

תרגיל בית 4

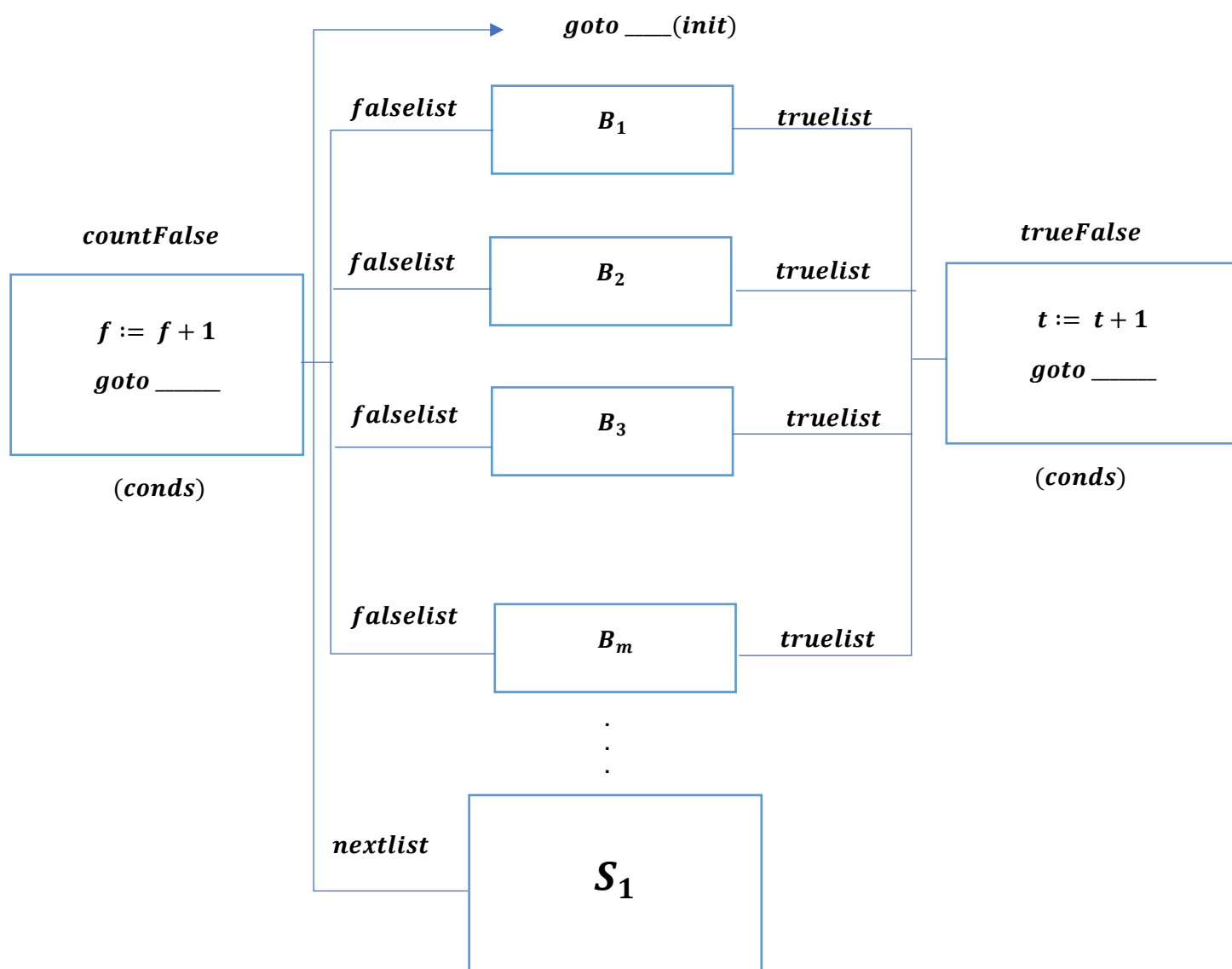
מגשים:

טל רוזנצוויג 307965806

שני אופיר 204512396

שאלה 1

סעיף א:



```

Init:      i: = 0
           f: = 0
           t := 0
           goto __ (conds)
conds:     i := i + 1
           if i = 1 goto B1
           if i = 2 goto B2
           .
           .
           if i = m goto Bn
           if t > f goto S1
           goto __ (next)

```

סעיף ב:

$$S \rightarrow \text{majority while } (N L) \text{ do } \{ M S_1 \}$$

$$L \rightarrow L_1 M B \mid M B$$

תכונות סמנטיות למשתני דקדוק:

B: truelist
falselist

L: nextlist
quadlist(queue)
truelist
falselist

S: nextlist

$L \rightarrow M B$

{

L.quadlist = *newqueue*();

L.quadlist.push(*M.quad*);

L.truelist = *merge*(*L.truelist*, *B.truelist*);

L.falselist = *merge*(*L.falselist*, *B.falselist*);

