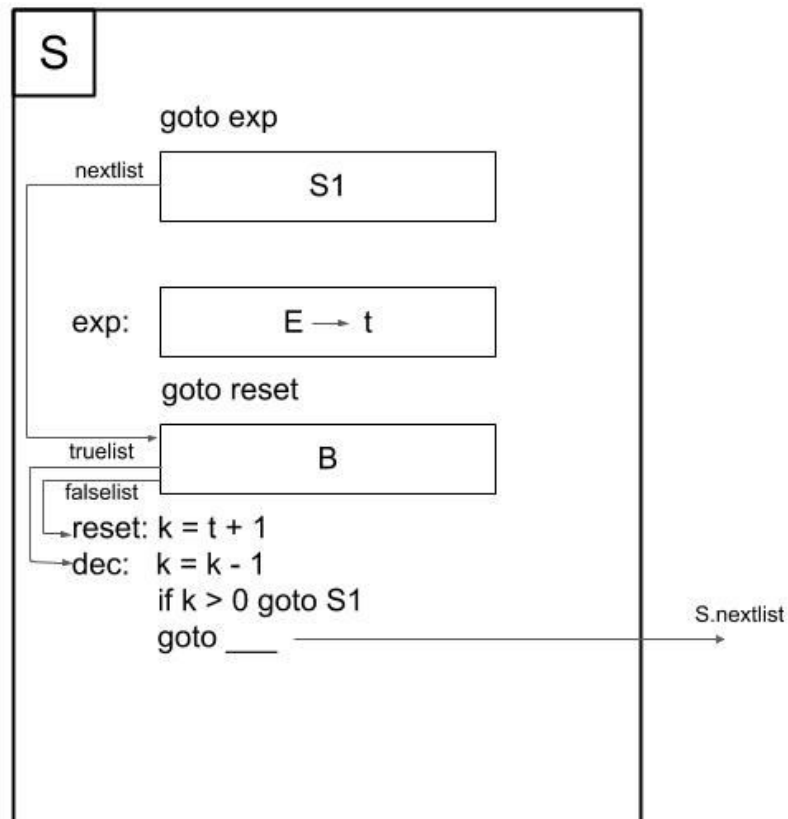


קומפילציה - תרגיל 4

שאלה 1:

א.



ב.

```
S -> repeat N1 M1 S1 until M2 E N2 times-in-a-row M3 B holds
{
    backpatch(N1.nextlist, M2.quad);
    backpatch(S1.nextlist, M3.quad);

    backpatch(merge(N2.nextlist, B.falselist), nextquad());
    counter = newtemp();
    emit(counter || " = " E.place || " + 1");

    backpatch(B.truelist, nextquad());
    emit(counter || " = " || counter || " - 1");
    emit("if " || counter || " > 0 goto " || M1.quad);

    S.nextlist = nextquad();
    emit("goto ____");
}
```

ג. ההטלאה של B.falselist תהיה עם M1.quad, במקום עם הכתובת שאחרי הקוד של B.

שאלה 2:

א.

1) $B \rightarrow \text{true}$

אין צורך בשינוי סכימת התרגום. הוספת ערך `dont_know` לא השפיעה על הסמנטיקה של `true`.

2) $B \rightarrow \text{false}$

אין צורך בשינוי סכימת התרגום. הוספת ערך `dont_know` לא השפיעה על הסמנטיקה של `false`.

3) $B \rightarrow \text{dont_know}$

זהו ערך חדש עם סמנטיקה חדשה. נצטרך להגדיר לו כללים סמנטיים.

4) $B \rightarrow B1 \text{ and } B2$

למרות שהסמנטיקה (טבלת האמת) של `and` הורחבה, אין צורך בשינוי סכימת הכללים הסמנטיים של כלל זה, בעקבות הגדרת סכימת התרגום של כלל 3, שיוגדר בסעיף הבא.

5) $B \rightarrow B1 \text{ or } B2$

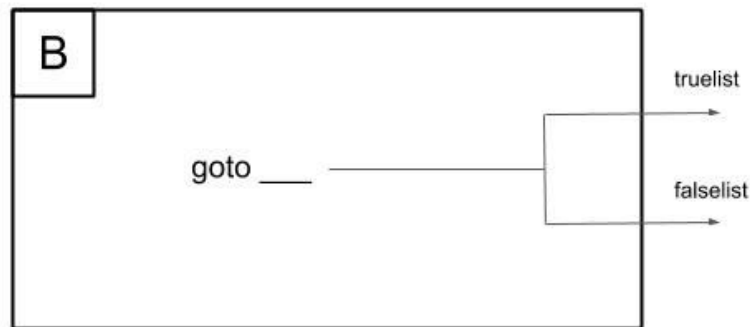
למרות שהסמנטיקה (טבלת האמת) של `or` הורחבה, אין צורך בשינוי סכימת הכללים הסמנטיים של כלל זה, בעקבות הגדרת סכימת התרגום של כלל 3, שיוגדר בסעיף הבא.

6) $B \rightarrow E1 \text{ rop } E2$

אין צורך בשינוי סכימת התרגום. `E1` ו-`E2` גוזרים ביטויים אריתמטיים, ותוצאת `rop` עבורם היא דטרמיניסטית, כלומר ערך הביטוי שייגזר יכול להיות רק `true` או `false`.

