

מרצה: נעם רינצקי

מתרגל: אורן איש שלום

חומר: פתוח

משך: שלוש שעות

מבחן בקומפילציה – 2016/17 – מועד א

- המבחן מורכב מחמש שאלות. יש לענות על כולן.
- מומלץ לקרוא את השאלה עד סופה לפני שמתחילים לענות.
- משקל השאלה ומספרה אינה מעיד על הקושי בפתירתה.
- יש לציין בראש העמוד את השאלה עליה עונים.
- אין לענות על שאלות שונות באותו העמוד.
- תשובה "איני יודע/ת" תזכה ב 20% מהניקוד על הסעיף הרלוונטי.

שאלה 1 (20 נקודות)

נתונה רשימת מאורעות. לכל אחד מהמאורעות הסבירו בקצרה (1-3 שורות)

1. האם הוא מתרחש במהלך בניית הקומפיילר/הקומפילציה/loading/linking/ריצת התוכנית.
2. אם בחרת בזמן קומפילציה, הסבר מהו השלב המסוים בקומפיילר בו מתרחש המאורע ומהם מבני הנתונים הרלבנטיים.

- א. (4 נקודות) התגלה כי המתכנת השתמש במילה שמורה כשם משתנה.
- ב. (4 נקודות) התגלה כי בכל ריצה של התוכנית, המתודה foo() מחזירה את ערך 0.
- ג. (4 נקודות) התגלה כי הדקדוק של שפת התכנות הינו LR(0).
- ד. (4 נקודות) התגלה כי אין די זיכרון בכדי להריץ את התוכנית.
- ה. (4 נקודות) התגלה כי אין צורך לשמור ערך הנמצא ברגיסטר שהינו Caller-saved.

שאלה 2 (25 נקודות)

לשאלה שני חלקים:

חלק 1: (5 נקודות)

נטען כי "בדיאגרמת המצבים של LR(0) לא יכולים להיות מעגלים מכוונים". מהי התשובה הנכונה ביותר לטענה זו מבין התשובות הבאות. נמקו בקצרה.

- א. נכון
- ב. לא נכון
- ג. יכול להיות מעגל מכוון, אבל אז אף מצב בתוכו איננו מצב reduce.
- ד. דיאגרמת המצבים היא גרף לא מכוון ולכן השאלה חסרת משמעות.

חלק 2: (20 נקודות):

הדקדוק הבא מייצג חלק משפת פעולות השורה על מטריצה:

(אותיות גדולות מייצגות tokens)

$S \rightarrow rows\$$

$rows \rightarrow row \ rows$

$rows \rightarrow row$

$row \rightarrow L \ R \ I$

$row \rightarrow L \ num \ R \ I$

$num \rightarrow I$

$num \rightarrow I \ D \ I$

- א. (10 נקודות) בנו את אוטומט ה LR(0) של הדקדוק.
- ב. (5 נקודות) האם הדקדוק שייך ל LR(0)?
- ג. (5 נקודות) האם הדקדוק שייך ל LR(1)? נמקו. (אין צורך לבנות אוטומט LR(1)).

שאלה 3 (25 נקודות)

הוחלט לייצר קוד משפת IC לאסמבלר אשר בו ישנה פקודת קפיצה אחת בלבד:

- **jump label** המאפשר קפיצה ל label נתון .

האם עדיין ניתן לתמוך במנגנון הקריאה לפונקציות נורשות דרך טבלת הפונקציות הוירטואליות?

אם התשובה שלילית – נמקו. אם התשובה חיובית – תארו את השינויים הנדרשים בתהליך יצירת הקוד.

שאלה 4 (20 נקודות)

במחשב שני רגיסטרים: $r1$ ו $r2$. לרגיסטר $r1$ יש תמיכה מיוחדת בפעולת ++ (הגדלת ערך הרגיסטר ב 1).

א. (15 נקודות) תארו שינוי לאלגוריתם הקצאת הרגיסטרים באמצעות צביעה אשר לוקח את התמיכה הזו בחשבון.

ב. (5 נקודות) הפעילו את האלגוריתם המשופר עבור הקוד הבא:

```
x = 1
y = 0
y ++
y ++
x ++
z = x + y
x = x + z
print (x) // x is live here!
```

שאלה 5 (10 נקודות)

אלונזו מוראלס לקח הדפסה של קובץ pseudo mips assembly

ומיהר לכיתה כדי להריץ עליו את אלגוריתם liveness Analysis שנלמד בקורס.

בדרך ירד גשם, ונמחקו הסימונות מכל השמות של ה temporaries למעט temporary אחד,

שאף מופע שלו בקוד לא נהרס: Temp_327. (שמות כל שאר ה temporaries הוא Temp).

האם עדיין ניתן להריץ את האלגוריתם ולקבל טווח חיות נכון עבור Temp_327?

בהצלחה!

אורן ונעם