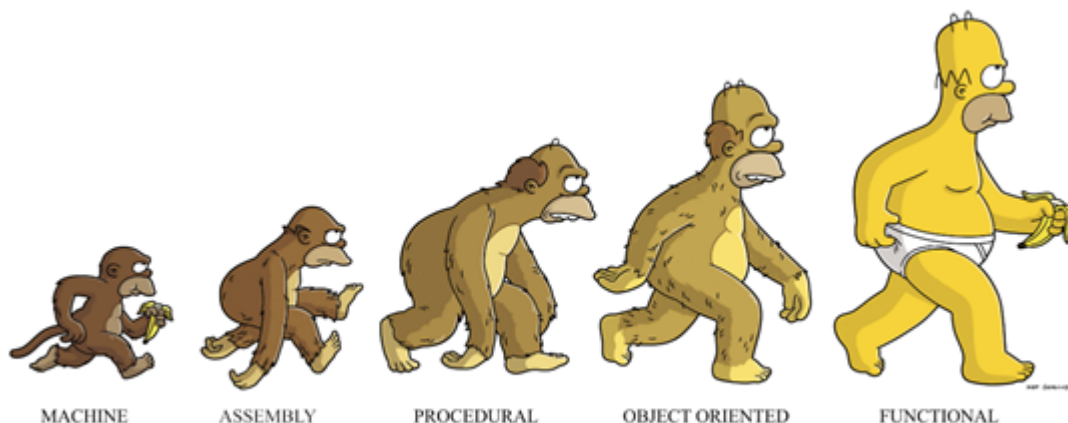


שפות תכנות, 234319

חורף תשע"ח



תרגיל

בית מס' 2 – טיפוסים ובנאי טיפוסים, פונקציות ב-ML

- תאריך פרסום: 14-11-2017
- מועד אחרון להגשה: 28-11-2017
- מתרגל אחראי: ליאור בן ימין
- אי-מייל: lior.b@cs.technion.ac.il
- בפניה בדוא"ל, נושא ההודעה (subject) יהיה "2PLW17-EX" (ללא המירכאות)

תרגיל בית זה מורכב משני חלקים, חלק יבש וחלק רטוב. לפני ההגשה, ודאו שההגשה שלכם תואמת את הנחיות ההגשה.

חלק יבש

1. קראו את [סיכום הרצאה 1 + 2](#). מהי שפה אוטרקית?
שפת תכנות היא אוטרקית, אם כל תכנית שכתובה בה מגדירה הן את נקודת ההתחלה, והן את תחום ההתפרשות של הקוד. כלומר, ניתן לקבוע מקריאת תכנית שכתובה בשפה זו, היכן התכנית תתחיל להתבצע ואיזה קוד בדיוק כלול בתכנית.
קלט/פלט, שגרות עזר וכו': מהווים חלק מהגדרת השפה או באמצעות מזהים מוגדרים מראש או מילים שמורות.
טיפוסים יסודיים: בשפה, מילים שמורות, מילים מוגדרות מראש.
כמו כן, בשפה זו הקוד רשום בקובץ אחד בלבד.

2. קראו את הסיכום בנושא [שמות, קישורים, וישיבות בנות-שיום](#). מה ההגדרה של המונח nameable?
מה התרגום לעברית או לערבית או לרוסית של המונח nameable?
Nameable הינו דבר שניתן לתת לו שם.
בשפות תכנות, ישויות בנות שיום (NAMEABLES) הם כאלו שהמתכנת יכול לתת להם שם.
nameable - הוא שיום.

3. קראו את המכתב למערכת העיתון Communications of the ACM שכתב Donald Knuth בשנת 1964 בנושא [Backus Normal Form vs. Backus Naur Form](#)¹. מה ההבדל, אם יש הבדל, בין צורה קנונית (canonical form) וצורה נורמלית (normal form)? מה ההבדל, אם יש הבדל, בין Backus Naur Form ובין Backus Normal Form?

צורה קנונית הינה צורה להציג או לתאר משהו, בצורה חד משמעית שלא יכולה להתפרש בצורה שגויה, ובצורה יחידה(חד-חד ערכית).
צורה נורמלית הינה דרך להציג בצורה מוסכמת תיאור לתחביר(למשל...), לעומת צורה קנונית, בצורה נורמלית לאותו הדבר יכולים להיות מספר תיאורים שונים המתארים אותו דבר.

אין הבדל בין Backus Naur Form לבין Backus Normal Form. בהתחלה רצו לקרוא לשיטה הזאת Backus Normal Form אבל בזכות תרומתו של Naur הוחלט לשנות זאת ל Backus Naur Form, כלומר BNF.

