# 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 .

# שאלה 3 – Static Analysis

1. תחילה נגדיר לכל משתנה בתכנית סריג משלו. נניח שבוצעה אנליזה מקדימה שמבצעת ספירה של כמות המשתנים – נסמן ב- את מספר המשתנים בתכנית. *בנוסף, נניח שבוצעה אנליזה לזמן המקסימאלי שאובייקט יכול להיות נגרר ונסמן בזמן זה ב-.*  עבור משתנה נסמן ב- את הסריג המתאים לו.

הדומיין של הסריגים הללו יהיה:

לשם המחשה, הסריג יוגדר באופן הבא:

כעת, הסריג המתאים לאנליזה יהיה כאשר זה מספר המשתנים בתכנית.

הדומיין של הסריג הוא .

1. נגדיר את באופן הבא:
2. בהינתן פקודה בתכנית, הפונקציה תוגדר באופן הבא: הביטוי בתוך מבטא את תוכן השורה בתכנית:

הוכחת מונוטוניות:

נוכיח תחילה עבור כל בנפרד ואז לפי משפט מההרצאה נקבל שמכיוון ש- מונוטונית עבור כל כניסה בוקטור, אז מונוטונית עבור הוקטורים.

יהיו זוג וקטורים כך ש- מכסה את . נסמן ב- את הכניסה ה- של הוקטורים בהתאמה. ולכן לפי משפט נפריד למקרים:

1. ו- הוא כל מצב. אז מהגדרת נובע שלאחר הפעלת על הוקטור מתקיים ולכל מצב שאליו יעבור תתקיים מונוטוניות כי הוא חסם תחתון.
2. ו- הוא כל מצב. אז מהגדרת נובע שלאחר הפעלת על הוקטור מתקיים ולכל מצב שאליו יעבור תתקיים מונוטוניות כי הוא חסם עליון.

* כעת, מכיוון שלכל מתקיים אז לפי המשפט .

1. *את הפיתרון לשאלה נוכל לקבל ע"י הוקטור המתקבל ב- בהתאם למיקום הנקודה , ואם נראה באחת הקואורדינטות של הוקטור מספר שגדול מהסף שהגדרנו אז נדע שהמשתנה נגרר:*
2. *כעת נרצה לשמור עבור כל משתנה את כל המצבים שבהם הוא יכול להיות בו-זמנית, לכן נשמור בכל כניסה בוקטור כקבוצה. באופן פורמאלי נגדיר כ- את הסריג של כניסה בוקטור. הדומיין של הוא:*

*הסריג שלנו הוא*

*הדומיין של הסריג יהיה .*

*יחס הסדר יהיה הכלה איבר-איבר בוקטור, כלומר:*

*לשם המחשת מראה הסריג , הוא בנוי מ- רמות כאשר בכל רמה יהיו כל הקבוצות עם איברים מתוך .*

1. *וקטור המצבים של המשתנים מכיל קבוצות ולכן נגדיר . המשמעות היא לבצע איחוד איבר-איבר של 2 וקטורים: בהינתן , נגדיר:*
2. ה- תהיה זהה לפונקציה שהגדרנו בסעיף א', אך תפעל בנפרד על כל איבר בתוך הקבוצות. נסמנה ב-. באופן פורמאלי:

בנוסף, נתון שבלוק מכיל פקודה אחת בלבד ולכן בכל בלוק יש לכל היותר כניסה אחת בוקטור שמושפעת מביצוע הבלוק. נסמן ב- את הכניסה שמושפעת מביצוע הפקודה, לכן:

מסעיף א' היא מונוטונית ולכן נקבל:

פונקציה בסעיף ב' זהה לפונקציה מסעיף א' פרט לצורה שבה תפעל על קבוצות במקום על מספרים ספציפיים. יחס ההכלה מסעיף א' לא הופר על ידי החלפת ב ולכן גם מונוטוניות מובטחת.

1. נגדיר פונקציות שתפעל בצורה הבאה:

נגדיר את באופן הבא:

נשתמש בפונקציות הנ"ל כדי לקחת איבר ולכל נמיר אותו למצב ב- ע"י לקיחת האיבר המקסימלי בקבוצה שהוא המרחק המקסימלי שיכול להיות בין שימוש אחרון של משתנה לשחרור שלו, כלומר מרחק גרירה מקסימלי. ניצור וקטור בצורה הבאה:

ולאחר מכן נקבל תוצאה שתואמת את המבנה של סעיף א' ונקבל את הפתרון לשאלה באותה הדרך כמו שקיבלנו בסעיף א'.