ה א ו נ י ב ר ס י ט ה ה פ ת ו ח ה

20905

שפות תכנות

חוברת הקורס- אביב 2019ב

כתב: דני כלפון

פברואר 2019 - סמסטר אביב – תשע"ט

פנימי – לא להפצה.

© כל הזכויות שמורות לאוניברסיטה הפתוחה.

תוכן העניינים

אל הסטודנט א

1. לוח זמנים ופעילויות ג

2. תיאור המטלות ה

2.1 מידע כללי ה

2.2 מבנה המטלות ה

3. התנאים לקבלת נקודות זכות ו

ממ"ן 11 7

ממ"ן 12 9

ממ"ן 13 13

ממ"ן 14 15

ממ"ן 15 19

ממ"ן 16 23

אל הסטודנט,

אנו מקדמים את פניך בברכה עם הצטרפותך אל הלומדים בקורס "שפות תכנות". בחוברת זו תמצא לוח הזמנים של הקורס, מטלות ותנאים לקבלת נקודות זכות בקורס.

לקורס קיים אתר באינטרנט בו תמצאו חומרי למידה נוספים, אותם מפרסם/מת מרכז/ת ההוראה. בנוסף, האתר מהווה עבורכם ערוץ תקשורת עם צוות ההוראה ועם סטודנטים אחרים בקורס. פרטים על למידה מתוקשבת ואתר הקורס, תמצאו באתר שה"ם בכתובת:

http://telem.openu.ac.il

מידע על שירותי ספרייה ומקורות מידע שהאוניברסיטה מעמידה לרשותכם, תמצאו באתר הספריה באינטרנט www.openu.ac.il/Library.

שעות הייעוץ הטלפוני שלי יפורסמו סמוך לפתיחת הסמסטר באתר הקורס. אפשר ורצוי לפנות אלי בדואר אלקטרוני: dannyca@openu.ac.il, תוך ציון שם מלא, מספר תעודת זהות ומספר טלפון. פגישות חובה לתאם מראש. לצורך בירורים בנושאים אדמיניסטרטיביים יש לפנות בכתב או טלפונית למחלקות האוניברסיטה הפתוחה.

הקורס הוא תכנותי באופיו, וכולל מטלות תכנותיות החשובות להבנת החומר ותרגולו. מומלץ שתקדישו זמן ראוי ללמידת חומר הקורס, שכן זה קורס השונה באופיו מקורסים תכנותיים אחרים המוכרים לכם באו"פ. הקורס כולל פיתוח מפרשים לשפות תכנות פשוטות המדגימות עקרונות הקיימים בשפות תכנות מודרניות. השתתפות במפגשים, הקדשת זמן ראוי ללמידת החומר והגשת המטלות הם הדרך הנכונה לסיום הקורס בהצלחה.

בברכת לימוד מהנה

כלפון דני

מרכז ההוראה בקורס

1.לוח זמנים ופעילויות (20905 / ב2019)

שבוע לימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח ממ״ן (למנחה)
1	1.3.2019-24.2.2019	1	מפגש ראשון	
2	8.3.2019-3.3.2019	1		
3	15.3.2019-10.3.2019	2	מפגש שני	ממ״ן 11 15.3.2019
4	22.3.2019-17.3.2019 (ה-ו פורים)	2		
5	29.3.2019-24.3.2019	2	מפגש שלישי	
6	5.4.2019-31.3.2019	3		ממ״ן 12 5.4.2019
7	12.4.2019-7.4.2019	3	מפגש רביעי	
8	19.4.2019-14.4.2019 (ו ערב פסח)	3		
9	26.4.2019-21.4.2019 (א-ו פסח)	3	מפגש חמישי	

** התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב״לוח מפגשים ומנחים״.*

לוח זמנים ופעילויות - המשך

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
10	3.5.2019-28.4.2019 (ה יום הזכרון לשואה)	3		ממ"ן 13 3.5.2019
11	10.5.2019-5.5.2019 (ד יום הזיכרון) (ה יום העצמאות)	4	מפגש שישי	
12	17.5.2019-12.5.2019	4		ממ"ן 14 17.5.2019
13	24.5.2019-19.5.2019 (ה ל"ג בעומר)	4	מפגש שביעי	
14	31.5.2019-26.5.2019	7		ממ"ן 15 31.5.2019
15	7.6.2019-2.6.2019	7	מפגש שמיני	
16	14.6.2019-9.6.2019 (א שבועות)	7		ממ"ן 16 14.6.2019

מועדי בחינות הגמר יפורסמו בנפרד

** התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".*

2. תיאור המטלות

לתשומת לבכם!

