m.

```
fun add n m = fn f => fn x => _________________________
```

לדוגמה:

```
- nat2str (add nat_2 nat_3);
val it = "*****" : string
```

שאלה 2

בהרצאה למדנו על העמסה בהקשרים שונים. בשאלה זו נחקור את שפת SML בתחום של העמסה. תענו על הסעיפים הבאים:

1. האם SML מאפשרת למתכנת להגדיר פונקציות מועמסות? אם כן, תנו דוגמה לפסקת קוד אשר מגדירה פונקציה מועמסת.

   הערה: פונקציות פולימורפיות ב-SML אינן פונקציות מועמסות.

2. האם ב-SML קיימת העמסה של מילה שמורה? אם כן, ציינו מהי ונמקו.
3. תנו דוגמה למזהה מעומס ב-SML. נמקו תשובתכם.
4. ידוע כי ב-SML קיים מנגנון אוטומטי לקביעת טיפוס פרמטר וערך החזרה של פונקציות. איך המנגנון מסתדר עם פונקציות מועמסות? האם שימוש בפונקציה מועמסת בתוך פונקציה אחרת יכול להפוך את האחרונה גם למועמסת?

שאלה 3

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

עברו על המודול typing בשפה Python (כל המידע הנחוץ לשאלה מופיע בעמוד הנתון).

1. ציינו את בנאי הטיפוסים התיאורטיים (type constructors) שיש להם מקביל במודול typing לפי הסמנטיקה שמתוארת בקישור המצורף.
2. לכל בנאי שציינתם בסעיף הקודם הביאו דוגמה קונקרטית בפייתון לשימוש בו.
3. האם קיים מקביל בקישור המצורף לטיפוס התיאורטי bottom? אם כן מהו?
4. האם קיים מקביל בקישור המצורף לטיפוס התיאורטי top? אם כן מהו?

חלק רטוב

- קבצים המסופקים לחלק זה תוכלו למצוא בגיטהאב של הקורס - Homework4.
- פתרון לכל אחת מהשאלות יש לכתוב בקובץ נפרד ששמו hw4_q<i>.sml עבור שאלה i. כמו כן, יש להוריד קובץ הגדרות לכל שאלה ולייבא אותו על ידי הוספת השורה הבאה בתחילת הפתרון:

```
use "hw4_q<i>_def.sml";
```

- לכל שאלה מסופקת לכם דוגמת קלט ופלט רצוי. לצורך העניין את הפתרון שלכם לשאלה 1 תוכלו לבדוק בעזרת הפקודות הבאות:

```
sml hw4_q1.sml < q1.in > q1.out
diff <(sed '1,/===TEST START===/d' q1.out) q1.expected
```

- יש לבדוק ולהריץ את פתרונותיכם לתרגיל זה בדוקר **twyair/safot-hw:4**
- מדריך התקנה ושימוש בדוקר

שאלה 1

עבר לא מעט זמן מהפעם הראשונה שהכרתם את ד"ר חיים מהפקולטה לביולוגיה. הוא נהנה מאוד לשתף פעולה איתכם בעבודה על המחקר שלו, אך הגיע הרגע לסיים את הסימולטור ולהגיד שלום אחד לשני.

בשאלה זו אתם תשתמשו בכל הפונקציות אשר מימשתם בשביל ד"ר חיים לאורך שני תרגילי הבית הקודמים. עליכם לייבא את קובץ ההגדרות hw4_q1_def.sml. קובץ ההגדרות מייבא גם את הקבצים הבאים:

hw2_q3.sml, hw3_q1.sml, hw3_q2.sml, hw3_q1_def.sml, hw3_q2_def.sml

כדי שתוכלו להריץ את הקוד אתם צריכים להעתיק את הקבצים הנ"ל מתרגילי הבית הקודמים לתיקיה בה נמצא קובץ ההרצה.

בנוסף, קובץ ההגדרות מכיל הגדרה של פונקציית עזר is_alive_bool:

```
is_alive_bool  : bool * bool * bool -> bool * bool * bool
-> bool * bool * bool -> bool
```

הפונקציה עוטפת את ה-is_alive אשר מימשתם בתרגיל בית 2. במקום לקבל ערכים מטיפוס Cell הפונקציה מקבלת ערכים מטיפוס bool, כאשר Empty שקול ל-false ו-alive שקול ל-true.

הערות:

- אין להגיש את הקבצים מתרגילי הבית הקודמים המצוינים לעיל.
- הבדיקה האוטומטית תשתמש בפתרון רשמי במקום הפתרונות שלכם מתרגילי הבית הקודמים על מנת למנוע הורדת נקודות על טעויות נגררות.

1. ממשו את הפונקציה **runCycle** אשר מקבלת מסגרת המתארת את מצב החיידקים במרחב ומחזירה את המסגרת אשר מתארת את מצב החיידקים אחרי איטרציה אחת לפי כללים של ד"ר חיים בתרגיל בית 2.

```
runCycle: frame -> frame
```

דוגמת הרצה:

```
- runCycle [" *** ", " * * ", " *** "];
val it = [" * * ","*   *"," * * "] : frame
```

2. ממשו את הפונקציה **gameOfLife** אשר מקבלת מסגרת התחלתית כלשהי המתארת את מצב החיידקים במרחב ומחזירה פונקציה. פונקצית הפלט מקבלת unit. עבור הקריאה הראשונה הפונקציה מדפיסה על המסך את המסגרת ההתחלתית ועבור כל קריאה הבאה מדפיסה על המסך את האיטרציה הבאה של הסימולטור.

```
gameOfLife: frame -> unit -> unit
```

דוגמת הרצה:

```
- val game = gameOfLife [" *** ", " * * ", " *** "];
val game = fn : unit -> unit
- game();
 ***
 * *
 ***
val it = () : unit
- game();
 * *
*   *
 * *
val it = () : unit
- game();

** **

val it = () : unit
```

משהו נחמד לסיום הסימולטור - אתם מוזמנים להריץ את הסימולטור על מסגרת התחלתית בעזרת הפונקציה run:

```
- run (gameOfLife start_frame) 100 0.5;
```

מימוש של run והמסגרת התחלתית אפשרית מסופקים לכם בקבצי התרגיל.

