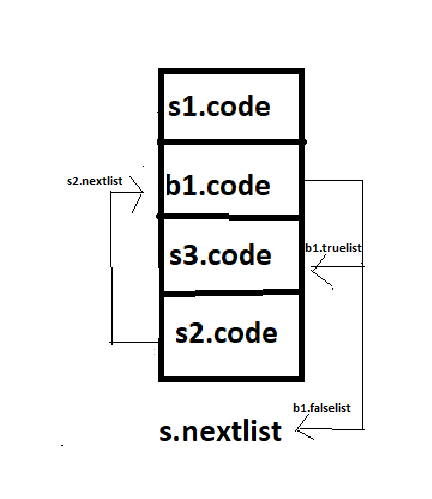
תרגיל 4

**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.

אם נדאג לטפל במקרה של don’t know כשגוזרים את הכלל B->don’t know, לא נצטרך לשנות את שאר הכללים, שהרי הם דואגים להתנהגות הרצויה, ללא ידע בכלל האם הביטויים משוערכים ל true/false (ולכן אם נדאג לטיפול ב don’t know, הכללים הנוכחיים יעבדו גם כאשר השערוך של הביטוי מגיע ל don’t know.)

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

B -> don’t\_know

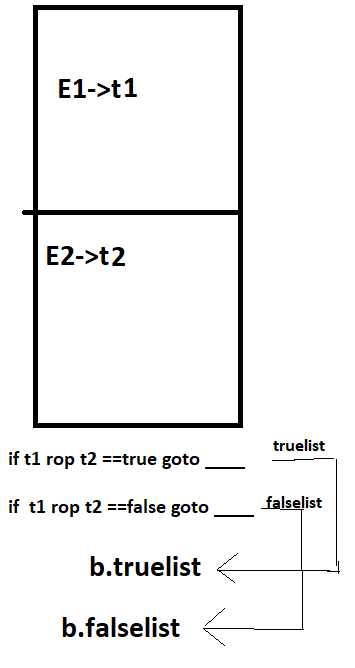
כך שרשימות ה-truelist וה-falselist יצביעו על אותה הכתובת GOTO החדשה.

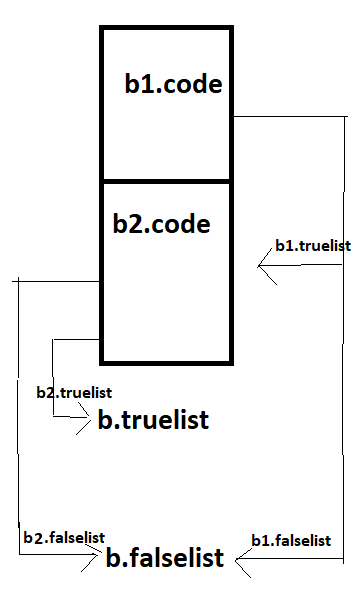
כך, כאשר יתבצע backpatching מתישהו בכלל אב הגוזר ביטוי B (שכאמור יכול להיות TRUE או FALSE בקוד המקורי), תעודכן פקודת ה-GOTO בשני המקרים.

כלומר בסה"כ אין צורך לשנות את סכימת התרגום, למעט כלל 3.

ב.

RULE 4: RULE 6:





ג.

B->don’t\_know {

Tmp = newTmp();

Tmp = nextInstr();

B.trueList = makeList(Tmp);

B.falseList = makeList(Tmp);

emit(“goto\_”);

}

B-> B1 and B2

{

Backpatch(b1.truelist, M.instr);

B.truelist = B2.truelist;

B.falselist = merge(B1.falselist, B2.falselist);

}

B-> E1 rop E2

{

B.truelist = makelist(nextquad());

B.falselist= makelist(nextquad()); // same thing as last row’s nextquad + 1

Emit(‘if’ E1.place relop.op E2.place ‘goto \_’);

Emit(‘goto \_’);

}