כדי לעודדכם להגיש לבדיקה מספר רב של מטלות הנהגנו את ההקלה שלהלן:

אם הגשתם מטלות מעל למשקל המינימלי הנדרש בקורס, **המטלות** בציון הנמוך ביותר, שציוניהן נמוכים מציון הבחינה (**עד שתי מטלות**), לא יילקחו בחשבון בעת שקלול הציון הסופי.

זאת בתנאי שמטלות אלה **אינן חלק מדרישות החובה בקורס** ושהמשקל הצבור של המטלות האחרות שהוגשו, מגיע למינימום הנדרש.

זכרו! ציון סופי מחושב רק לסטודנטים שעברו את בחינת הגמר בציון 60 ומעלה והגישו מטלות כנדרש באותו קורס.

שימו לב!

בקורס זה **חובה** להגיש את המטלות בזמן, בהתאם לתאריך ההגשה המצוין עליהן. במקרים חריגים, כאשר יש סיבה מוצדקת להגשת המטלה באיחור, יש לפנות **בכתב** בדואר אלקטרוני אל מרכז ההוראה בקורס. **את הבקשה יש להגיש מראש!** יש לצרף לבקשה אישורים רשמיים, להצדקת סיבת הבקשה.

מטלות שיוגשו באיחור ללא אישור יבדקו והציון שיוזן עבורן יהיה 0, ללא תלות בציון של הבדיקה.

שימו לב, טיפול בבקשות שנשלחות לאחר מועד ב׳ של הסמסטר אינו בסמכות מרכז ההוראה, ויש להפנותן את האחראית על פניות סטודנטים של החטיבה למדעי המחשב.

להלן פירוט הניקוד לכל מטלה:

ממ״ן	ניקוד
11	5
12	5
13	5
14	5
15	5
16	5

2.2 מבנה המטלות

כל מטלה מורכבת מכמה שאלות. בראש כל שאלה מצוין משקלה היחסי בקביעת ציון המטלה. אם השאלה בממ״ן אינה ברורה לך, אל תהסס להתקשר אל אחד המנחים (בשעות הייעוץ הטלפוני שלו) לצורך קבלת הסבר, או הפנה את שאלתך אל פורום הדיונים של הקורס.

3. התנאים לקבלת נקודות זכות

כדי לקבל נקודות זכות בקורס זה עליך לעמוד בדרישות הבאות:

א) **צבירת משקל של לפחות 20 נקודות במטלות.**

ב) ציון של לפחות 60 נקודות בבחינת הגמר.

ג) ציון סופי בקורס של 60 נקודות לפחות.

לתשומת לבכם:

מדיניות קורס זה היא לאשר הזנת ציון אפס במטלות שלא הוגשו כנדרש בקורס.

סטודנטים אשר לא הגישו את מכסת המטלות המינימאלית לעמידה בדרישות הקורס ולקבלת זכאות להיבחן, ומבקשים שמטלות חסרות יוזנו בציון אפס, יפנו למוקד הפניות והמידע בטלפון **09-7782222** או **יעדכנו בעצמם** באתר שאילתא http://www.openu.ac.il/sheilta **קורסים ⇦ ציוני מטלות ובחינות ⇦ הזנת ציון 0 למטלות רשות שלא הוגשו.**

יש לקחת בחשבון כי מטלות אשר יוזן להן ציון אפס ישוקללו בחישוב הציון הסופי ובכך יורידו ציון זה ולא ניתן יהיה להמירן במטלות חלופיות במועד מאוחר יותר. על כן קיימת אפשרות שסטודנט אשר יעבור את הבחינה בהצלחה ייכשל בקורס (כשהממוצע המשוקלל של המטלות והבחינה יהיה נמוך מ- 60).

כלל זה איננו חל על מטלות חובה או על מטלות שנקבע עבורן ציון מינימום.

מטלת מנחה (ממ"ן) 11

הקורס: שפות תכנות (20905)

חומר הלימוד למטלה: פרק 1 בספר הלימוד, פרק 3 במדריך הלמידה, חלקו השני של מדריך הלמידה העוסק בשפת scheme ובסביבת העבודה racket וכן מדריכים באתר הקורס.