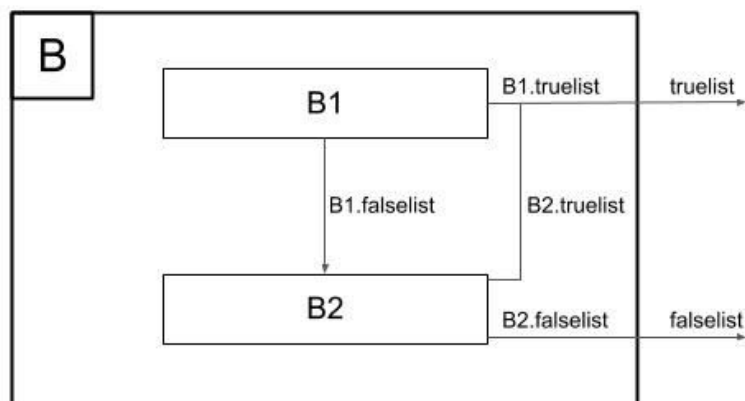
ב.

3) B -> dont_know



בפריסת קוד זו, ה-truelist וה-falselist של B מכילות את הכתובת של שורת הפקודה goto. זאת על מנת שאחרי שתבוצענה ההטלות של truelist ו-falselist, בשורת פקודה זו יופיע שרשור של הכתובת עבור המצב שהתנאי מתקיים, ושל הכתובת שהתנאי לא מתקיים (בסדר הזה או ההפוך), ואז בריצת התוכנית תתבצע קפיצה אקראית לאחת מכתובות אלו, בהתאם לסמנטיקת dont_know.

5) B -> B1 or B2

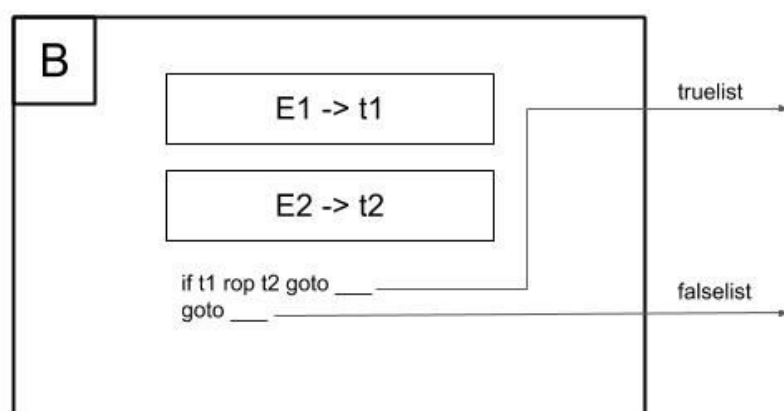


זוהי פריסת הקוד המקורית של כלל זה. נראה מדוע היא מייצגת סמנטיקה נכונה עבור טבלת or המורחבת, ע"י חלוקה למקרים:

- B1 משתערך ל-true - ה-truelist של B1 מוכלת ב-truelist של B, ולכן תתבצע קפיצה בהתאם לסמנטיקה שבה B משתערך ל-true, ללא תלות ב-B2, ובהתאם לטבלת האמת המורחבת של or.

- B1 משתערך ל-`false` - ה-`falselist` של B1 מוטלאת לכתובת של B2, ולכן תתבצע קפיצה בהתאם לסמנטיקה שבה B משתערך לערך של B2, בהתאם לטבלת האמת המורחבת של `or` (אם B1 הוא `false` אז B זהה ל-B2).
- B1 משתערך ל-`dont_know` - ה-`truelist` של B1 מוכלת ב-`truelist` של B, וה-`falselist` של B1 מוטלאת לכתובת של B2, לכן, תחילה, תתבצע קפיצה אקראית: או בהתאם לסמנטיקה ש-B משתערך ל-`true` או בהתאם לסמנטיקה ש-B מקבל את הערך של B2.
 - אם B2 משתערך ל-`true` אז בכל מקרה קיבלנו שתהיה קפיצה בהתאם לסמנטיקה של `true`, בין אם מדובר על הקפיצה הראשונה או על הקפיצה לאחר שערוך B2, בהתאם לטבלת האמת של `or`.
 - אם B2 משתערך ל-`false` אז או שהקפיצה תהיה בהתאם לסמנטיקה של `true` (הקפיצה הראשונה), או שהקפיצה הראשונה תהיה ל-B2, ואז הקפיצה הסופית תהיה בהתאם לסמנטיקה של `false`. כלומר, קיבלנו שתהיה סמנטיקה של `true` או `false` באופן אקראי, קרי סמנטיקה של `dont_know`, בהתאם לטבלת האמת.
 - אם B2 משתערך ל-`dont_know` אז או שהקפיצה תהיה בהתאם לסמנטיקה של `true` (הקפיצה הראשונה), או שהקפיצה הראשונה תהיה ל-B2, ואז הקפיצה הסופית תהיה עם סמנטיקה של `true` או `false` באופן אקראי. סה"כ מדובר בסמנטיקה של `true` או `false` באופן אקראי, קרי סמנטיקה של `dont_know`, בהתאם לטבלת האמת.

7) B -> E1 rop E2



כמוסבר בסעיף א', הסמנטיקה של כלל זה לא השתנתה, ולכן גם הכללים הסמנטיים של כלל זה לא השתנו, לרבות פריסת הקוד.

B -> dont_know

```
{  
    B.truelist = nextquad();  
    B.falselist = nextquad();  
  
    emit("goto ____");  
}
```

B -> B1 or M B2

```
{  
    backpatch(B1.falselist, M.quad);  
  
    B.truelist = merge(B1.truelist, B2.truelist);  
    B.falselist = B2.falselist  
}
```

B -> E1 rop E2

```
{  
    B.truelist = nextquad();  
    emit("if " || E1.place || " rop " || E2.place || " goto ____");  
  
    B.falselist = nextquad();  
    emit("goto ____");  
}
```