4. ציינו כמה שגיאות בשקפים של פרק 2.4, אם ישנן כאלה, והציעו תיקון.

$$e = (e')^* \Rightarrow S(e) = \bigcup_{i=0}^{\infty} S(\overbrace{e' \dots e'}^{i \text{ times}})$$

בשקופית 9,

לא קיימת הגדרה של הפעלת כוכב על e וזה סינטקטית לא קיים
בנוסף, בשקופית 10 יש טעות כתיב Range Range.

¹ ניתן לגשת למאמרים בתשלום דרך [אתר ספריית הטכניון](#).
הנה [לינק ישיר](#).

5. מה מציין הסמל ε ? ציינו שלוש משמעויות בשלושה הקשרים שונים.

ε , כלומר אפסילון היא האות החמישית באלפבית היווני. בספרות יווניות היא מסמלת את המספר 5 וזהו גם שווה.

- אפסילון במדעי המחשב, הוא "המילה הריקה" בביטויים רגולריים.
- בחשבון אינפיניטסימלי מקובל לסמל באפסילון את רדיוס הסביבה של נקודה כלשהי, שיכול להיות "מספר ממשי קטן כרצוננו", ומשמש לדוגמה בהגדרת מושגי הגבול והרציפות.
- באסטרונומיה, אפסילון הוא שמה של הטבעת הגדולה ביותר של אורנוס.

6. מנו את כל האפלויות שנהוגות בבנאי הטיפוסים בפסקל.

- ניתן ליצור קבוצות חזקה של טיפוסים פרימיטיביים בלבד (המכונים, Base types), לא ניתן ליצור קבוצות חזקה של מערכים או Records למשל.
- בדומה, ניתן ליצור קבוצות (Sets) של הטיפוסים Integer, Real, Char, Alfa בלבד.
- ניתן לבנות Subrange על בסיס הטיפוסים Integer, Real, Char, על בסיס Subrange גדול ממנו או על בסיס קבוצה (ואז ניתן לבנות את הטווח על בסיס Alfa, במידה ואלה מסודרים בקבוצה).
- הערכים של מערכים יכולים להיות מטיפוס כלשהו, אך המפתחות שלהם יכולים להיות טיפוסים פרימיטיביים למעט Alfa, קבוצה (ואז ניתן להשתמש ב Alfa או Subrange בלבד).

7. מהם ה-nameables של שפת BATCH של DOS?

הישויות בנות הישום של שפת BATCH בדוס הם:

- משתנים: שם של משתנה יכול להכיל רווחים, אותיות קטנות וגדולות, מספרים והתוויות הבאות:
~{}`[]@?.,+*()'\$_#
- תגיות: שם של תגית מוגדר בשורה בפני עצמה, מתחיל עם נקודתיים ומסתיים עם רווח, נקודתיים או שורה חדשה. (CR/LF) יש לשים לב שפקודות כמו GOTO מתייחסות רק ל-8 התווים הראשונים בשם של התגית.

8. כתבו או מצאו באינטרנט תוכנית בפסקל שמדפיסה את עצמה. הסבירו את התוכנית. מה ה"טריק" שמאפשר זאת?

```
program Fixedpoint(output); const A='program Fixedpoint(output); const A='; B='; begin
writeln(A,chr(39),A,chr(39),chr(59),chr(32),chr(66),chr(61),chr(39),B,chr(39),B) end.';
begin writeln(A,chr(39),A,chr(39),chr(59),chr(32),chr(66),chr(61),chr(39),B,chr(39),B)
end.
```

ה"טריק", הוא האופן בו אנחנו מתארים מחרוזת. בתוך המחרוזת אנו יכולים לרשום למשל chr(61) שמתפרש בתור התו שווה. וכן הלאה עד לקבלת הקוד התוכנית ה"מקורי".

9. עיינו בתיעוד שפת התכנות Kotlin:

a. הסבירו את המושגים frontend ו-backend של מהדר (קומפיילר)? מנו 3 backend-ים של המהדר של kotlin.

Frontend – זהו החלק "הקדמי" במהדר העוסק בשפה עצמה, כלומר בסריקת הטוקנים (tokens), parsing ויצירת עץ הפעולות (parse-tree). חלק זה יכול להיות גנרי לכל מכונה/סביבת עבודה.
Backend – זהו החלק "האחורי" במהדר העוסק בעבודה מול המכונה, כלומר בקוד המכונה, קידוד לפורמט המכונה עלייה תרוץ התוכנית. כל חלק כזה יכול לעבוד (בדרך כלל) רק על מכונה אחת כי הוא מותאם אליה.