מספר השאלות: 4 **משקל המטלה:** 5 נקודות

סמסטר: 2019ב **מועד הגשה:** **15.3.2019**

- שליחת המטלות תתבצע רק באמצעות מערכת המטלות המקוונת באתר הבית של הקורס.

שימו לב, בכל מקום במטלה בו נכתבה המילה סמל, הכוונה היא ל-scheme symbol.

שאלה 1 (25 נקודות)

כתבו פרוצדורה בשם calc_nested המקבלת רשימה (היכולה להכיל גם תתי רשימות) ומחזירה מספר המייצג את רמת הקינון של הרשימה.

להלן מספר דוגמאות:

```
>(calc_nested  `(a b c))
0
>( calc_nested  `(a ( ?) c))
1
>( calc_nested  `((a ((?)) ((d) b c))))
3
```

שאלה 2 (25 נקודות)

כתבו בשפת Scheme פרוצדורה בשם double-val המקבלת רשימה ומספר. הפרוצדורה מחזירה רשימה חדשה באותו מבנה של הרשימה שקיבלה אך עם מופע כפול של המספר שהתקבל כפרמטר.

הרשימה המתקבלת כפרמטר יכולה להיות רשימה מקוננת.

להלן מספר דוגמאות:

```
>( double-val  '(7 b c 7)  7 )
(7 7 b c 7 7)
>(double-val  '( 2 2 2 ) 2)
(2 2 2 2 2 2)
>(double-val  '((a ((5)) ((5) b 5)))  5)
((a ((5 5)) ((5 5) b 5 5)))
```

שאלה 3 (25 נקודות)

zip היא פרוצדורה המקבלת 2 רשימות באורך זהה, הפרוצדורה מחזירה רשימה חדשה בה כל איבר מורכב מרשימה בת שני איברים שנלקחו ממקומות תואמים ב-2 הרשימות הנתונות.

לדוגמא:

```
>(zip '(1 2 3)  '(a b c))
'((1 a) (2 b) (3 c))
```

א) ממשו את zip באמצעות שימוש ב- map

ב) ממשו כעת את zip באמצעות שימוש ב- foldr (fold right) **<u>וללא</u>** שימוש ברקורסיה מפורשת!!

שאלה 4 (25 נקודות)

כתבו פרוצדורה בשם my_andmap הפועלת בצורה דומה לפרוצדורה andmap.

להלן הסבר על andmap כפי שמופיע ב- Racket Documentation:

```
(andmap proc lst ...+) → any                                  procedure
  proc : procedure?
  lst : list?
```

Similar to `map` in the sense that *proc* is applied to each element of *lst*, but

- the result is `#f` if any application of *proc* produces `#f`, in which case *proc* is not applied to later elements of the *lst*s; and
- the result is that of *proc* applied to the last elements of the *lst*s; more specifically, the application of *proc* to the last elements in the *lst*s is in tail position with respect to the `andmap` call.

If the *lst*s are empty, then `#t` is returned.

Examples:

```
> (andmap positive? '(1 2 3))
#t
> (andmap positive? '(1 2 a))
positive?: contract violation
  expected: real?
  given: 'a
> (andmap positive? '(1 -2 a))
#f
> (andmap + '(1 2 3) '(4 5 6))
9
```

מטלת מנחה (ממ"ן) 12

הקורס: שפות תכנות (20905)

חומר הלימוד למטלה: פרק 2 בספר הלימוד, פרק 4 במדריך הלמידה.

מספר השאלות: 3 **משקל המטלה:** 5 נקודות

סמסטר: 2019ב **מועד הגשה:** **5.4.2019**

- שליחת המטלות תתבצע רק באמצעות מערכת המטלות המקוונת באתר הבית של הקורס.

לצורך פתרון מטלה זו עליכם ללמוד היטב את פרק 2 בספר הלימוד בליווי של פרק 4 במדריך הלמידה.

שאלה 1 (50 נקודות)

בשאלה זו עליכם לממש טיפוס נתונים מופשט המייצג מספר בינרי ופעולות שונות על טיפוס נתונים זה.