שאלה 2

בקובץ ההגדרות מסופקות לכם הגדרות של טיפוסים tree ו-union כפי שהוצגו בתרגולים. בשאלה זו עליכם לממש מספר פונקציות אשר פועלות על עצים בינארים ב**רקורסיית זנב**.

לאורך השאלה ניתן להשתמש בפונקציות עזר, אך יש להסתיר את המימוש שלהן. אסור להשתמש ב-ref.

ניתן להניח כי הפונקציה List.rev ממומשת ברקורסיית זנב. אין הנחה כזו לגבי שאר הפונקציות שלמדנו.

1. ממשו את הפונקציה **flatten** אשר מקבלת עץ בינארי ומחזירה preorder traversal שלו כרשימה.

```
flatten: 'a tree -> 'a list
```

דוגמת הרצה:

```
- flatten (Br(0, Br(1, Nil, Nil), Br(2, Nil, Nil)));
val it = [0, 1, 2] : int list
```

2. ממשו את הפונקציה **map** אשר מקבלת פונקציה f ועץ בינארי. הפונקציה **map** מחזירה עץ חדש בעל אותו מבנה כמו עץ הקלט שבכל צומת שלו מופיע תוצאת הפעלת f על הערך בצומת המתאים בעץ המקורי.

```
map: ('a -> 'b) -> 'a tree -> 'b tree
```

דוגמת הרצה:

```
- map (fn x => 2 * x) (Br(0, Br(1, Nil, Nil), Br(2, Nil, Nil)));
val it = Br(0, Br(2, Nil, Nil), Br(4, Nil, Nil)) : int tree
```

הערות:

- מומלץ, אך לא חובה, להשתמש בטיפוס union כדי לסמן באיזה מצב נמצאים איברים ברשימת עזר.

שאלה 3

בקובץ ההגדרות מסופקת לכם הגדרה של טיפוס seq כפי שהוצג בתרגולים. כמו כן, מסופקת לכם פונקציה counter אשר מחזירה סדרה של מספרים טבעיים. כל פעם שמחשבים את המשך הסדרה מודפסת הודעה מתאימה.

בשאלה זו עליכם לממש רצף דו-כיווני biseq. רצף דו-כיווני "עוטף" רצף רגיל ומאפשר מעבר דו-כיווני עליו (אחורה וקדימה). על מנת לחסוך בחישובים הרצף דו-כיווני שומר כל איבר ברצף שחושב כבר. המיקום הנוכחי ברצף דו-כיווני נקרא האינדקס שלו. האינדקס הראשון הוא 0.

1. הגדירו את הטיפוס **biseq** ע"י datatype או type וחריגה **SeqErr**.

2. ממשו את הפונקציה **new** אשר מקבלת רצף רגיל ומחזירה רצף דו כיווני המתאים לו.

```
new: 'a seq -> 'a biseq
```

3. ממשו את הפונקציה **curr** אשר מקבלת רצף דו כיווני ומחזירה את האיבר הנוכחי (באינדקס המתאים). יש לזרוק חריגה SeqErr אם לא נותרו איברים ברצף.

```
curr: 'a biseq -> 'a
```

4. ממשו את הפונקציה **empty** אשר מקבלת רצף דו כיווני ומחזירה true אם לא נותרו איברים ברצף (הגענו לקצה הימני ביותר של הרצף).

```
empty: 'a biseq -> bool
```

5. ממשו את הפונקציה **next** אשר מקבלת רצף דו כיווני ומחזירה את אותו רצף, אך עם אינדקס גדול מ-1. יש לזרוק חריגה SeqErr אם לא נותרו איברים ברצף.

```
next: 'a biseq -> 'a biseq
```

6. ממשו את הפונקציה **prev** אשר מקבלת רצף דו כיווני ומחזירה את אותו רצף, אך עם אינדקס קטן מ-1. יש לזרוק חריגה SeqErr אם האינדקס שווה ל-0.

```
next: 'a biseq -> 'a biseq
```

דוגמת הרצה המכילה **רק** את ההדפסות של counter:

```
- new (counter ());
- next it;
exec: 1
- next it;
exec: 2
- prev it;
- next it;
- next it;
exec: 3
```

הנחיות הגשה

- את הפתרון לתרגיל הבית יש להגיש בקובץ zip המכיל את הקבצים הבאים בלבד:
hw4_q1.sml, hw4_q2.sml, hw4_q3.sml, dry.pdf.

- על החלק היבש להיות מוקלד. אין להגיש סריקה או צילום של התשובות לחלק זה.
- שם קובץ ההגשה יהיה EX4_ID1_ID2.zip כאשר ID1, ID2 הם מספרי ת.ז. של המגישים.
- מסופק לכם סקריפט לבדיקת עצמית של קובץ הגשה בשם **selfcheck.sh**. אנא בדקו את תקינות ההגשה שלכם בעזרת הסקריפט בסביבת ה-docker. הקובץ עודכן ב-31.05.
- 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!