**שפות תכנות – תרגיל 2**

**תאריך הגשה**: 25.12.2022

**הוראות הגשה**: ההגשה בזוגות או לבד. 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. רק אחד מבני הזוג מגיש את התרגיל. 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=lGN3a7czwXk&list=PLaMkJ2Pfx92I7DbMteYLYmMDyn3N0dDIT&index=3>

לנוחיותכם בנספח מופיעים הטבלאות של הסמנטיקה מהספר כפי שראינו בהרצאה ובתרגול.

**מה להגיש:**

ast.ml

semantics.ml

nos.ml

bubble\_sort.ml

nos.txt

ex2.pdf - בו יש את הפתרון לכל השאלות התיאורטיות

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

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

* **חלק א: הוכחות בסמנטיקה**

**שאלה 1:**

1. הוכיחו את השקילות הסמנטית הבאה ב-**Natural** Operational Semantics:

(S1;S2);S3 S1;(S2;S3)

1. הוכיחו שבמקרה הכללי לא מתקיימת השקילות ב- **Natural** Operational Semantics:

S1;S2 S2;S1

כלומר קיימים S1,S2 כך שהשקילות לא מתקיימת.

**שאלה 2:**

1. הוכיחו כי ב- **Axiomatic** Semantics מתקיים: {false} S {P} עבור כל P.

הדרכה: יש להוכיח בעזרת אינדוקציה מבנית על S.

1. הוכיחו כי ב- **Axiomatic** Semantics מתקיים:

{x=5} if x=5 then y:=10; z:=y {z=10}

1. הוכיחו את השקילות הסמנטית הבאה ב-**Axiomatic** Semantics:

(S1;S2);S3 S1;(S2;S3)

**שאלה 3:**

נרצה להוסיף לשפת While את הפקודה של **repeat** לפקודה הבאה:

**repeat** S **until** b

זוהי לולאה שתמיד מתבצעת פעם אחת לפחות, והביצוע שלה נמשך כל עוד התנאי b אינו מתקיים. לדוגמא. הקוד הבא:

**repeat** x = x-10 **until**  x<10

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

1. הוסיפו כלל/ים לטבלה של **Natural** Operational Semantics (טבלה 2.1 בנספח) שיגדירו את פקודת **repeat**. הכללים אינם יכולים להסתמך על מבנה לולאת while הקיימת בשפת while המקורית.
2. הוכיחו את השקילות הסמנטית הבאה ב- **Natural** Operational Semantics המורחב שיצרתם בסעיף א:

**reapet** S **until** b S; **if** b **then** Skip **else**  (**reapet** S **until** b)b

* **חלק ב: תכנות סמנטיקה**

**שאלה 1**

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

עם התרגיל כלולים 4 קבצים בנוסף (שתתבקשו לתקן על מנת שיעבדו):

ast.ml – while סינטקס של שפת

semantics.ml – סמנטיקה של ביטויים בוליאניים ואריתמטיים

nos.ml –nos סמנטיקה בצורת

bubble\_sort.ml קובץ המייצר תוכניות למיון בועות -

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 bubble\_sort.ml צריך את פקודת הקמפול:

ocamlc -o bubble\_sort ast.ml semantics.ml nos.ml bubble\_sort.ml

ואז ניתן להריץ את bubble\_sort.ml בעזרת בעזרת הפקודה bubble\_sort/.

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

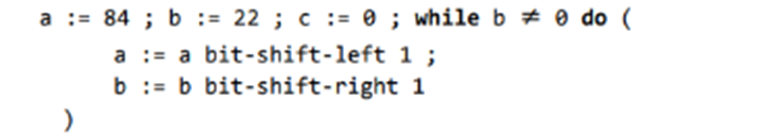
בסעיפים הבאים תצטרכו להוסיף לשפת while ביטויים חדשים כדי שה-interpeter יתמוך גם בהם. חובה לבדוק שהקבצים מתקמפלים ועובדים כראוי אחרי כל שינוי:

1. השלימו את הקובץ semantics.ml ואת הקובץ nos.ml. בקבצים אלו יש את הפונקציות ללא מימוש, ויש לממש את הפונקציות על מנת שתוכלו להריץ את קובץ nos עם הבדיקות שיש בו ללא שגיאות. ניתן להוסיף פונקציות עזר.
2. הוסיפו כללים לסמנטיקה של ביטויים אריתמטיים עבור טיפול בפעולות הבאות (שימו לב: סעיף זה אינו תכנותי):

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

*=* x \*

` =

1. הרחיבו את עץ הסמנטיקה שנמצא בקובץ של ast.ml כדי שיתמוך באופרטורים שהוספתם בסעיף 2. כלומר יש 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 ב.
2. 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 כדי לתמוך בשינוי החדש.
3. הריצו את התוכנית הבאה עם nos.ml. בדקו שהתשובות שחזרו הגיוניות!

כלומר, הוסיפו ביטוי המתאר את התוכנית כ-test5 ב-ast.ml. הוסיפו פקודת הרצה מתאימה ב-nos.ml עם ה- default\_state(הנמצא בקובץ semantics.ml). שימרו את הפלטים בקובץ nos.txt (יש להגיש קובץ זה).

1. בחלק א שאלה 3 הוספנו כללים לסמנטיקה של nos. הוסיפו לעץ הגזירה שבקובץ ast.ml את הכללים לייצוג פקודה זו שמקבלת פקודה וביטוי בוליאני (בסדר הזה). ממשו ב-nos.ml את הסמנטיקה של על פי כללי הגזירה שלכם בחלק א שאלה 3.
2. ברצוננו ליצור פונקציה המקבלת מספר n ומייצרת statement בשפת while הממיין במיון בועות n איברים (של מספרים שלמים). ממשו את כל הפונקציות עם פונקציות העזר בקובץ bubble\_sort.ml.

הדרכה:

* + עבור מספר טבעי n כלשהו צריך ליצור n משתנים חדשים.
  + לכתוב תוכנית בשפת while הממיינת אותם.
  + להוסיף בתחילת התוכנית n השמות כלשהן (שיהיה קל לשנות אותם בשביל לבצע בדיקות).
  + להדפיס למסך את המשתנים לפי הסדר בתחילת התוכנית ובסופה (כאשר נריץ את קובץ bubble\_sort צריך שיהיה הדפסה של המשתנים בתחילת הפונקציה ובסופה כאשר בסוף נראה אותם באופן ממויין).
* דוגמה עבור n=5 ניתן ליצור את ה-statment הבא:

x1 = 3;;

x2 = 5;;

x3 = 1;;

x4 = -2;;

x5 = 100;;

// print function of x1, x2, x3, x4, x5

##########################################

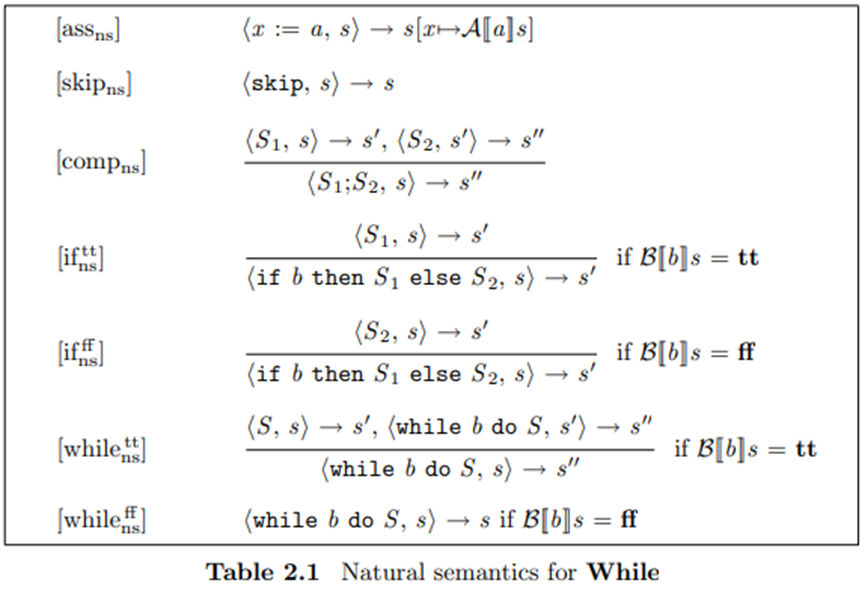
Your buble sort code here

##########################################

// print function of x1, x2, x3, x4, x5 sorted by values.

**נספח:**

**טבלה של Natural operation semantic:**



**טבלה של axiomatic semantic:**

