

שפות תכנות – תרגיל 2

תאריך הגשה: 11.12.2022

הוראות הגשה: ההגשה בזוגות. כל זוג נדרש לחשוב, לפתור ולכתוב את התרגיל בעצמו. מותר להתייעץ עם זוגות אחרים אך חל איסור מוחלט להחזיק ולהיעזר בתרגיל כתוב של זוג אחר. יש לקרוא הוראות אלה בקפידה. הגשה שלא על פי הוראות אלה תוביל להורדת ניקוד ולא יתקבלו על כך ערעורים!

חומר עזר מומלץ: כדאי להבין היטב את הרצאות ותרגולים מספר 3-4. קישור לתרגולים (המצגות נמצאות בתיאור הסרטון):

<https://www.youtube.com/watch?v=IGN3a7czwXk&list=PLaMkJ2Pfx92I7DbMteYLYmMDyn3N0dDIT&index=3>

לנוחיותכם בנספח מופיעים הטבלאות של הסמנטיקה מהספר כפי שראינו בתרגול (אלו טבלאות 2.1 ו-2.2 מהספר המוזכרים בשאלות).

מה להגיש:

ast_sol.ml

semantics_sol.ml

nos_sol.ml

sos_sol.ml

nos.txt

sos.txt

בו יש את הפתרון לחלק א - ex2.pdf

בו יש את שם המשתמש מהסבמיט ותז של כל אחד מהמגישים, כל אח בשורה – id.txt.
חדשה.

יש להגיש את כל הקבצים בקובץ zip בשם ex2.zip.

- חלק א: הוכחות בסמנטיקה

שאלה 1:

א. הוכיחו את השקילות הסמנטית הבאה ב-Natural Operational Semantics:
 $(S1;S2);S3 \sim S1;(S2;S3)$

ב. הוכיחו את השקילות הסמנטית הבאה ב-Structural Operational Semantics:
 $(S1;S2);S3 \sim S1;(S2;S3)$

ג. הוכיחו שבמקרה הכללי לא מתקיימת השקילות ב-Natural Operational Semantics:
 $S1;S2 \sim S2;S1$

כלומר קיימים $S1, S2$ כך שהשקילות לא מתקיימת.

שאלה 2:

הוכיחו שה-Structural Operational Semantics של שפת While היא דטרמיניסטית. כלומר שמתקיימים שני תנאים:

- (1) אם $\langle S, s \rangle \Rightarrow^* s_1$ וגם $\langle S, s \rangle \Rightarrow^* s_2$ אז $s_1 = s_2$. בצורה פורמלית:
 $(\langle S, s \rangle \Rightarrow^* s_1 \text{ and } \langle S, s \rangle \Rightarrow^* s_2) \rightarrow s_1 = s_2$
- (2) לא ייתכן שתוכנית S גם תעצור וגם תגיע ללולאה אינסופית החל ממצב s מסוים (כלומר לא ייתכן של- $\langle S, s \rangle$ קיימת גזירה סופית וגם קיימת גזירה אינסופית).

שאלה 3:

נרצה לשנות לשפת While את הפקודה של while לפקודה הבאה:

do S while b

זוהי לולאה שתמיד מתבצעת פעם אחת לפחות, והביצוע שלה נפסק כאשר התנאי b אינו מתקיים. לדוגמא. הקוד הבא:

do x = x-10 while x>10

יסתיים במצב בו $x=5$ אם יתחיל במצב בו $x=55$, ויסתיים במצב בו $x=-3$ אם יתחיל במצב בו $x=7$.

הבהרה: בשפה המתקבלת לא קיימת פקודת **while b do S** אלא רק **do S while b**.

א. הוסיפו כללים/ים לטבלה של Natural Operational Semantics (טבלה 2.1 בנספח) שיגדירו את פקודת **do while**. הכללים אינם יכולים להסתמך על מבנה לולאת **while** הקיים בשפת **while** המקורית.