BinaryNum הוא טיפוס נתונים מופשט (ADT) המייצג מספר בינרי ותומך בפעולות הבאות:

- create- מקבלת ספרה 0 או 1 מחזירה מספר בינרי עם סיבית אחת שערכה זהה לספרה שהתקבלה.
- add-bit - מקבלת כפרמטרים מספר בינרי וספרה 0 או 1, ומוסיפה ספרה זו למספר הבינרי כביט חדש מימין. פעולה זו מחזירה את המספר הבינרי החדש.
- change-bit - מקבלת מספר בינרי, אינדקס (החל מ- 1 מימין) ומספר 0 או 1. פעולה זו משנה את הביט במספר הבינרי באינדקס הנתון לערך שהתקבל כפרמטר, ומחזירה את המספר הבינרי החדש.
- to-base10 - מקבלת מספר בינרי ומחזירה את ערכו לפי בסיס 10.
- rotate-right - מקבלת מספר בינרי ומסובבת את הסיביות שלו ימינה בסיבית אחת, ומחזירה את המספר הבינרי החדש.
- rotate-left - מקבלת מספר בינרי ומסובבת את הסיביות שלו שמאלה בסיבית אחת, ומחזירה את המספר הבינרי החדש.

א) כתבו אפיון לחתימות של הפעולות. סווגו את הפעולות השונות לסוגים (בנאים, מחלצים, מנבאים).

ב) כתבו אפיון אלגברי לכל אחת מהפעולות המגדיר את הסמנטיקה (המשמעות) שלהן.

הקפידו על הגדרה מלאה לכל פעולה.

ג) ממשו את טיפוס הנתונים המופשט BinaryNum עלפי הפעולות שהוגדרו באפיון של טיפוס הנתונים. ניתן לכתוב בנוסף פרוצדורות עזר במקרה הצורך.
אופן המימוש נתון לבחירתכם ויכול להיות ע"י רשימות או פרוצדורלי או ע"י שימוש ב-define-datatype

שאלה 2 (30 נקודות)

שאלה זו עוסקת בתחביר מוחשי (concrete syntax) ובתחביר מופשט (abstract syntax).

ביטוי חשבוני בכתיב prefix (כתיב פולני) הוא ביטוי שבו תחילה רושמים את הפעולה החשבונית שרוצים לבצע ולאחריה את האופרנדים הדרושים. בשאלה זו נעסוק רק בפעולת חיסור.

להלן נתון דקדוק המגדיר prefix-list שהיא רשימה שכוללת בתוכה ביטוי בכתיב prefix.

Prefix-list ::= (***Prefix-exp***)

Prefix-exp ::= ***Int***

::= - ***Prefix-exp*** ***Prefix-exp***

למשל,הביטוי בכתיב המוחשי (7 12 - 4 - 2 3 - -) הוא *Prefix-list* חוקי.

להלן נתון ייצוג של *Prefix*-exp באמצעות define-datatype:

```
(define-datatype  prefix-exp prefix-exp?
  (const-exp
    (num integer?))
  (diff-exp
    (operand1 prefix-exp?)
    (operand2 prefix-exp?)))
```

הביטוי הבא הוא דוגמא לביטוי הכתוב בכתיב מוחשי:

(7 12 - 4 - 2 3 - -)

להלן עץ תחביר מופשט (AST) עבור ביטוי זה :

```
(diff-exp
  (diff-exp
    (const-exp 3)
    (const-exp 2))
  (diff-exp
```

```
(const-exp 4)
(diff-exp
  (const-exp 12)
  (const-exp 7))))
```

כתבו פרוצדורה בשם parse-prefix המקבלת רשימה המייצגת *Prefix-list* חוקי (בתחביר מוחשי)

וממירה אותו לתחביר מופשט. (מדפיסה את ה-AST)

רמז : כתבו פרוצדורה שמקבלת רשימה ומחזירה *Prefix-exp* וגם את יתרת הרשימה שעדיין לא טופלה.

לפני פתרון השאלה רצוי לחזור על הדוגמא להמרת ביטוי מתחביר מוחשי לתחביר אבסטרקטי המופיעה בעמוד 53 בספר הלימוד.

שאלה 3 (20 נקודות)

בעמוד 40 בספר הלימוד נתון מימוש לטיפוס הנתונים **סביבה.** מימוש זה עושה שימוש בייצוג פרוצדורלי לייצוג הסביבה.

על סמך מימוש זה, כתבו פרוצדורה בשם ?has-binding המקבלת סביבה e ומשתנה v ובודקת האם ל- v ישנה כריכה תחת הסביבה e.

שימו לב, בשאלה זו הייצוג של סביבה הוא פרוצדורלי.

מטלת מנחה (ממ"ן) 13

הקורס: שפות תכנות (20905)

חומר הלימוד למטלה: פרק 3 בספר הלימוד, פרק 5 במדריך הלמידה, נספח B בספר הלימוד.

