שפות תכנות, 236319

פרופ' ד. לורנץ

אביב 2023

תרגיל בית 2

תאריך פרסום: 18.4.2023

מועד אחרון להגשה: 4.5.2023 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 2*"
- 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

לאחר פשיטת רגל, מנהלי חברת 'החישובים הלא מיותרים בע"מ' החליטו להקים חברה חדשה בשם 'הדקדוקים הלא מיותרים'. הם הבינו כי התוצר הקודם שלהם לא תאם לזמנו והחליטו ליצור דקדוק חדש שימצא חן בעיניו של לקוח מודרני.

לפניכם דקדוק של EmojiLang הנתון כ-EBNF:

```
<statements> = <statement> <statements> | <statement>
<statement> = ✍️<variable>⬅️<expression>
<statement> = 📖<expression>
<expression> = ↪️<variable>
<expression> = <term>
<expression> = <expression> <operation> <expression>
<variable> = 📗|📘|📙|📕|📔
<term> = 😇|🙂|😐|🙁|☹️|☹️|😣
<operation> = ➕|➖|✖️|➗
```

1. רשמו את רשימת הטרמינלים, הלא טרמינלים וסימבול התחלתי של הדקדוק.
2. עבור כל אחת מסדרות הטרמינלים הבאות קבעו האם היא שייכת לדקדוק של EmojiLang. אם כן, פרטו את כללי הגזירה המובילים אליה. אם לא, הסברו בקצרה למה.

הערה:

צריך להתייחס לכל סעיף בתור סדרת טרמינלים **יחידה** ללא התייחסות לירידת שורה (היא לא שייכת לסדרה, אלא רק מקלה על הקריאות).

a.

😐➕😇✖️↪️📘

b.

📖↪️📔
✍️📘⬅️😐
✍️📙⬅️📘
📖↪️📙

c.

✍️📕⬅️🙁
📖😣✖️↪️📕➖😇
✍️📗⬅️🙂➖🙂

d.

✍️📔⬅️😡
📖↪️📔➗↪️📔
📖↪️📔➗↪️📔

3. האם הדקדוק מכיל ambiguities? אם כן, תנו דוגמה לסדרת הטרמינלים שמראה את ה-ambiguity. אם לא, הסבירו למה.

שאלה 2

עבור כל אחת מהקריאות הבאות הסבירו בקצרה למה היא מחזירה שגיאה ב-SML:

```
1.  "h" ^ #"C";
2.  84 / 2;
3.  fun f x = if x = 0 then x else false;
4.  fun f x = if x = #"a" then x ^ "b" else x ^ "c";
5.  5 - (-3);
6.  Math.sqrt 9;
7.  sin 3.14;
8.  val if = 5;
9.  String.sub ("hello", 5);
10. fun sqrt_of_int x:int = Math.sqrt (real x);
```

חלק רטוב

- קבצים המסופקים לחלק זה תוכלו למצוא בגיטהאב של הקורס - Homework2.

  שימו לב כי קבצי out ו-expected עודכנו ב-29.04.2023.

- פתרון לכל אחת מהשאלות יש לכתוב בקובץ נפרד ששמו hw2_q<i>.sml עבור שאלה i. בנוסף לכל שאלה מסופקת לכם דוגמת קלט ופלט רצוי. לצורך העניין את הפתרון שלכם לשאלה 1 תוכלו לבדוק בעזרת הפקודות הבאות:

```
sml hw2_q1.sml < q1.in | sed '1,/===TEST START===/d' >
q1.out
diff q1.out q1.expected
```

- יש לבדוק ולהריץ את פתרונותיכם לתרגיל זה בדוקר **twyair/safot-hw:2**
- מדריך התקנה ושימוש בדוקר

שאלה 1

כתבו פונקציות בשמות sig1,sig2,...,sign כאשר הפונקציה sigi מתאימה לסעיף i. למשל, הפונקציה sig4 תהיה תשובתך לאתגר בסעיף 4.

הבהרה - **אסור** להשתמש ב-pattern matching ו-type constraints (כלומר דרישה בחתימה על טיפוס החזרה או טיפוס הפרמטר של הפונקציה).

אין חשיבות למה שהפונקציה מבצעת. כל שעליכם להבטיח הוא כי חתימת הפונקציה תהיה כנדרש:

```
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
7.'a -> 'a * 'a -> 'a
8.int * string * string -> int * string * string
```

שאלה 2

ממשו את הפונקציות:

```
curry: ('a * 'b -> 'c) -> 'a -> 'b -> 'c
uncurry: ('a -> 'b -> 'c) -> 'a * 'b -> 'c
```

כפי שראינו בתרגולים, כל פונקציה ב-SML מקבלת ארגומנט יחיד ומחזירה ערך יחיד.
למרות זאת, יש ב-SML שתי דרכים לדמות פונקציה המקבלת שני פרמטרים:

- Uncurried - פונקציה המקבלת tuple של שני איברים ומבצעת את הפעולה הנדרשת (ניתן להגיד שבכל זאת הפונקציה מקבלת פרמטר אחד מסוג tuple).
- Curried - פונקציה המקבלת את הארגומנט הראשון ומחזירה פונקציה שניה המקבלת את הארגומנט השני ומבצעת את הפעולה הנדרשת על שני הארגומנטים.

