**ממ״ן 15**

**שאלה 1:**

main

f

r

f

foo

bar

מצד שמאל מוצגים ה-Dynamic Links – הם תמיד מצביעים לפריים הקודם, כלומר לפונקציה הקוראת. לעומת זאת, בצד שמאל מוצגים ה-Static Links – הם דווקא מצביעים לפריים של המופע האחרון של הפונקציה המכילה, ובהתאם זהו המצב – f,r שניהם מוכלים ישירות ב- main, אך bar מוכלת ב-foo שמוכלת ב-f.

**שאלה 2:**

ניעזר במשתנה עזר labels, שייצר את המשתנים הזמניים והתוויות עבור ייצור הקוד:

*וכעת נוכל לבנות את הפעולות לייצור הקוד בצורה קלה:*

*הסבר: השתמשתי בתכונה target של המשתנה boolexpr – תכונה נבנית שהיא למעשה ערך המשתנה הזמני אשר מחזיק את תוצאת חישוב הביטוי.*

*מייצרים 4 תוויות – יציאה מהבלוק, משפט אמת, משפט שקר וביטוי בוליאני. קופצים קודם כל לחישוב הביטוי הבוליאני, לאחר החישוב בודקים האם הניב אמת או שקר – ובהתאם קופצים לענף המתאים בקוד. בסיום ביצוע הענף, יש קפיצה לסוף הבלוק והמשך של התוכנית.*

***שאלה 3:***

void stmt() {

switch(lookahead) {

case …:

case SILLY\_IF:

boolexp\_label = newlabel();

gen(“JUMP” boolexp\_label);

exit\_label = newlabel();

// True branch

true\_label = newlabel();

label(true\_label);

stmt();

gen(“JUMP” exit\_label);

// False branch

false\_label = newlabel();

label(false\_label);

stmt();

gen(“JUMP”, exit\_label);

// Boolean expression calculation

label(boolexp\_label);

result\_var = boolexp();

gen(“JMPZ” false\_label result\_var);

// End of block

label(exit\_label);

}

}

*דרך הפתרון זהה לחלוטין לפתרון שאלה 2, רק במבנה של Recursive Descent Parser.*

***שאלה 4:***

*באופן אנלוגי לסעיפים הקודמים, אך עם שינוי שפת הפלט:*

*נגדיר משתנה עזר:*

*וכעת הפעולות הנדרשת:*

***שאלה 5:***

1. *בהתאם לרמז, נתחיל בהגדרת הפעולות על טבלת הסמלים:*

*פעולת init – מאתחלת את טבלת הסמלים המקוננת הגלובלית של המנתח התחבירי.*

*פעולת make\_table – יוצרת טבלת סמלים חדשה שהיא טבלת בת של טבלת הקלט.*

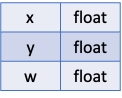
*פעולת insert – מוסיפה לטבלת הסמלים הנתונה את הסימבול הנתון בהתאם לטיפוסו הנתון.*

*פעולת lookup – מחפשת באופן רקורסיבי כלפי מעלה את הסימבול מתוך טבלאות הסימבולים שנבנו עד כה, ומחזירה את הטיפוס של הסימבול.*

*אם כך, סכימת התרגום יכולה להיות:*

*הסבר: בכל פתיחת בלוק יוצרים טבלת סמלים מקוננת חדשה על גבי טבלת הסמלים של הבלוק הקודם. כדי לבדוק המרה מ-Float ל-Int, שיכולה להתרחש אך ורק בעת השמה, אנו מחשבים את הטיפוס של E, ועל פיו לעומת ה-Lookup של יעד ההשמה – מגדילים את המונה.*

1. *טבלת הסמלים בהגעה לשורה 13:*



***שאלה 6:***

1. *אם המערך נשמר לפי שורות, כתובת האיבר הינה:*

*אם המערך נשמר לפי עמודות, כתובת האיבר הינה:*

1. *התשובה הנדרשת:*
2. *עץ כנדרש:*

***שאלה 7:***

1. *ריצת אלגוריתם Baker:*

|  |  |  |
| --- | --- | --- |
| *Scanned* | *Unscanned* | *Unreached* |
|  |  | *A B C D E F* |
|  | *A* | *B C D E F* |
| *A* | *C E* | *B D F* |
| *A C* | *E B* | *D F* |
| *A C E* | *B* | *D F* |
| *A C E B* | *F* | *D* |
| *A C E B F* |  | *D* |

*האובייקט שיועבר לרשימת ה-Free הוא היחיד שנותר ב-Unreached – והוא D.*

1. *הכתובות שיועתקו ל-To Space הן:*

|  |  |
| --- | --- |
| *אובייקט* | *כתובת* |
| *A* | *10,000* |
| *B* | *10,300* |
| *C* | *10,100* |
| *D* | *----------* |
| *E* | *10,200* |
| *F* | *10,400* |

*אין ייחוס לאובייקט D, הוא מתנקה, ועל כן – ברור כי הוא לא מועתק ל-To.*