מספר השאלות: 2 **משקל המטלה:** 5 נקודות

סמסטר: 2019ב **מועד הגשה:** **3.5.2019**

- שליחת המטלות תתבצע רק באמצעות מערכת המטלות המקוונת באתר הבית של הקורס.

לצורך פתרון מטלה זו עליכם ללמוד היטב את פרק 3 בספר הלימוד בליווי של פרק 5 במדריך הלמידה, וכן בנספח B בספר הלימוד.

שאלה 1 (50 נקודות)

הרחיבו את שפת LET (שפת "ויהי") כך שביטוי if יבוצע על פי הדקדוק הבא:

להלן הדקדוק המגדיר את התחביר החדש עבור ביטוי if:

$$\textit{Expression} ::= \textbf{if } \textit{Expression} \ \textbf{ then } \{ \{\textit{Expression}\}^{+(;)}\} \ \textbf{else } \{ \{\textit{Expression}\}^{+(;)}\}$$

if-exp (bool exps1 exps2)

כאשר,

- bool הוא ביטוי בוליאני.
- exps1 הם אוסף ביטויים המופרדים בתו ; שאמורים להתבצע כאשר ערכו של הביטוי הבוליאני הוא true. הערך המוחזר במקרה זה יהיה ערכו של הביטוי האחרון שבוצע.
- exps2 הם אוסף ביטויים המופרדים בתו ; שאמורים להתבצע כאשר ערכו של הביטוי הבוליאני הוא false. הערך המוחזר במקרה זה יהיה ערכו של הביטוי האחרון שבוצע.

להלן דוגמא להמחשת השימוש בביטוי החדש:

```
let x= 7
in
  if zero? (x) then { -(x,4); -(x,9); x}
              else { if –(x,3) then {3} else {6} ; -4 }
```

בדוגמה זו, ערכו של x אינו 0 ולכן יבוצע בלוק ה-else ויוחזר ערכו. בבלוק ה-else מופיעים 2 ביטויים לביצוע. לאחר ביצועם, יוחזר ערכו של האחרון שבהם, במקרה זה (num-val -4)

שאלה 2 (50 נקודות)

הרחיבו את שפת LET (שפת "ויהי") עם ביטוי חדש בשם letsum.

להלן הדקדוק המגדיר את התחביר החדש עבור ביטוי letsum :

Expression ::= **letsum** ***identifier*** **= [{*Expression* }*(,)]**
in [{*Expression* }*(,)]

letsum-exp (id exps1 exps2)

כאשר,

- id הוא משתנה חדש המוכר רק במסגרת הביטוי.
- exps1 היא רשימה של אפס או יותר ביטויים המופרדים בפסיקים.
- exps2 היא רשימה של אפס או יותר ביטויים המופרדים בפסיקים.

שתי הרשימות, exps1 ו- exps2, אמורות להכיל את אותה כמות ביטויים. הביטוי letsum מחשב כל ביטוי מהרשימה exps2 עם id שערכו שווה לערך הביטוי מהמיקום התואם ברשימה exps1. כלומר, הביטוי הראשון ב-exps2 יחושב כאשר ערכו של המשתנה המצוין ע"י id שווה לערכו של הביטוי הראשון מהרשימה exps1, לאחר מכן יחושב ערכו של הביטוי השני מהרשימה exps2 כאשר ערכו של המשתנה המצוין ע"י id שווה לערכו של הביטוי השני מהרשימה exps1, וכן הלאה. ביטוי letsum יחזיר את סכום תוצאות חישוב הביטויים מהרשימה exps2.
במקרה של רשימות ריקות יוחזר אפס. במקרה של חוסר התאמה באורך הרשימות תודפס הודעת שגיאה. ניתן להניח שערכם של כל הביטויים ברשימות הוא מספרי.

להלן דוגמא להמחשת השימוש בביטוי החדש :

```
let x= 10
in
  let z=letsum x =[1,-(x,8),3] in [-(x,-1) , -(x,30),  -(x,-7)]
   in  -(z,x)
```

בדוגמה זו, בביטוי letsum הרשימה השנייה מכילה 3 ביטויים, הביטוי הראשון מחושב כאשר ערכו של x הוא 1. הביטוי השני מחושב כאשר ערכו של x הוא 2 (מכיוון שהביטוי השני ברשימה הראשונה הוא (x,8)– והוא מחושב על פי ה-x החיצוני שערכו 10).
ולבסוף הביטוי השלישי ברשימה השניה מחושב עם ערך של 3. ולכן בסה"כ ערכו של z יהיה שווה לסכום של הביטויים שחושבו ברשימה השניה. ולכן, z שווה ל- 16-.
הערך המוחזר של כל הביטוי המתואר בדוגמה הוא (z,x)– המחושב עם x שערכו 10. ולכן התשובה הכוללת שתוחזר היא (num-val -26)

מטלת מנחה (ממ"ן) 14

הקורס: שפות תכנות (20905)

חומר הלימוד למטלה: פרק 3 בספר הלימוד, פרק 5 במדריך הלמידה, נספח B בספר הלימוד.