לכל אחת מהאפשרויות יש יתרונות וחסרונות משלה וניתן לבחור באחת האפשרויות ע"פ הצורך. כמו כן, ניתן להמיר פונקציה שהיא uncurried להיות curried ולהפך. בשאלה זו עליכם להגדיר שתי פונקציות המבצעות המרה מהאפשרות הראשונה לשנייה ולהפך.

הפונקציה curry ממירה פונקציה מ-uncurried ל-curried.

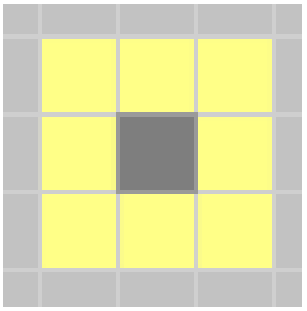
הפונקציה uncurry ממירה פונקציה מ-curried ל-uncurried.

דוגמאות הרצה:

```
- curry op*;
val it = fn : int -> int -> int
- val a = it 3 4;
val a = 12 : int
- uncurry it;
val it = fn : int * int -> int
- val a = it (3,4);
val a = 12 : int
```

שאלה 3

ד"ר חיים מהפקולטה לביולוגיה מבקש עזרתכם במחקר שלו על שכפול חיידקים במצב אוכלוסיות יתר. הוא רוצה לסמלץ את תהליך החיים של מושבת חיידקים במרחב קרטזי דו-מימדי. כל תא במרחב יכול להכיל חיידק או להיות ריק. לכל תא יש 8 שכנים כפי שניתן לראות בציור:



ד"ר חיים דרך ניסויים רבים הצליח להבחין כללים מסוימים שלפיהם ניתן לקבוע האם באיטרציה הבאה התא יהיה מאוכלס בחיידק או ריק על סמך המידע על האיטרציה הנוכחית. לפניכם הכללים:

- אם בסוף האיטרציה הקודמת התא מאוכלס בחיידק:
  - אם לתא היה לכל היותר שכן אחד מאוכלס בחיידק אזי החיידק בתא מת מבדידות.
  - אם לתא היו 4 שכנים מאוכלסים או יותר אזי החיידק בתא מת מאכלוס יתר.
  - חיידק בתא עם שניים או שלושה שכנים מאוכלסים בדיוק שורד את האיטרציה.
- אם בסוף האיטרציה הקודמת התא ריק ויש לו בדיוק שלושה שכנים מאוכלסים אזי התא יאוכלס בחיידק חדש.

ממשו את הפונקציה:

```
is_alive: cell * cell * cell-> cell * cell * cell-> cell *
cell * cell-> cell
```

אשר מקבלת שלושה טאפלים של ערכים מטיפוס חדש, כאשר כל ערך מייצג מצב התא בתבנית 3X3.

הפונקציה מחזירה האם התא המיוצג ע"י הערך השני בטאפל השני (באמצע של התבנית) יהיה מאוכלס בחיידק באיטרציה הבא כאשר שאר הערכים מציינים את השכנים של התא.

כמו כן, עליכם לספק את ההגדרה של הטיפוס החדש. את שמו וערכיו תוכלו להסיק מדוגמאות ההרצה המסופקות.

הערות:

- ניתן להשתמש בפונקציות עזר, אך יש להסתיר מימושיהן בעזרת local או let.
- על הטיפוס cell להיות מוגדר כטיפוס חדש, ולא כ-alias לטיפוס קיים.
- לקריאה נוספת (לא חובה) - Conway's Game of Life.

דוגמת הרצה:

```
- is_alive (alive, empty, empty) (empty, alive, alive)
(empty, empty, empty);
val it = alive: cell
- is_alive (alive, alive, alive) (empty, alive, empty)
(alive, alive, alive);
val it = empty: cell
```

הנחיות הגשה

- את הפתרון לתרגיל הבית יש להגיש בקובץ zip המכיל את הקבצים הבאים בלבד:
hw2_q1.sml, hw2_q2.sml, hw2_q3.sml, dry.pdf.

- על החלק היבש להיות מוקלד. אין להגיש סריקה או צילום של התשובות לחלק זה.
- שם קובץ ההגשה יהיה EX2_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!