עקרונות הקומפילציה – הפרוייקט הסופי

מאיר גולדברג

הגשה: יום ו', 31 ינואר, 2025, בשעה 12:00 **בצהרים**

1 הנחיות כלליות

- העבודה היא ביחידים או בזוגות. צירוף לקבוצות גדולות יותר אסור, עבודות בשלושה שותפים או יותר לא תבדקנה, וּכְאִלּוּ לֹא הֻגְּשׁוּ!
- אין להשתמש בקוד של תלמידים או קבוצות אחרות, לא מִשָּׁנָה זו, לא משנים קודמות, ולא בקוד אחר שנמצא באינטרנט. הדברים נוגעים בהעתקה של קוד, בשלמותו או בחלקו, עם או בלי נסיונות להסתיר את ההעתקה. מקרים שיתגלו יועברו לטיפולה של **ועדת משמעת**. אין זו הדרך בה אנחנו, סגל הקורס, מעוניינים להתייחס לתלמידים שלנו, אבל העתקות הן בעיה חמורה ונפוצה, ואנחנו ננהג בהתאם **לתקנון משמעת הסטודנטים**.
- אתם אחראים בלעדית לבדוק את העבודה שלכם בטרם ההגשה:
  - אנחנו נשחרר דוגמאות להרצת הקוד, שמטרתן להדגים את הממשק ולספק מספר קטן של דוגמאות לקלט ולפלט תקינים. **הדוגמאות הללו לא תמצנה את בדיקת הנכונות**, ועליכם להצמד לטכסט של המטלה על מנת לממש אותה נכון!
  - **פלט מיותר הוא פלט שגוי**. ודאו שהקוד שלכם לא מייצר פלט שלא נתבקשתם ליצר (שְׁאֵרִיּוֹת של קוד ל־*debug* ו־*trace*, וכו'). ככלל, סגנון התכנות בקורס הוא פונקציונאלי, כלומר ללא *side effects*, ועל־כן אינכם אמורים להדפיס דבר. **החריג היחיד לכך הוא בפרוייקט הסופי**, לכשתתבקשו לכתוב לקבצים קוד באסמבלי. עד אז, מה שעשוי להראות כפלט הוא בפועל ההדפסה האוטומטית של הערך המוחזר מקריאה לפונקציה.
- אנא קראו את המסמך הזה בזהירות מתחילתו ועד סופו, וּוַדְּאוּ שאתם מבינים מה אתם מתבקשים לעשות בטרם תתחילו לתכנת!

2 הפרוייקט הסופי

לעבודה על הפרוייקט הסופי אתם מקבלים מספר קבצים:

- `add-to-compiler.ml`: הקובץ הזה כולל את הטיפוסים ואת רוב הקוד על מנת לממש את רכיב ה־*code generation* ב־*pipeline* של הקומפיילר
- `prologue-1.asm`: הקובץ הזה כולל את החלק הראשון של ההגדרות שיש לצרף לפני הפלט של ה־*code generator*
- `prologue-2.asm`: הקובץ הזה כולל את החלק השני של ההגדרות שיש לצרף לפני הפלט של ה־*code generator*
- `epilogue.asm`: : הקובץ הזה כולל את החלק הראשון של ההגדרות שיש לצרף אחרי הפלט של ה־*code generator*, וכולל קוד תמיכה באסמבלי ואת כל הפרוצדורות הפרימיטיביות

- `init.scm`: הקובץ הזה כולל תמיכה בפרוצדורות מובנות שאינן כתובות באסמבלי, אלא בסקים. הקובץ הזה משורשר לפני קוד המשתמש, ומתקמפל ביחד איתו, על מנת לספק תמיכה בפונקיות המובנות בסקים

- `makefile`: הקובץ הזה משמש לבניה מהירה של קבצי ביצוע מקבצי אסמבלי

את הקובץ `add-to-compiler.ml` עליכם להוסיף לקובץ `compiler.ml` ממטלות 2 ו־3. כלומר עליכם להעתיק ולהדביק את התוכן של הקובץ `add-to-compiler.ml` בסוף הקובץ `compiler.ml`. הקובץ `compiler.ml` יהיה הקומפיילר השלם.
בקוד תמצאו הערות מסוג:

```
raise (X_not_yet_implemented "final project")
```