מספר השאלות: 3 **משקל המטלה:** 5 נקודות

סמסטר: 2019ב **מועד הגשה:** **17.5.2019**

- שליחת המטלות תתבצע רק באמצעות מערכת המטלות המקוונת באתר הבית של הקורס.

לצורך פתרון מטלה זו עליכם ללמוד היטב את פרק 3 בספר הלימוד בליווי של פרק 5 במדריך הלמידה, וכן בנספח B בספר הלימוד.

שאלה 1 (30 נקודות)

שאלה זו עוסקת בשדרוג של שפת "וישגור" (שפת PROC) המתוארת בספר הלימוד בפרק 3 בעמודים 74-81. שפה זו מאפשרת הגדרה והפעלה של פרוצדורות לא רקורסיביות.

ברצוננו להרחיב את השפה עם ביטוי חדש שתחבירו נתון על-ידי הדקדוק הבא:

Expression ::= **argmax** ***Expression*** **[{** ***Expression*** **}**$^{+(;)}$**]**

```
argmax-exp (prc args)
```

כאשר,

- prc הוא ביטוי שתוצאתו היא פרוצדורה בעלת פרמטר יחיד המחזירה מספר.
- args הוא אוסף של 1 או יותר ביטויים שלכל אחד מהם טיפוס כשל הפרמטר של prc.

הביטוי החדש מפעיל את הפרוצדורה prc על כל אחד מהביטויים הרשומים ב- args. ומחזיר את את הערך המקסימלי שהתקבל מכל ההפעלות.

להלן דוגמה להמחשת השימוש בביטוי החדש:

```
let p2 = proc (y) –(y,5)
in
  argmax  p2  [-7;-(9,6);1]
```

בדוגמה זו, הפרוצדורה p2 מופעלת על כל אחד מהביטויים ברשימה, ומוחזר ערך ההחזר המקסימלי שהתקבל מכל ההפעלות של p2 על איברי הרשימה. ולכן ערכו של כל הביטוי במקרה זה יהיה (num-val -2)

שאלה 2 (30 נקודות)

א) בשפת "וישגור" (שפת PROC) הוגדר שלפרוצדורה יש רק ארגומנט יחיד. אך ניתן לעקוף מגבלה זו ולכתוב פרוצדורה עם מספר ארגומנטים ע"י שימוש בפרוצדורות שמחזירות בעצמן פרוצדורות.

נסתכל למשל על התוכנית הבאה בשפת "וישגור":

```
let f = proc (x) proc (y) …
in ((f 3) 4)
```

טכניקה זו נקראת Currying וכבר נתקלנו בה במסגרת השיעור הראשון בקורס.
כתבו בשפת "וישגור" באמצעות שימוש בטכניקה זו, פרוצדורה בעלת 2 ארגומנטים המחזירה את סכומם.
על מנת לחשב סכום של x+y בשפת "וישגור" ניתן לכתוב: -(x,-(0,y))

ב) להלן תוכנית בשפת "וישגור".

```
let makemult = proc (maker)
                proc (x)
                  if zero? (x)
                  then 0
                  else  -(((maker maker)  -(x,1)) , -4)
in  let times4 = proc (x) ((makemult makemult) x)
    in  (times4  3)
```

מהו הערך של תוכנית זו?

ג) כתבו בשפת "וישגור" פרוצדורה בשם f המחשבת את נוסחת הנסיגה:

f(0)=3 f(n)=2*f(n-1)

לשם כך השתמשו בטכניקה מסעיף ב', וכן ב- Currying.

שאלה 3 (40 נקודות)

שאלה זו עוסקת בשדרוג של שפת "וישגור" (שפת PROC) המתוארת בספר הלימוד בפרק 3 בעמודים 74-81. שפה זו מאפשרת הגדרה והפעלה של פרוצדורות לא רקורסיביות.

