

תורת הקומפילציה

תרגיל 4

מתרגל אחראי: יונתן יעקבי – yonatan.yac@campus.technion.ac.il

ההגשה בזוגות

עבור כל שאלה על התרגיל, יש לעין ראשית בפיאצה ובמידה שלא פורסמה אותה השאלה, ניתן להוסיף אותה ולקבל מענה, אין לשלוח מיילים בנושא התרגיל בית כדי שנוכל לענות על השאלות שלכם ביעילות.

תיקונים לתרגיל יסומנו בצהוב, חובתכם להתעדכן בהם באמצעות קובץ התרגיל.

שאלה 1 – Backpatching (40 נקודות):

בשאלה זו נציע מבנה בקרה חדש שהתחביר שלו נתון על ידי:

- (1) $S \rightarrow \text{majority-while } (L) \text{ do } \{ S1 \}$
- (2) $L \rightarrow L1 \ B \mid B$

משמעות המבנה:

- כל עוד רוב התנאים ב-L מתקיימים, יש לבצע את S1.
 - רוב מוגדר באופן הבא: בין אם מספר התנאים זוגי (2n) או אי זוגי (2n+1) כדי שיהיה רוב לפחות n+1 מהתנאים חייבים להתקיים.
 - על התנאים שנבדקים להבדק בסדר בו הם מופיעים בקוד.
 - באיטרציה הראשונה בה לא מתקיים רוב התנאים הנגזרים מ-L, הלולאה מסתיימת מבלי לבצע את גוף הלולאה.
- א. הציעו פריסת קוד, מתאימה לשיטת **backpatching** עבור מבנה בקרה זה. על הקוד הנוצר להיות יעיל.
- ב. כתבו סכימת תרגום בשיטת backpatching המייצרת את פריסת הקוד שהצעתם בסעיף הקודם. אל תשכחו לכתוב לכל התכונות של L את השמות שלהן ואת התיאור שלהן.
- ג. אילו שינויים הייתם עושים בסכימה שכתבתם אם מבנה הבקרה היה מוגדר לבדוק את התנאים ב-L-מהתנאי האחרון לראשון (בסדר הפוך)?
- בתשובתכם, אין צורך להציג סכימת תרגום מלאה, אלא רק לתאר את השינויים הדרושים.

שימו לב:

- המשתנים S ו-B הם המשתנים הסטנדרטיים המופיעים בדף הנוסחאות, ויש להם כללי גזירה בנוסף לכלל המופיע בשאלה.
- למשתנה L אין כללי גזירה פרט לכללים המוצגים בשאלה.

הנחיות:

- הן בפריסת הקוד בסעיף א' והן בסכימת התרגום בסעיף ב', יש לדאוג שבעת בדיקת התנאים, לא יבדקו יותר תנאים ממה שנחוץ כדי לוודא שיש או שאין רוב.
- אין להשתמש בכללים סמנטיים באמצע כלל גזירה.
- אין להשתמש במשתנים גלובליים בזמן קומפילציה.
- אין לשנות את הדקדוק מלבד הוספת המרקרים M, N עם התכונות הסמנטיות שנלמדו בכיתה.
- יש להשתמש בתכונות נוצרות בלבד, ואין להוסיף תכונות סמנטיות ל-S, B.
- ניתן להשתמש בפונקציות `newstack()`, `stack.pop()`, `stack.top()`, `stack.is_empty()`, ובפונקציות המקבילות ל-`queue(FIFO)`.
- ניתן להשתמש בפונקציות העזר שהשתמשו בהם בתרגול.
- ניתן להגדיר משתנה עזר בסכימת התרגום ולשים בו את `nextquad()`, ולהשתמש בו כרצונכם.

שאלה 2 – DFA (60 נקודות):

בשיעור הגדרנו את המושג "משתנה חי", וראינו ניתוח חיות וכיצד ניתן להשתמש בו לצרכי האופטימיזציה useless code elimination.

נתונה התוכנית הבאה (זו התוכנית במלואה):

```
t1 := 5
t2 := t1
t3 := t2
t3 := 5
use(t3)    # this is some IR operation that is a definite use of
```

1. נתייחס ל-CFG של התוכנית הנתונה כאשר כל בלוק בסיסי מכיל פקודה בודדת. הפעילו ניתוח חיות כדי שנלמד בכיתה על התוכנית, וציינו את תוכן הקבוצות in/out של כל בלוק בסיסי ב-CFG.
2. הפעילו את האופטימיזציה dead code elimination כדי שנלמדה בכיתה על בסיס תוצאות האנליזה מהסעיף הקודם, וכתבו את התוכנית החדשה.
3. נשים לב שבתוצאה שקיבלנו בסעיף הקודם נשארו בתוכנית שורות שהם למעשה useless code. בשביל לפתור את הבעיה הזו, נצטרך להרחיב את הגדרת החיות של משתנה. לצורך זה, הגדירו מושג חדש של "משתנה חי ממש" כהרחבה של ההגדרה הקיימת של "משתנה חי".
4. הציעו אנליזת DFA (הפועלת על בלוקים בסיסיים בגודל פקודה בודדת) המחשבת את קבוצת המשתנים החיים ממש בכל נקודה בתוכנית. עליכם להגדיר את האנליזה כדי שלמדנו בשיעור. שימו לב שהאנליזה לא חייבת להיות מהצורה gen/kill.
5. האם האנליזה תתכנס? נמקו (אין צורך בהוכחה פורמלית).
6. הפעילו את האנליזה שהגדרתם על התוכנית הנתונה בתחילת השאלה, וציינו את תוכן הקבוצות in/out בכל בלוק. הראו כיצד הפעלת dead code elimination באמצעות תוצאות האנליזה שהגדרתם פותרת את הבעיה שראינו בסעיף 2.

Data-Flow Analysis

ההגדרות מתייחסות ל-CFG: $G = (V, E)$

הצורה הכללית של המשוואות לסריקה קדמית:

$$\begin{aligned} \text{in}(B) &= \bigcap_{(S,B) \in E} \text{out}(S) & \text{או} & \quad \text{in}(B) = \bigcup_{(S,B) \in E} \text{out}(S) \\ \text{out}(B) &= f_B(\text{in}(B)) \end{aligned}$$

הצורה הכללית של המשוואות לסריקה אחורית:

$$\begin{aligned} \text{in}(B) &= \bigcap_{(B,D) \in E} \text{out}(D) & \text{או} & \quad \text{in}(B) = \bigcup_{(B,D) \in E} \text{out}(D) \\ \text{out}(B) &= f_B(\text{in}(B)) \end{aligned}$$