

תורת הקומפילציה – אביב 2017

מועד א', פתרון חלקי

שאלה 2

א. הטרנספורמציה הנדרשת היא המרת בקרת הזרימה (כלומר, מבני בקרה כמו if) ל-data flow על ידי פרדיקציה. הטרנספורמציה נדרשת מכיוון שקיימים מקרים שונים במהלך הריצה בהם חלקים מהווקטור יתעדכנו לפי הפקודות עבור ענף ה-then (ה-taken logic) וחלקים אחרים עבור ענף ה-else (ה-non-taken logic).

ב. החיסרון העיקרי הוא הצורך להריץ את תוכן שני הענפים, על אף שבאיטרציות רבות של הלולאה הדבר לא נדרש.

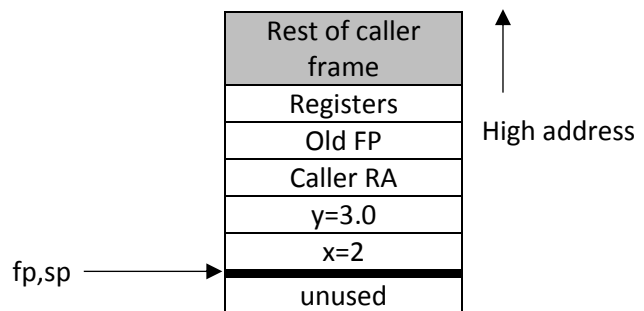
ג. האופטימיזציה הדינאמית בודקת בזמן ריצה האם עבור כל הכניסות ב-mask הנוכחי הן 0, ואם כן, מדלגת על ביצוע הענף לגמרי. הדבר בעצם מחזיר את בקרת הזרימה שהוסרה בטרנספורמציה בסעיף 1 במקרים בהם כל הכניסות בווקטור צריכות את הביצוע של אותן פקודות. האנליזה הנדרשת לשם כך תזהה את הצומת ב-CFG שאליו יש לקפוץ, כלומר איזה קוד שייך לענף הנוכחי ואיפה מסתיים הענף הנוכחי (שימו לב: ייתכן ויש לו ענפים מקוננים נוספים).

ד. האנליזה תתחיל מכל צומת N המופיעה אחרי הסתעפות בגרף, ותמצא צומת N' שהיא הפוסט-דומיננטור (post-dominator) האחרון ש-N הוא הדומיננטור שלו. כל הצמתים בין N ל-N' (כולל N ו-N') הם האיזור בגרף השייך לענף הנוכחי, ולכן ניתן לדלג עליהם כאשר כל ערכי ה-mask עבור taken/not taken הם אפס.

ה. סעיף זה נועד לעזור לזהות כי יש לקחת את הפוסט-דומיננטור האחרון, מכיוון שלקחת הפוסט-דומיננטור הראשון יוביל לפתרון לא אופטימלי.

שאלה 5

א.



ב. הפונקציה תחפש את הפרמטר השלישי שלא הוכנס על ידי הפונקציה הקוראת, מכיוון שהמימוש מצפה לה ומצפה להסכמה על החתימה (למי שיצא לו להיתקל: ככה נראות הרבה פעמים בעיות linkage). לכן, היא תיגש לחפש את הערך של z במיקום 3- או מעשית ב-

$$fp - (\text{sizeof}(\text{int}) + \text{sizeof}(\text{double}) + \text{sizeof}(\text{char} *))$$

או בעצם בזיכרון בגודל מצביע שנמצא מעל y. נשים לב כי נמצא שם כבר מצביע RA של הפונקציה הקוראת שנשמר טרם הקפיצה ל-printParams. הוא מצביע לאזור הקוד בזיכרון, ולכן הדפסה שלו כ-%s תבצע גישה לזיכרון המוצבע (נקודת החזרה של הפונקציה הקוראת באזור הקוד בזיכרון התכנית) ותכנו יודפס כמו מחרוזת ascii עד שיימצא terminating null ברצף הזיכרון. או בקיצור – יודפס הייצוג בזיכרון של הקוד שאחרי נקודת החזרה.

לאחר מכן, בהנחה ולא הייתה במהלך ההדפסה חריגת קריאה לתוך זיכרון מוגן, הפונקציה `printParams` תסיים את ריצתה בהצלחה ותקפוץ אל ה-RA שלה שנשמר בעת הקריאה ל-`printf`, וקוד האפילוג שאחרי הקריאה יוציא מהמחסנית את הפרמטרים, יטען את `Caller RA` בחזרה לרגיסטר `$ra`, ויאחזר את `fp` להצביע אל הפריים של הפונקציה הקוראת. שאר הריצה תמשיך כרגיל.