}

$L \rightarrow L_1 M B$

{

L.quadlist = *L₁.quadlist*;

L.quadlist.push(*M.quad*);

L.truelist = *merge*(*L.truelist*, *B.truelist*);

L.falselist = *merge*(*L.falselist*, *B.falselist*);

}

$S \rightarrow \text{majority while } (N L) \text{ do } \{ M S_1 \}$

{

backpatch(*N.nextlist*, *nextquad*())

//init

emit("i := 0")

emit("f:=0")

emit("t := 0")

next_quad = *makelist*(*nextquad*());

emit("goto ____");

cond_quad = *nextquad*();

backpatch(*next_quad*, *cond_quad*);

```

//conds
emit("i := i + 1");
cnt = 1;
while(!L.quadlist.empty()){
    quad = L.quadlist.pop();
    emit("if i =" || cnt || "goto" | |quad);
    cnt = cnt + 1;
}
emit("if t > f goto" | | M.quad);
emit("goto ____")    (next)
true_quad = nextquad();
//countTrue
emit("t := t + 1");
emit("goto" || cond_quad);
backpatch(L.truelist, true_quad);
//countFalse
false_quad = nextquad();
emit("f := f + 1");
emit("goto" || cond_quad);
backpatch(L.falselist, false_quad);
}

```

סעיף ג:

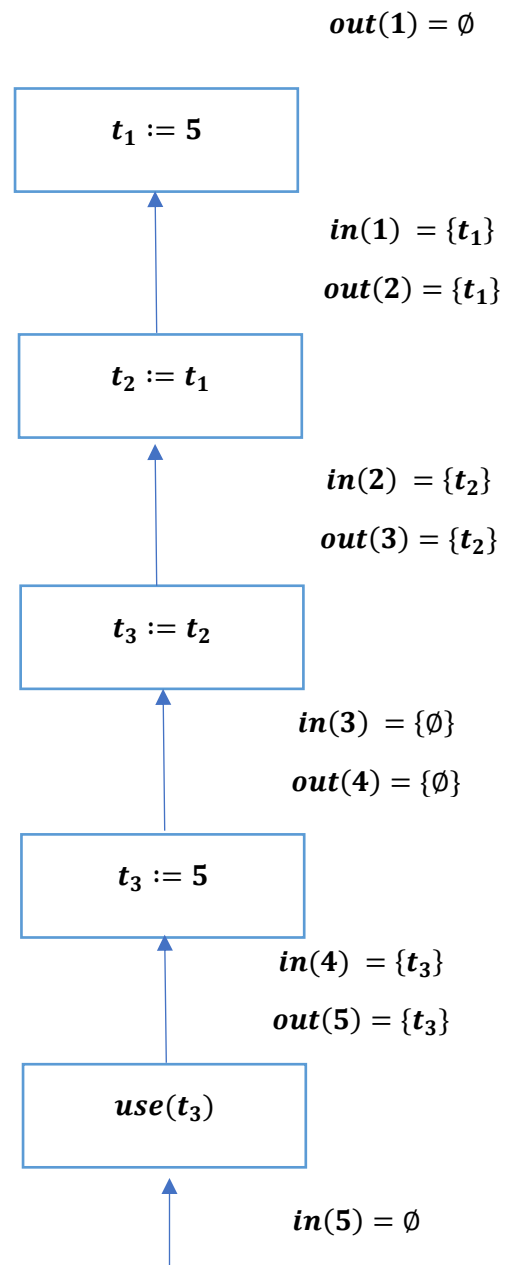
השינוי שנעשה בסכמה הוא להחליף את ה-queue במחסנית(כל השאר ללא שינוי). ההחלפה למחסנית תגרום לכך שהקוד לבדיקת התנאים יפלט ל-buffer בסדר הפוך.

שאלה 2:

סעיף 1

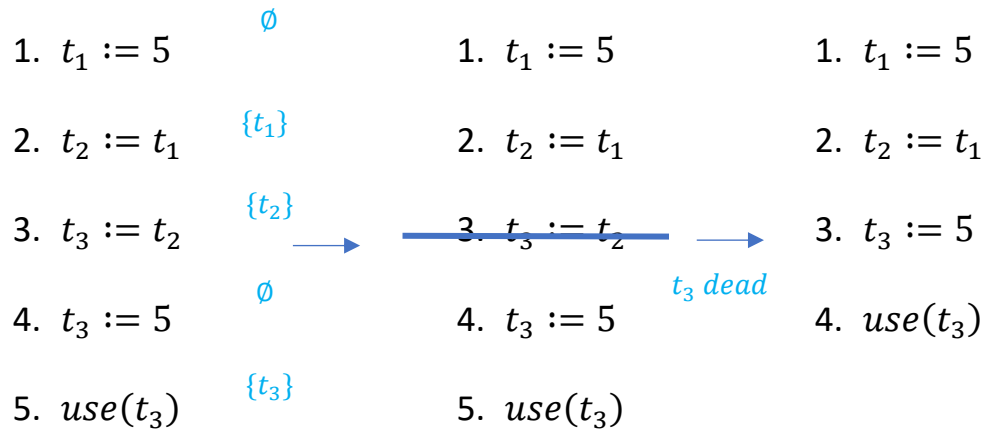
ניתוח חיות על התוכנית:

- 1) $t_1 := 5$
- 2) $t_2 := t_1$
- 3) $t_3 := t_2$
- 4) $t_3 := 5$
- 5) $use(t_3)$



סעיף 2

הפעלת האופטימיזציה Dead code elimination:



סעיף 3

נגדיר "משתנה חי ממש" בנקודה מסוימת בתוכנית אם מתקיים אחד מהבאים:

- משתמשים בערכו בהמשך התוכנית לא למטרת השמה.
- משתמשים בערכו להשמה ל"משתנה חי", לפי הגדרתו מההרצאה.

(בין 2 הנקודות קיים קשר "או").

סעיף 4

הצעת אנליזת DFA

נגדיר את האנליזה הבאה:

1. (Domain) $L = \{ x \mid x \in Var \}$.
2. כיוון הזרימה: אחורית.
3. אתחול בניסה לבלוק הראשון \emptyset .
4. הבעיה היא בעיית may.
5. $f(l) = (l \setminus kill(l)) \cup gen(l)$.

<i>Statement</i>	<i>Out(l)</i>
$x := exp$	$f(l)$
<i>use of x</i>	$f(l)$
<i>other</i>	$in(l)$

<i>Statement</i>	<i>kill(l)</i>	<i>gen(l)</i>
$x := exp$	$\{x\}$	$\{y \mid y \in \mathbf{exp} \text{ and } x \in (l)\}$
$use(exp)$	\emptyset	$\{x \mid x \in \mathbf{exp}\}$

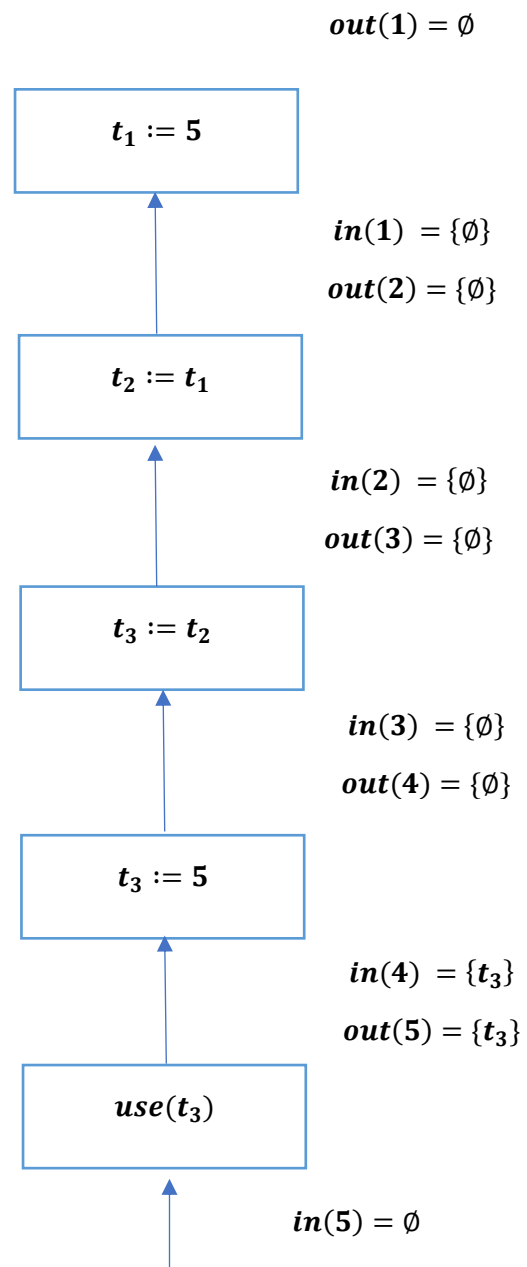
הסבר במילים ל-kill ו-gen:

עושים למשתנה *kill* אם התבצעה אליו השמה ועושים למשתנה *gen* אם הוא בשימוש או שעושים לו השמה למשתנה חי אחר.

סעיף 5

ראינו בהרצאה שבמידה והפונקציות *kill* ו-*gen* תלויות רק בבלוק הנוכחי אזי האנליזה תתכנס. באנליזה שהגדרנו מתקיים כי הפונקציות אכן תלויות בבלוק הנוכחי ולכן נקבל כי האנליזה תתכנס בהכרח.

- 1) $t_1 := 5$
- 2) $t_2 := t_1$
- 3) $t_3 := t_2$
- 4) $t_3 := 5$
- 5) $use(t_3)$



הפעלת האופטימיזציה Dead code elimination:

