

מושגים בשפות תכנות, שנה"ל תשע"ח, סמסטר ב', מועד א' פרופ' מולי שגיב, עודד פדון

הנחיות כלליות:

- משך הבחינה 3 שעות.
- מותר להשתמש בכל חומר כתוב (אין להשתמש במחשבון, מחשב, או פלאפון).
- בכל השאלות עליכם לתת הסבר משכנע מדוע התשובה שכתבתם נכונה, אלא אם מצויין אחרת.
- במבחן 7 שאלות. שאלה 1 היא חובה, ועליכם לבחור לענות על 5 שאלות מבין שאלות 2-7.
- את כל התשובות יש לכתוב במחברת הבחינה בלבד, תשובות על טופס הבחינה לא תבדקנה.
- חלוקת הניקוד: שאלה 1, 10 נקודות, שאלות 2-7, 18 נקודות כל אחת. סה"כ 100 נקודות.

נא להקיף בעיגול את השאלות שבחרת לענות עליהן:

7 6 5 4 3 2 1

חובה לענות על השאלה הבאה:

שאלה 1 - שאלת חובה (10 נקודות - 2 נקודות לכל סעיף)

בשאלה זו עליכם לסמן נכון / לא נכון בכל אחד מהסעיפים (בשאלה זו אין צורך לנמק).

1. בשפת JavaScript אפשר לשחרר את רשומת ההפעלה מהזיכרון בסוף ביצוע פונקציה.
2. בשפת OCaml המתכנת לא משחרר זיכרון בצורה מפורשת.
3. קיים T כך שלביטוי $(\lambda x:T. x)$ יש טיפוס ב simply typed lambda calculus.
4. ב OCaml אין תמיכה ב High-Order Functions.
5. התוכניות הבאות שקולות תחת Natural Operational Semantics:
 - `while true do skip ; x := 1`
 - `x := 1 ; while true do skip`

בחרו 5 שאלות מהשאלות הבאות. משקל כל שאלה 18 נקודות.

שאלה 2 - הגדרות אינדוקטיביות והוכחות באינדוקציה

נתון הדקדוק הבא, כאשר E הוא סמל ההתחלה:

$$E ::= T \mid E + T$$
$$T ::= id \mid (E)$$

א. כתבו את הדקדוק כהגדרה אינדוקטיבית של קבוצות, כפי שראינו בקורס.

ב. עבור כל אחת מהמילים הבאות, קבעו האם היא שייכת לשפה של הדקדוק הנ"ל. במקרה שהמילה בשפה - הראו עץ גזירה, במקרה שהמילה אינה בשפה - הוכיחו זאת.

id + id + id

id + id +

ג. הוכיחו שלכל מילה בשפה של הדקדוק הנ"ל מתקיים שמספר הסוגריים השמאליים שווה למספר הסוגריים הימניים.

שאלה 3 - סמנטיקה

נרצה להוסיף לשפת While את הפקודה הבאה:

```
for i := e downto zero do S
```

כאשר i הוא משתנה בתוכנית, e ביטוי אריתמטי, ו S היא פקודה בשפת While. זוהי לולאה עם אינדקס, i (שיכול להיות כל משתנה), וגוף הלולאה S צריך להיות מבוצע פעם אחת כאשר לאינדקס יש את הערך של הביטוי e בתחילת הלולאה, ואז שוב עם ערך קטן ב 1 , ושוב עם ערך קטן ב 2 (מהערך ההתחלתי), וכך הלאה עד שלבסוף גוף הלולאה יבוצע כאשר לאינדקס יש את הערך 0 , וזו האיטרציה האחרונה. ניתן להניח שגוף הלולאה S לא משנה את האינדקס. לדוגמה, עבור התוכנית הבאה:

```
for y := x downto zero do (x := x + y + 1)
```

אם נריץ אותה ממצב בו x הוא 1 , הריצה תסתיים במצב בו x הוא 4 ו y הוא 0 .
אם נריץ אותה ממצב בו x הוא 2 , הריצה תסתיים במצב בו x הוא 8 ו y הוא 0 .
אם נריץ אותה ממצב בו x הוא 3 , הריצה תסתיים במצב בו x הוא 13 ו y הוא 0 .

א. הרחיבו את ה $Natural\ Operational\ Semantics$ כדי לטפל בפקודה החדשה. הכללים אינם יכולים להסתמך על מבנה לולאת $while$ הרגילה בשפה (כלומר לא ניתן להתייחס בכללים למבנה לולאת $while$ הרגילה, אלא רק לפקודת הלולאה החדשה). הכללים חייבים להוות הגדרה אינדוקטיבית כפי שלמדנו בקורס, ואסור להם להוסיף משתנים נוספים לשפה.

ב. הדגימו את הסמנטיקה המורחבת שהגדרתם בסעיף א' ע"י בניית עץ גזירה לתוכנית הדוגמה מפסקת הפתיחה ממצב בו x הוא 1 .

ג. הרחיבו את ה $Structural\ Operational\ Semantics$ כדי לטפל בפקודה החדשה. הכללים אינם יכולים להסתמך על מבנה לולאת $while$ הרגילה בשפה (כלומר לא ניתן להתייחס בכללים למבנה לולאת $while$ הרגילה, אלא רק לפקודת הלולאה החדשה). הכללים חייבים להוות הגדרה אינדוקטיבית כפי שלמדנו בקורס, ואסור להם להוסיף משתנים נוספים לשפה.

ד. הדגימו את הסמנטיקה המורחבת שהגדרתם בסעיף ג' ע"י בניית סדרת גזירה לתוכנית הדוגמה מפסקת הפתיחה ממצב בו x הוא 1 .

שאלה 4 - λ -calculus

א. ציירו את ה AST המתאים לביטוי הבא ב λ -calculus :

$(\lambda x. \lambda y. y y) ((\lambda x. x x) (\lambda x. x)) (\lambda x. z)$

ב. כתבו סדרת חישוב (reduction) לביטוי מסעיף א' תחת $call\ by\ value\ semantics$.

ג. כתבו סדרת חישוב (reduction) לביטוי מסעיף א' תחת $lazy\ evaluation\ semantics$.

ד. מצאו טיפוס T כך שקביעת הטיפוס הבא תתקיים, וכתבו עץ גזירה שמוכיח אותה לפי כללי הטיפוס (typing):
(rules)

$x : bool \rightarrow bool \rightarrow bool, y : T \vdash x (y\ true) : T$

שאלה 5 - OCaml

א. נרצה לבנות ב OCaml אינטרפרטר לשפת While לפי Natural Operational Semantics. לצורך כך, הגדרנו את הטיפוס aexp עבור ביטויים אריתמטיים ואת הטיפוס bexp עבור ביטויים בוליאניים, ואנו רוצים להגדיר טיפוס statement עבור פקודות בשפת While. השלימו את החלקים החסרים בהגדרת הטיפוס statement:

```
type statement = Skip
                | Assign of string * aexp
                | Seq of statement * statement
                | If of bexp * _____
                | While of _____
```

ב. הניחו שהגדרנו גם טיפוס state עבור מצב של התוכנית, וכן שהגדרנו את הפונקציות הבאות:

```
aeval : aexp -> state -> int
beval : bexp -> state -> bool
setvar: state -> string -> int -> state
```

כאשר הפונקציות beval ו aeval מממשות את הסמנטיקה של ביטויים בוליאניים וביטויים אריתמטיים בהתאמה, והפונקציה setvar מעדכנת ערך של משתנה במצב. כתבו פונקציה:

```
nos: state -> statement -> state
```

שתקבל מצב s ופקודה S, ותחזיר מצב s' כך ש

```
<S,s> → s'
```

לפי Natural Operational Semantics. אם אין מצב s' כזה, הפונקציה לא חייבת להסתיים.

ג. נרצה לממש מצב ע"י הטיפוס string -> int, כתבו מימוש של הפונקציה setvar עם הטיפוס הבא:

```
setvar: (string -> int) -> string -> int -> (string -> int)
```

בסעיף זה אין להשתמש בפונקציות שהוגדרו בסעיף ב'.

שאלה 6 - JavaScript

נתון הקוד הבא ב JavaScript:

```
1 function make_f(n) {
2     var i;
3     var f;
4     f = function () { console.log(0); };
5     for (i = 1; i <= n ; i++) {
6         f = function () { console.log(i); f(); };
7     }
8     return f;
9 }
```

א. הפונקציה `make_f` אמורה לקבל מספר n , ולהחזיר פונקציה שכאשר נקרא לה יודפסו המספרים מ n עד 0 (כולל). למרבה הצער, הפונקציה לא פועלת כשורה. מה יקרה אם נריץ את הקוד הבא (אין צורך לנמק):
`make_f(10)();`

ב. תקנו את הקוד ע"י שינוי של גוף לולאת ה `for` בלבד, כלומר החלפה של שורה 6 בקטע קוד אחר. בקטע הקוד המוחלף אסור להשתמש בלולאות, ואסור לגשת למשתנה n .

ג. הסבירו כיצד פועל הקוד המתוקן שכתבתם בסעיף ב'.

שאלה 7 - Types

נתון המימוש הבא של פונקציית OCaml שאמורה לבדוק האם רשימה אחת היא תחילית (prefix) של רשימה אחרת:

```
let rec is_prefix xs ys = match xs with
| [] -> true
| x::xs' -> match ys with
| [] -> false
| y::ys' -> is_prefix xs' ys'
```

א. נתחו את הטיפוס של הפונקציה `is_prefix` ע"י אלגוריתם Hindley-Milner. עליכם לפרט את אופן פעולת האלגוריתם עד להגעה לטיפוס הכללי ביותר של הפונקציה.

ב. כיצד ניתן להבין מהטיפוס שיש באג במימוש?

ג. תקנו את הבאג.

ד. נתחו את הטיפוס של הפונקציה המתוקנת. עליכם להסביר בפירוט מה ישתנה בפעולת אלגוריתם Hindley-Milner על הקוד המתוקן לעומת סעיף א'.

בהצלחה,

מולי ועודד