

**מבחן בקומפילציה – 2016/17 – מועד ב**

- המבחן מורכב מחמש שאלות. יש לענות על כולן.
- מומלץ לקרוא את השאלה עד סופה לפני שמתחילים לענות.
- משקל השאלה ומספרה אינה מעיד על הקושי בפתיחתה.
- יש לציין בראש העמוד את השאלה עליה עונים.
- אין לענות על שאלות שונות באותו העמוד.
- תשובה "איני יודעת/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.
3. אם מאורע יכול להתרחש במספר שלבים, ציינו את כולם.

- א. (4 נקודות) התגלה קונפליקט מסוג shift-reduce.
- ב. (4 נקודות) התגלה כי ביצוע הפקודה  $x=y/z$  יגרום בוודאות לחלוקה באפס.
- ג. (4 נקודות) הוחלט להחליף את הפקודה  $x = y * z$  ב  $x = \text{shift-left}(z)$  המבצעת shift left בביט אחד ל  $z$ .
- ד. (4 נקודות) נקבעת הכתובת הווירטואלית של משתנה גלובאלי.
- ה. (4 נקודות) נקבע כי פעולת החיבור בפקודה  $x + y$  הינה מסוג חיבור floats.

**שאלה 2 (20 נקודות)**

נתונים שלושה דקדוקים: ( $X$  is the only non terminal)

1.  $X \rightarrow a \mid X + a \mid X + a + X$
  2.  $X \rightarrow a \mid X + a$
  3.  $X \rightarrow a \mid X + a + X$
- א. (4 נקודות) מהי השפה המתקבלת על ידי כל אחד מהדקדוקים.
  - ב. (4 נקודות) מי מהדקדוקים הנתונים (אם בכלל) שייך  $LR(0)$ ? נמקו.
  - ג. (4 נקודות) מי מהדקדוקים הנתונים הדקדוקים (אשר אינו ב  $LR(0)$ ) ניתן להפוך ל  $LR(0)$  ע"י פתירת קונפליקטים מסוג shift/reduce או reduce/reduce?
  - ד. (4 נקודות) מי מהדקדוקים הנתונים (אם בכלל) שייך  $LL(1)$ ? נמקו.
  - ה. (4 נקודות) האם ניתן להפוך את אחד הדקדוקים (אשר אינו ב  $LL(1)$ ) ל  $LL(1)$  ע"י הטרנספורמציות שנלמדו בכיתה?

**שאלה 3 (25 נקודות)**

הוחלט להוסיף לשפת ic את הביטוי reachableHeap אשר משתערך למספר מסוג int הנותן את גודל ה heap הנגיש בתוכנית.

תארו במדויק את השינויים הנדרשים בקומפיילר ובמערכת זמן הריצה בכדי לתמוך בביטוי זה.

#### שאלה 4 (15 נקודות)

בשפת התכנות ישנם ביטויים חשבוניים מסוג חיבור (+) וכפל (\*). ביטוי יכול להכיל משתנים וקבועים מסוג int ומסוג double. כל הפעולות מתבצעות בין רגיסטרים. לדוגמא, בכדי לחבר את משתנה x עם הקבוע 4, יש לטעון לרגיסטר אחד את התוכן של x, ולרגיסטר אחר את הקבוע 4, ורק לאחר כך ניתן לבצע את פעולת החיבור. ישנם שני סוגי רגיסטרים: כאילו המסוגלים לבצע פעולות בין ערכים מטיפוס int וכאילו המאפשרים לבצע פעולות בין ערכים מסוג double. כמו כן, יש פעולה מיוחדת  $R_d = \text{toDouble}(R_i)$  הממירה את הערך מסוג int הנמצא ברגיסטר  $R_i$  לערך מסוג double שמוצב לרגיסטר  $R_d$ .

א. (3 נקודות) תארו אילו שינויים נדרשים (אם בכלל) לאלגוריתם sethi-ullman להקצאת

רגיסטרים עבור ביטויים המכילים רק משתנים וקבועים מסוג double.

ב. (12 נקודות) תארו אילו שינויים נדרשים (אם בכלל) לאלגוריתם sethi-ullman להקצאת

רגיסטרים עבור ביטויים המכילים רק משתנים וקבועים גם מסוג int וגם מסוג double.

הערה: ניתן להתעלם מבעיות מסוג overflow ועיגול. הציון יינתן על פי נכונות וטיב האלגוריתם.

#### שאלה 5 (20 נקודות)

רוצים להוסיף לשפת ה IC את היכולת לזהות בזמן קומפילציה בכל נקודה בתוכנית לכל משתנה מסוג מצביע לאיזה טיפוסים (מחלקות) הוא יכול להצביע בזמן ריצה.

1. (15 נקודות) תארו אנליזה סטאטית מתאימה. זכרו: לאנליזה יש 5 אלמנטים  $(D, V, \sqcup, F, I)$ . ויש לתאר את כולם.

- על האנליזה להיות קונסרבטיבית (sound)
  - חלק מהציון יקבע על פי הדיוק של האנליזה (precision)
  - ניתן להניח כי המחלקות מכילות אך ורק שדות מטיפוס int.
2. (5 נקודות) הפעילו את האנליזה שתארתם על גבי התוכנית הבאה

```
L0: x = new C1()
L1: if (*) then y = new C2() else y = new C3() // if (*) means nondeterministic choice
L2: x = null
L3: x = y
L4: Goto L1
```

בהצלחה!

אורן ונעם