באתר שפת kotlin נאמר שהשפה תומכת בריצה על:

- android כלומר טלפונים ניידים
- browser(js) כלומר ריצה על צד לקוח (pc)
- JVM כלומר מעבר ל-JAVA.

b. באילו מגנוני הגדרות של שפות תכנות משתמשת Kotlin, ובאיזה אופן? המגנונים להגדרת השפה הם:

- EBNF כדי להגדיר את סינטקס השפה (הגדרה סינטקטית)
- Language Documentation של השפה לטובת הסבר על התכונות/אפשרויות הקיימות. (הגדרה סמנטית)
- Compiler

c. האם השפה אוטרקית? באיזה אופן נקבעים גבולות התכנית? באיזה אופן נקבעת נקודת תחילת הביצוע של התכנית?

kotlin אינה שפה אוטרקית משום שתוכנית שהיא יכולה להיות ביותר בקובץ יחיד.
גבולות התוכנית נקבעים ע"י כל הקבצים בהם נעשה שימוש ישיר או עקיף, כלומר אלה שהתוכנית משתמשת בהם (packages) או כאלה שpackages משתמשים בpackages אחרים.
נקודת תחילת ביצוע התוכנית נקבעת בצורה דומה ל-C בפונקציה עם שם מוגדר מראש, kotlin זו פונקציית main.

d. מהם סוגי הישויות להם ניתן לתת שמות (nameable) בשפה? ניתן לתת שמות למשתנים, טיפוסים, פונקציות, מחלקות, אובייקטים.

e. האם אפשר להשתמש בשפה באותו שם לישויות מסוגים שונים? אם כן, תנו דוגמה.
 בדוגמא ניתן לראות כי למשתנה ולפונקציה יש אותו משתנה, למשתנה a ולפונקציה a.

```
fun main(args: Array<String>) {
    var a = 3

    var b = a(a)
    println(b)
}
fun a(x: Int) = x*x
```

9

f. מהם בנאי הטיפוסים בשפה?

בנאי הטיפוסים בשפה הם:

- פונקציות,
- מערכים,
- Closures-
- classes-
- list-
- set-
- map-
- Enumerations-
- Enum Classes-

לבדוק שכל המצגות
 תקינות לפני
 תחילת הסמסטר



לתקן מצגות
 לילה לפני
 ההרצאה



לגלות טעויות
 בזמן ההרצאה
 ולתקן אותן בלייב



לבקש מסטודנטים
 לתקן
 את המצגות



חלק רטוב - ML

הנחיה כללית: בתרגול 4 נלמדים המבנים let/local שנועדו להסתיר פונקציות עזר. החל מהתרגיל הזה ועד סוף הקורס, אתם רשאים להגדיר ולהשתמש בפונקציות עזר בתרגילי ה-ML כרצונכם (אלא אם מצוין במפורש אחרת), אך עליכם להסתיר אותם. בחלק של [הנחיות ההגשה](#) תוכלו למצוא את רשימת החתימות שקובץ הפתרון שלכם צריך להניב. מערכת הבדיקות האוטומטית מוודאת שאין חתימות נוספות פרט לרשימה הזו, ואם תופענה חתימות אחרות, ירדו לכם 5 נקודות. הקפידו לבדוק את החתימות שטעינת הקובץ מניבה.

משימה 1

1. כתבו את הפונקציה dubchar שחתימתה היא char -> string, המקבלת תו c, ומחזירה את המחרוזת המורכבת משני מופעים של התו c. לדוגמא:

```
- dubchar #"$";  
val it = "$$": string
```

2. כתבו את הפונקציה apply_on_nth_char שחתימתה היא:

```
(char -> 'a') -> int -> string -> 'a
```

המקבלת פונקציה f ומספר שלם n (באמצעות currying), ומחזירה פונקציה חדשה המבצעת את הפעולה הבאה: בהינתן מחרוזת s, הפונקציה תחזיר את הפלט של ההפעלה של f על התו ה-n-י ב-s. אם אורכה של s קטן מ-n+1 או ש-n שלילי, הפונקציה תחזיר את התוצאה של ההפעלה של f על התו "#!".
רמז: האופרטור <code>orelse</code> שימושי בתרגיל הזה.

