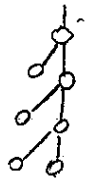
נמדדנו במטרה על סתי-זיכרון שמאפשר לשנות את דרך השימוש ברגיסטרים בתנאי שהשינוי לא ישפיע על התוכנית.

במידה ונמדדנו לגבי הרגיסטרים אנו משתמשים בשיטת השפירה של רגיסטרים לציבורין. תשים לב שמה הלוגיות מאוד, בדרך הם פועלים ומה הם עושים.

סתי זיכרון בעל 2-3 שלבים. בשלב הראשון מתבצע חישוב של רגיסטרים הדרושים לחישוב עבור כל פונקציה. במעבר השני הוא חוזר ומבצע חישוב של כל פונקציה עם תת הפלג הבבה קודם.

חשוב לציין שהשיטה פיבית אך ורק לבינאים שניין להם תיבות לוגי, כלומר, למשל, אם יש קריאה לפונקציה, לא נוכל לבצע האופה (כי יכול להיות שתיבות החישוב לפני ואחרי השינוי לא יהיו זהות).

עוד מאיפון הינו המקרה הבעיה של עץ בינארי מסוים (כי אנו נדחיק רגיסטר עבור כל רמה). המקרה הטוב ביותר הינו עץ בינארי חיוני:



בעץ מהבינארי הזה, אם נחשב קודם תמיד את הבן הימני נצטרך ל- 4 רגיסטרים. לעומת זאת, אם נחשב תמיד את בן שמאל, נצטרך לרגיסטר יחיד.

בדרך לדעת האם מספיקים לנו הרגיסטרים שהשתמשנו, נזקוק בעץ תלויות ונבדוק האם ניתן לבצע אותו במספר בפעולות הנדרש למספר הרגיסטרים. אם הקלינו, מספר הרגיסטרים מספיק. כיוון שהבעיה היא NP, אנו נשתמש בהיוריסטיקות בכדי לפתור את הבעיה.

בהינתן k רגיסטרים, על לינה בעץ תהיה בעיית שפירה של רגיסטרים? (קיים בגרף) k-1 תגרום לשפירה (כיוון שאנו לא ניתן לבצע את הגרף ב- k בבעיות).

מה המבנה הטוב ביותר לבצעת כל הדמתיב בגרף? אם נרצה לבצע אותו במספר הדמתיב בקירקה הגבוה ביותר בגרף.

מה המבנה הגרוע ביותר לבצע הדמתיב בגרף? אם נרצה לבצע את הגרף רק בדרגה הגבוהה ביותר שיש לבצע בגרף (מספר הדמתיב בגרף אי תשיבה לא הבולה).

Activation Record - הינו סביבה המידע המוקדה בתוכנית (שקופית 15). אנו מסתכלים על מחסנית רק שהתבית רגיל כלפי מעלה. בעת קריאה לפונקציה, תחילה בוחפש למחסנית את כל הרגיסטרים שבאחריות הפונקציה הקוראת לשמור מהפלג הקוראת, SP יורד כל כי מטה, מוסיפים את המשתנים שבדקנו לעמוד ולאחריהם את כתובת החזרה.

שחזר מכן, שומרים את הכתובת של ה-FP הקודם, SP מגבים לעמוד למחסנית שמתאים בתיבת בי וכן מעדכנים את FP להגבים לשם במחסנית (ישחזר בפלג לכתובת בי).

בשלב חזרה מוקדה מקום לכל המשתנים הלוקליים (SP מעשיק לרדת מטה), לעמוד, נשמרים כל הרגיסטרים שבאחריות המתקדמת הנקראת לשמור.

u.multimet.co.il

בסוף הריצה, מעדכנים את SP להצביע לאיפה שהוא נמצא כעת FP, את ה-FP מעדכנים לערך הקודם לפני הריצה. SP מצביע על ה Return Add, כדי שנוכל לבדוק את הפונקציה הקודמת באותו אופן, להמשיך את הפרמטרים שנדחפו למחסנית ולשחזר את הרגיסטרים שבאחריותה.

אנחנו משתמשים ב-FP כדי לגשת למשתנים. כלומר, כל הכתובות שלנו מהמחסנית יהיו ביחס אליו.

ניתן לשנות כיצד נרשמת הכתובת לגישה ריצה של תכונות מסוימות בשפה כלשהי.

קומה נוספת היא מהי נתון תקין תרגם לשון, כיצד ניתן לאפשר בשמירתו להקצות מקום בכתובות המקום בסדר?

היתרון של השימוש ב לשון יהיה שאי אפשר להבדל אוסף גבל. ההשפעה הינו שבעת הריצה מתיחסה הקוד של המשתנה לפי מיקומו.

כיצד תומכים בזה בקומפיילר? להוסיף לאותה הלקסיקה מילת מפתח חדשה, הוספת הכללים המתאימים לפרסר, ויתרת סמנטי - שימוש בטיפוס כללים של עשר סלה לעשר. ובחלק המסביב ל IR יש לטרגם בצורה דומה ל עשר חלק שימוש המתאים הדשרת הקצאת מקום במחסנית. בשלב המתרגם לאסמבלר, יש לתרגם את הפעולות המתאימות שירדנו. (שקפים 21-22).

דוגמה לשאלה:

* מתבקשים לממש אופרטור קבוע המגדה By Ref (בדומה ל- & ב-C). השינויים המתבקשים: הוספת & לקסר, מיכון הפרסר לכללי RefType, ניתוח סמנטי, בדיקה לבדוק את חוקי בדיקת טיפוסים וכן מיכון חלק של Scope Check. IR יש לעשות תיקון בקריאה ByRef ומיכון בחלק המתרגם. (שקפים 25-26).

* מתבקשים להוסיף בנאים ל- IC. בנאים דומים למגדה עם חתימה שונה. בחלק הלקסיקה לגדרך לשנות כלום. בחלק של הפרסר יש להגדיר כמגדה ללא ערך מוחזר ולקרוא לה בגוף (כאת מהבנאים האחרים ...). (שקפים 27-29).

* הוספת מילת מפתח super ל- IC. ומבור בנאים בלבד. כמו לגבי הפקודה הקודמת בלבד). ניתן להתייחס ל super כמילת שמורה נפרדת, או להתייחס אליו כקריאה לפונקציה ורק בשלב הסמנטי להתחיל להתייחס ליחיד של הפקודה. (שקפים 29 - 32).

שאלות נוספות הן משקפים 32.