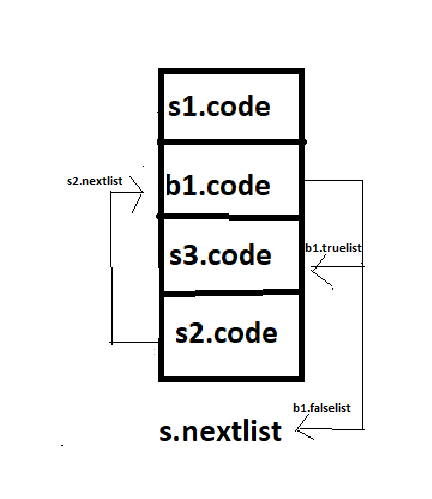
תרגיל 4

**שאלה 1**



א.

ב.

S→ for (S1; M1 B1; S2 N2) M2 S3

{

backpatch(N2.nextlist, M1.quad);

backpatch(B1.truelist, M2.quad);

S.nextlist = B1.falselist;

}

**שאלה 2**

א.

לפי ההגדרה החדשה של פונקציית ה- backpatching, אנחנו יכולים לבצע backpatching פעמיים לאותה הכתובת (התוויות יופיעו עם פסיק ביניהן). אנחנו לא מעוניינים לשנות כללים ש"דואגים" לבצע backpatching לכללים שהם גוזרים. כלומר, כפי שבדקדוק שראינו בהרצאה אנחנו דואגים לבצע backpatching נכון לכללים שגוזרים ביטויי FALSE ו-TRUE (ה-truelist וה-falselist שלהם מתעדכנים היטב ללא ידיעה אם באותו הרגע מדובר ב FALSE או ב-TRUE), עלינו לבצע עכשיו עדכון backpatching ל-truelist וה-falselist של ביטויי ה-don’t\_know.

נוכל, אם כן, לעשות זאת באמצעות שינוי של סכימת התרגום של הכלל:

B -> don’t\_know

כך שרשימות ה-truelist וה-falselist יצביעו על אותה הכתובת GOTO החדשה.

כך, כאשר יתבצע backpatching מתישהו בכלל אב הגוזר ביטוי B (שכאמור יכול להיות TRUE או FALSE בקוד המקורי), תעודכן פקודת ה-GOTO בשני המקרים.

ב.

ג.

B->don’t\_know {

Tmp = newTmp();

Tmp = nextInstr();

B.trueList = makeList(Tmp);

B.falseList = makeList(Tmp);

emit(“goto\_”);

}