ב. הוכיחו את השקילות הסמנטית הבאה ב-Natural Operational Semantics המורחב שיצרתם בסעיף א:

do S while b \sim S; if b (do S while b) else skip

ג. הוסיפו כלל/ים לטבלה של **Structural Operational Semantics** (טבלה 2.2 בנספח) שיגדירו את פקודת **do while**. הכללים אינם יכולים להסתמך על מבנה לולאת **while** הקיים בשפת המקורית.

ד. (בנוס 5 נקודות): הוכיחו (עבור הכללים מהפתרון של הסעיף הקודם) ב- **Structural Operational Semantics** שמתקיים:

$$\langle \text{do } S \text{ while } b, s \rangle \Rightarrow^* s' \sim \langle S; \text{while } b \text{ do } S, s \rangle \Rightarrow^* s'$$

שימו לב שצד ימין משתמש בלולאת **while** הרגילה ואילו בצד שמאל משתמשים בלולאת **do while** שיצרתם בשאלה הקודמת.

- חלק ב: תכנות סמנטיקה

שאלה 1

בתרגיל זה נעשה שימוש ב-ocaml, שימו לב שהתוכנית שלכם רצה גם על לינוקס (שם אני בודק את התרגיל).

עם התרגיל כלולים 4 קבצים בנוסף:

סינטקס של שפת while – ast.ml

סמנטיקה של ביטויים בוליאניים ואריתמטיים – semantics.ml

סמנטיקה בצורת nos – nos.ml

סמנטיקה בצורת sos – sos.ml

ניתן לקמפל את הקובץ ast.ml בנפרד ולהוסיף בתחתית הקובץ עוד טסטים לבדיקה (טסטים אלו משמשים לבדיקות בקבצים nos.ml, sos.ml).

```
ocamlc -o ast ast.ml
```

כדי לקמפל את nos.ml צריך לקמפל ביחד עם הקבצים ast.ml ו- semantics.ml בעזרת פקודת קמפול:

```
ocamlc -o nos ast.ml semantics.ml nos.ml
```

לאחר מכן ניתן יהיה להריץ את בעזרת הפקודה ./nos.

באופן דומה כדי לקמפל את sos.ml צריך את פקודת הקמפול:

```
ocamlc -o sos ast.ml semantics.ml sos.ml
```

ואז ניתן להריץ את sos.ml בעזרת בעזרת הפקודה ./sos.

לפני פתרון הסעיפים מומלץ להסתכל על הקוד של הקבצים ולוודא הבנה של מימוש הדברים.

בסעיפים הבאים תצטרכו להוסיף לשפת while ביטויים חדשים כדי שה- interpreter יתמוך גם בהם. חובה לבדוק שהקבצים מתקמפלים ועובדים כראוי אחרי כל שינוי:

א. השלימו את הקובץ sos.ml כדי להשלים צריך להוסיף לפונקציה sos את הפירוש של פקודת If ופקודת While.

ב. הוסיפו כללים לסמנטיקה של ביטויים אריתמטיים עבור טיפול בפעולות הבאות (שימו לב סעיף זה אינו תכנותי):

$$x \ll y$$

$$x \gg y$$

תזכורת: האופרטורים <<, >> מקיימים את הזהויות הבאות:

$$x \ll y = x * 2^y$$

$$x \gg y = \lfloor x / 2^y \rfloor$$

- ג. הרחיבו את עץ הסמנטיקה שנמצא בקובץ של ast.ml כדי שיתמוך באופרטורים שהוספתם בסעיף א. כלומר אתם צריכים להוסיף את הבנאים Shl, Shr, ל-variant של aexp, שימו לב שכל אחד מאופרטורים אלו מקבל כפרמטרים שני ביטויים אריתמטיים ומחזיר ביטוי אריתמטי חדש. בנוסף תצטרכו להוסיף לפונקציה solve_a שבקובץ semantics.ml את הסמנטיקה כפי שהגדרתם בסעיף א.
- ד. שנו את המימוש של solve_b שבקובץ semantics.ml של הביטויים הבוליאניים כך שערכי האמת שיוחזרו ייצוגו כמחרוזות tt ו-ff במקום True ו-False. שימו לב שתצטרכו גם לשנות קצת בסמנטיקה שבקבצים nos.ml ו-sos.ml כדי לתמוך בשינוי החדש.
- ה. הריצו את התוכנית הבאה, גם עם nos.ml וגם עם sos.ml. בדקו שהתשובות שחזרו זהות!

```
a := 84 ; b := 22 ; c := 0 ; while b ≠ 0 do (
  a := a bit-shift-left 1 ;
  b := b bit-shift-right 1
)
```

