# שאלה 1 – Backpatching

המארקרים שנגדיר:

goto init

פריסת קוד:

Check:

nextlist

truelist

next

goto check

init: t = 0

goto check

BT: if t==0:

goto \_\_ (next)

goto iterate

BF: if t==1:

goto \_\_ (next)

goto iterate

iterate: t = 1 – t

goto

falselist

תכונות:

סכימת תרגום:

{

bp(.nextlist, .quad);

bp(N.nextlist,nextquad()); // goto init

emit(goto .quad); // goto check

emit(t = newtemp());

emit(goto .quad); // goto check

bp(B.truelist,nextquad());

S.nextlist = makelist(nextquad());

emit(“if” “= 0: goto”);

emit(“goto” );

bp(B.falselist,nextquad());

S.nextlist=merge(S.nextlist,makelist(nextquad());

emit(“if” “= 1: goto”);

emit(“goto” );

bp(.quad,nextquad());

emit(t = 1 – t);

emit (“goto” .quad);

}

# שאלה 2 – Gen/Kill DFA

1. נגדיר את הדומיינים הבאים ולאחר מכן נבצע מכפלה ביניהם:

פריטי המידע שנשמור הם:

1. צורת החישוב היא סריקה אחורית – התכונה המתוארת תלויה בפקודות שבאות אחרי הנקודה ולכן יש לפעפע את המידע מהסוף עד לנקודה שבה רוצים לבחון את מצב המשתנה.
2. סוג הבעיה היא כי לפי הדרישה נקראת לפני ההשמה הבאה ל- ו- לא נקראת לפני ההשמה הבאה ל- **בכל מסלול חישוב**. מספיק שיהיה מסלול חישוב אחד שהדרישה לא מתקיימת כדי ש- לא יקרא -חיוני--קטלני לפי הגדרת הבעיה.
3. נגדיר את שתכיל את הזוגות לכל אם מתבצעת השמה ל-:
4. *נגדיר את שתכיל את הזוגות כך שמתבצעת ב- הקריאה :*
5. *משוואות הזרימה:*
6. מאופן פעולת החיתוך, בכל פעם אנחנו מורידים פריטים מ-.

לכל בלוק נאתחל באופן שמרני עבור בעיית must:

האתחול תקף גם עבור הבלוק בסוף התכנית, כי לא יהיו עוד פקודות השמה ולכן כל התנהגות תהיה חוקית.

1. נוכל לקבל את קבוצה זו ע"י אנליזת הבלוק של נקודה באופן הבא:

הקבוצה תהיה קבוצת המשתנים שמתאימים להגדרה המבוקשת.

הסבר: בכל הפעלה של אנו מסירים מ- את הזוגות המתארות קריאה לפונקציה. לכן, כדי לוודא ש- לא נקראת נרצה לבדוק ש- עדיין נמצא ב-. בנוסף, נראה לוודא שהפונקציה נקראה ולכן נבדוק ש- לא ב-. מאופן פעולת החיתוך, נקבל שהתכונה חייבת להישמר בכל מסלול שמתחיל בנקודה .