מחקו נא את ההערה (או החליפו אותה באחרת!) וכתבו את הקוד שתומך במבנה שנתבקשתם לתמוך בו.
בקובץ `epilogue.asm` תמצאו את הקוד הבא:

```
L_code_ptr_bin_apply:
;;; fill in for final project!
```

מחקו נא את ההערה והחליפו אותה בהגדרה מתאימה עבור הפרוצדורה הפרימיטיבית `apply`, שלוקחת שני ארגומנטים, פונקציה ורשימה, ומפעילה את הפונקציה על איברי הרשימה. זו גירסא פשוטה יותר של הפרוצדורה `apply` בשפת סקים, ואת הפרוצדורה המלאה תמצאו בקובץ `scm.init`: היא משתמשת בפרוצדורה באסמבלי, אבל מבצעת עיבוד נוסף בסקים, שמקומפל על ידי הקומפיילרים שלכם.

- אנא התחילו לעבוד על המטלות הללו מוקדם, ואל תחכו לרגע האחרון!

- אל תתחילו לכתוב קוד מיד, אלא עברו קודם על כל המטלות, הבינו היטב למה אתם מתבקשים.

- עברו על הקובץ `compiler.ml`, סִפְּרוּ בכמה מקומות עליכם להשלים קוד עבור כל מטלה, עברו על הפונקציות והטיפוסים השונים, חשבו טוב על הקשר בין הפונקציות הנתונות לכם לבין מה שאתם צריכים לכתוב, ונסו לתכנן את העבודה שלכם.

- שִׁמְרוּ את הקובץ `compiler.ml`, עם הקוד שכתבתם במטלות. הקובץ הזה והקוד שלכם יְהַוּוּ את הבסיס לפרוייקט הסופי.

העבודות שתעשו במסגרת הקורס הזה קשורות קשר הדוק לשאלות שתראו בבחינות. קל לנו מאד לשכנע את עצמנו שאנחנו מבינים את הקוד כשאנחנו מסתכלים על פתרון של מישהו אחר. אבל זו לא רק הונאה לפי **תקנון המשמעת**, זו קודם כל **הונאה־עצמית**: בבחינות אתם תתקלו בבעיות שדומות, אך לא בהכרח זהות, לאלו שראיתם בעבודות. אותם תלמידים שיכתבו את הקוד בעצמם, שיבינו כל בורג בקוד שלהם, יוכלו להתמודד בהצלחה עם האתגרים בבחינה. אלו שיסתפקו "בְּלְהָבִין" קוד של אחרים במקום ליצור אותו בעצמם, לא יהיו מסוגלים לאלתר פתרונות בזמן הבחינה באותה מדה של אפקטיביות.

3 הוראות הגשה

1. אם אתם מגישים בזוגות, רק אחד משני השותפים צריך להגיש את המטלה בשם הקבוצה

2. עליכם להגיש קובץ *zip* ששמו מורכב ממספרי תעודות הזהות של השותפים, מופרדים על ידי קו־תחתון, או מספר תעודת הזהות במקרה של מגיש יחיד, והסיומת *zip*

3. קובץ ה־*zip* צריך ליצור תיקיה בשם `compiler`

4. אנא צרפו קובץ `readme.txt` שיכיל עבור כל אחד מהשותפים את השם המלא ואת מספר תעודת הזהות

5. כל הקבצים הרלוונטיים צריכים להמצא בתיקיה

6. בבקשה השאירו זמן פנוי לבדוק את קובץ ה־*zip* בטרם תגישו אותו: פתחו אותו בתיקיה חדשה, וודאו שכל הקבצים הדרושים שם!

7. אני מזכיר לכם שוב לשמור את הקובץ `compiler.ml`, משום שהוא ישמש אתכם בפרוייקט הסופי!

מבנה הקובץ ה־*zip* וקבצים שבו הוא כדלקמן:

- `123456789_987654321.zip`
  - `compiler`
    - `pc.ml`
    - `compiler.ml`
    - `prologue-1.asm`
    - `prologue-2.asm`
    - `epilogue.asm`
    - `init.scm`
    - `makefile`
    - `readme.txt`