ברצוננו להרחיב את השפה עם ביטוי חדש שתחבירו נתון על-ידי הדקדוק הבא:

Expression ::= **mapsum** ***Expression*** **[{** ***Expression*** **}**$^{*(;)}$**]**

```
mapsum-exp (prc exps)
```

כאשר,

- prc היא ביטוי המתאר פרוצדורה המקבלת מספר ומחזירה מספר
- exps אוסף של 0 או יותר ביטויים שתוצאתן היא מספר. הביטוי החדש מפעיל את הפרוצדורה prc על כל אחד מהביטויים באוסף exps ומחזיר את סכום התוצאות שהתקבלו מכל ההפעלות. במקרה שהאוסף exps ריק יוחזר סכום שערכו 0.

להלן דוגמה להמחשת השימוש בביטוי החדש:

```
let p= proc (x) –(x,1)
in
  mapsum p [6; -(9,2); 2]
```

בדוגמה זו, הפרוצדורה p מופעלת על כל אחד מ-3 הביטויים באוסף, ומחזירה עבור כל אחד מהם את ערכו פחות 1. ולכן התוצאה הסופית תהיה הסכום של כל הערכים שהוחזרו מהפעלת p על כל אחד מהביטויים באוסף. במקרה זה, נקבל (num-val 12)

מטלת מנחה (ממ"ן) 15

הקורס: שפות תכנות (20905)

חומר הלימוד למטלה: פרק 4 בספר הלימוד, פרק 6 במדריך הלמידה.

מספר השאלות: 2 **משקל המטלה:** 5 נקודות

סמסטר: 2019ב **מועד הגשה:** **31.5.2019**

- שליחת המטלות תתבצע רק באמצעות מערכת המטלות המקוונת באתר הבית של הקורס.

לצורך פתרון מטלה זו עליכם ללמוד היטב את פרק 4 בספר הלימוד בליווי של פרק 6 במדריך הלמידה.

שאלה 1 (50 נקודות)

בספר הלימוד בעמודים 113-120 מתוארת שפת "**וירמוז**"(IMPLICIT-REFS).
ברצוננו להרחיב שפה זו עם ביטוי חדש בשם range.
להלן הדקדוק המגדיר את הביטוי החדש שברצוננו להוסיף:

Expression ::= range (identifier [{ ***Expression***}$^{+(,)}$]) ***Expression***

range-exp (tmpid exps body)

לביטוי **range-exp** המרכיבים הבאים:

- **tmpid** – משתנה זמני שמוכר רק במסגרת פעולת ביטוי range-exp.
- **exps** – רשימת ביטויים (לפחות אחד) המופרדים בפסיק.
- **body** – ביטוי אותו יש לבצע שוב ושוב

שלבי חישוב ביטוי range-exp הם:
המשתנה הזמני יקבל כערך את הערכים ברשימה exps, בכל פעם ערך יחיד (החל מהראשון ועד לאחרון) , עבור כל ערך שמקבל המשתנה הזמני, נדרש לבצע את הביטוי המתואר ע"י body.
הערך המוחזר מכל הביטוי הוא ערכו האחרון של body.

להלן דוגמא :

```
let z= 10
in let y= range (x [1,5,-2,-(3,8)]) begin
                                      set z= -(z,x);
                                      z
                                    end
   in y
```

בדוגמה זו, מוגדר משתנה z שערכו 10.
לאחר מכן, מוגדר משתנה y המקבל את ערכו של ביטוי range.
ביטוי ה-range מגדיר משתנה זמני x, בתחילה x מקבל ערך של 1 ועם ערך זה מבוצע ה-body שבמקרה זה משנה את ערכו של z להיות 9 (1-10) . לאחר מכן x משתנה ל- 5, ואז שוב מבוצע ה-body ששוב את ישנה את z ל- 4 (5-9). לאחר מכן, ערכו של x הוא 2- וערכו של z יהיה 6 (2+4). ולבסוף ערכו של x יהיה 5- וערכו של z 11 (5+6). ולכן y יקבל 11. ולכן ערכה של כל הדוגמה הוא : (num-val 11)

שאלה 2 (50 נקודות)

בספר הלימוד בעמודים 113-120 מתוארת שפת "**וירמוז**"(IMPLICIT-REFS).