כלומר, הוסיפו ביטוי המתאר את התוכנית כ-test5 ב-ast.ml. הוסיפו פקודת הרצה מתאימה ב-nos.ml ו-sos.ml עם ה-default_state (הנמצא בקובץ semantics.ml). שימרו את הפלטים בקבצים nos.txt ו-sos.txt בהתאמה (יש להגיש קבצים אלו).

- ו. בחלק א שאלה 3 הוספנו כללים של הפקודה do...while לסמנטיקה של nos. הוסיפו לעץ הגזירה שבקובץ ast.ml את הכללים לייצוג פקודת Do_While שמקבלת פקודה וביטוי בוליאני (בסדר הזה). ממשו ב-nos.ml את הסמנטיקה של do...while על פי כללי הגזירה שלכם בחלק א שאלה 3.
- ז. בחלק א שאלה 3 הוספנו כללים של הפקודה do...while לסמנטיקה של sos. ממשו את הסמנטיקה ב-sos.ml על פי כללי הגזירה שלכם בחלק א שאלה 3.

נספח:

טבלה של Natural operation semantic:

$[ass_{ns}]$	$\langle x := a, s \rangle \rightarrow s[x \mapsto \mathcal{A}[a]s]$
$[skip_{ns}]$	$\langle skip, s \rangle \rightarrow s$
$[comp_{ns}]$	$\frac{\langle S_1, s \rangle \rightarrow s', \langle S_2, s' \rangle \rightarrow s''}{\langle S_1; S_2, s \rangle \rightarrow s''}$
$[if_{ns}^{tt}]$	$\frac{\langle S_1, s \rangle \rightarrow s'}{\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \rightarrow s'} \text{ if } \mathcal{B}[b]s = tt$
$[if_{ns}^{ff}]$	$\frac{\langle S_2, s \rangle \rightarrow s'}{\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \rightarrow s'} \text{ if } \mathcal{B}[b]s = ff$
$[while_{ns}^{tt}]$	$\frac{\langle S, s \rangle \rightarrow s', \langle \text{while } b \text{ do } S, s' \rangle \rightarrow s''}{\langle \text{while } b \text{ do } S, s \rangle \rightarrow s''} \text{ if } \mathcal{B}[b]s = tt$
$[while_{ns}^{ff}]$	$\langle \text{while } b \text{ do } S, s \rangle \rightarrow s \text{ if } \mathcal{B}[b]s = ff$

Table 2.1 Natural semantics for **While**

טבלה של structural operation semantic:

$[ass_{sos}]$	$\langle x := a, s \rangle \Rightarrow s[x \mapsto \mathcal{A}[a]s]$
$[skip_{sos}]$	$\langle skip, s \rangle \Rightarrow s$
$[comp_{sos}^1]$	$\frac{\langle S_1, s \rangle \Rightarrow \langle S'_1, s' \rangle}{\langle S_1; S_2, s \rangle \Rightarrow \langle S'_1; S_2, s' \rangle}$
$[comp_{sos}^2]$	$\frac{\langle S_1, s \rangle \Rightarrow s'}{\langle S_1; S_2, s \rangle \Rightarrow \langle S_2, s' \rangle}$
$[if_{sos}^{tt}]$	$\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \Rightarrow \langle S_1, s \rangle \text{ if } \mathcal{B}[b]s = tt$
$[if_{sos}^{ff}]$	$\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \Rightarrow \langle S_2, s \rangle \text{ if } \mathcal{B}[b]s = ff$
$[while_{sos}]$	$\langle \text{while } b \text{ do } S, s \rangle \Rightarrow \langle \text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip}, s \rangle$

Table 2.2 Structural operational semantics for **While**