דוגמאות הרצה:

```
- apply_on_nth_char ord 2;  
val it = fn : string -> int  
- it "ABC";  
val it = 67 : int (* ord #"C" = 67 *)
```

```
- apply_on_nth_char dubchar 7;  
val it = fn : string -> string  
- it "ABC";  
val it = "!!" : string (* dubchar #"!" = "!!" *)
```

משימה 2

מספר מושלם (משוכלל) הוא מספר טבעי השווה לסכום כל המחלקים הטבעיים שלו מלבד המספר עצמו. כתבו פונקציה perfect אשר מקבלת מספר טבעי ובודקת האם הוא מושלם.

```
perfect = fn: int -> bool
```

דוגמאות הרצה:

```
perfect 6;  
val it = true : bool  
- perfect 496;
```

```

val it = true : bool b
- perfect 497;
val it = false : bool
- perfect 8130;
val it = false : bool

```

משימה 3

כתבו את הפונקציה `balance` אשר בודקת איזון של סוגריים במשפט.

```

balance = fn : string -> bool

```

הפונקציה תקבל מחרוזת ותבדוק שהסוגריים במחרוזת מאוזנים, משמע יש מספר זהה של '(' ו') וגם לא ייתכן מצב בו ברישא של המחרוזת יופיע יותר סוגרים "סוגרים" מסוגרים "פותחים".

דוגמאות הרצה:

```

- balance “()”;
val it = true : bool
- balance “if(true) then (foo(5))”;
val it = true : bool

```

```

- balance “:”;
val it = false : bool

```

```

- balance “(”);
val it = false : bool

```

משימה 4

כתוב פונקציות בשמות `sig1, sig2, ..., sign` אשר הפונקציה `sig1` מתאימה לסעיף `i`. למשל, הפונקציה `sig2` תשובתך לאתגר בסעיף 2.

הבהרה - אסור להשתמש ב-`type constraints`! (כלומר דרישה בחתימה על טיפוס החזרה או טיפוס הפרמטר של הפונקציה).

אין חשיבות למה שהפונקציה מבצעת. כל שעליך להבטיח הוא כי חתימת הפונקציה תהיה כבסעיף:

1. `'a->'b->('a * 'b ->'b)->'b`
2. `int * real -> (real -> string) -> bool`
3. `('a -> 'b -> 'c) -> 'a -> 'b -> 'd -> 'c`
4. `'a -> 'b -> int -> int -> int`
5. `('a -> 'b) -> 'a -> ('b * 'b -> 'c) -> 'c`
6. `unit -> unit -> int`

הנחיות הגשה

- בתרגיל זה ניתן להשתמש רק בחומר שנלמד בשפת ML עד (כולל) **תרגול 4**. אין להשתמש באף פונקציה או תכונה של השפה שלא נלמדה בתרגולים.
- שימו לב כי ML היא case-sensitive, ודאו על כן כי שמות הפונקציות הן בדיוק כפי שנדרש בתרגיל.
- ההגשה האלקטרונית תתבצע באתר הקורס ותכיל את הקבצים הבאים:

- הקובץ `dry.pdf` הכולל את התשובות לחלק היבש של התרגיל.
- הקובץ `ex2.sml` הכולל את כל הפונקציות שנדרשתם לממש בחלק הרטוב
 - רשימת הפונקציות שטעינת הקובץ `ex2.sml` צריכה להניב:

```
perfect ■  
dubchar ■  
apply_on_nth_char ■  
balance ■  
sig1 ■  
sig2 ■  
sig3 ■  
sig4 ■  
sig5 ■  
sig6 ■
```

לכל קובץ שמכיל קוד:

- הוסיפו בשורה הראשונה הערה המכילה את השם, מספר ת.ז. וכתובת המייל של המגישים מופרדים באמצעות רווח.

- וודאו שהקבצים נטענים/מתקמפלים ללא שגיאות גם לאחר הוספת ההערה שלעיל.

יש להגיש את הקבצים דחוסים בתוך קובץ zip. הקבצים יהיו בשורש קובץ ה-zip ולא בתוך ספרייה. שם הקובץ יהיה EX2_ID1_ID2.zip כאשר EX2_ID1_ID2 הם מספרי ת.ז. של המגישים. שימו לב שהבדיקה של החלק הרטוב היא אוטומטית, ולכן הקפידו על מילוי כל ההוראות בשביל למנוע בעיות מיותרות.

- על החלק היבש להיות מוקלד, אין להגיש סריקה או צילום של התשובות לחלק זה.
- אין צורך להגיש ניירת הסמסטר. תא הקורס לא יבדק במהלך הסמסטר, אז אנא חסכו בנייר.
- בודקי התרגילים מאוד אוהבים Memes. שתפו את תחושותיכם במהלך פתירת התרגיל באמצעות Meme מתאים על דף השער בהגשה - אולי יצא מזה משהו מעניין!

תיקונים והבהרות

- [חלק רטוב משימה 4](#) - אסור להשתמש בtype constraints! (כלומר דרישה בחתימה על טיפוס החזרה או טיפוס הפרמטר של הפונקציה).
- [חלק יבש שאלה 3](#) - נוסף קישור למאמר לקריאה המכווין לתשובה, וגם הוראות גישה למאמרים בתשלום מחוץ לרשת הטכניונית.