ברצוננו להרחיב שפה זו עם ביטוי חדש בשם do-let

להלן הדקדוק המגדיר את הביטוי החדש שברצוננו להוסיף :

Expression ::= do-let (identifier ***Expression Expression) (Expression Expression***)

do-let-exp (tmpid init nxt bool retval)

לביטוי **do-let-exp** המרכיבים הבאים:

- **tmpid –** משתנה זמני שמוכר רק במסגרת פעולת הביטוי. כלומר בסיום חישוב הביטוי חוזרים לעבוד עם הסביבה שהייתה טרם ביצוע ביטוי ה-do-let.
- **init –** ערך התחלתי שיינתן למשתנה הזמני בתחילת ביצוע הביטוי.
- **nxt** – ביטוי שערכו יינתן למשתנה הזמני בכל פעם שערכו של ביטוי bool הוא false
- **bool –** ביטוי בוליאני שכל עוד ערכו **false** כל הביטוי יבוצע מחדש.
- **retval –** ביטוי שערכו יוחזר כתשובה סופית כאשר התנאי הבוליאני יהיה true

שלבי הביצוע של ביטוי do-let הם :
בתחילת חישוב הביטוי יוגדר משתנה זמני לצורך חישוב הביטוי, המשתנה המצוין ע"י tmpid. משתנה זה יאותחל בערכו של הביטוי init. לאחר מכן יחושב הביטוי הבוליאני המצוין ע"י bool, אם ערכו false, המשתנה tmpid יקבל כעת את ערכו של הביטוי nxt. ואז שוב ייבדק ערכו של התנאי הבוליאני bool וכל עוד הוא false, tmpid ימשיך לקבל את ערכו nxt. כאשר ערכו של הביטוי הבוליאני הוא true, יש להפסיק לבצע את ביטוי do-let וערכו של ביטוי retval יוחזר כתשובה סופית.
שימו לב, המשתנה הזמני המצויין ע"י tmpid הוא משתנה חדש שיש ליצור באופן זמני. כמו כן, כל אחד מהביטויים המצויינים בביטוי do-let צריך להיות מחושב מחדש בכל פעם שהוא נדרש.

להלן דוגמא :

```
let x=7
in let y= do-let (x 10 +(x,1)) (zero? -(x,15) x)
   in -(y,x)
```

בדוגמה זו, מוגדר משתנה x עם ערך 7. לאחר מכן, מוגדר משתנה y המקבל את ערכו של ביטוי do-let, ביטוי ה-do-let מגדיר משתנה x (חדש וזמני רק לצורך ה- do-let) משתנה זה מאותחל ב- 10, וכל עוד x שונה מ- 15 ערכו גדל ב-1. בהגיע x ל- 15 מוחזר ערכו של x, שהוא כרגע 15. ולכן משתנה y מקבל ערך של 15. כעת המשתנה x אליו תבוצע פנייה במסגרת גוף -v let השני הוא ה-x שערכו 7. ולכן ערכו של כל הביטוי הוא 8= 7-15
ולכן יוחזר (num-val 8)

מטלת מנחה (ממ"ן) 16

הקורס: שפות תכנות (20905)

חומר הלימוד למטלה: פרק 7 בספר הלימוד, פרק 7 במדריך הלמידה.

מספר השאלות: 1 **משקל המטלה:** 5 נקודות

סמסטר: 2019ב **מועד הגשה:** **14/6/2019**

- שליחת המטלות תתבצע רק באמצעות מערכת המטלות המקוונת באתר הבית של הקורס.

לצורך פתרון מטלה זו עליכם ללמוד היטב את פרק 7בספר הלימוד בליווי הפרק המתאים במדריך הלמידה.

שאלה 1 (100 נקודות)

להלן נתונה תוכנית בשפת "ויקיש" (INFERRED):

```
let  p = proc (w:?) (w 7)
in
  proc (m:?)
    (p m)
```

ברצוננו לקבוע מהו טיפוס התוכנית (אם קיים).לשם כך, עליכם להשתמש באלגוריתם שנלמד בפרק 7 להקשת הטיפוס.

א. (40 נקודות)

הקצו לביטוי הנתון בשאלה ולתתי הביטויים שלו משתני-טיפוס (Type Variables) מתאימים, וחברו משוואות מתאימות לביטוי הנתון ולתתי הביטויים שלו.

ב. (60 נקודות)

פתרו את המשוואות שהרכבתם בסעיף א' תוך שימוש ב- substitution ו- unification בדומה למתואר בספר בעמודים 252-258. בסיום פתרון המשוואות רשמו מהו הטיפוס שאותו הקשתם עבור התוכנית הנתונה.