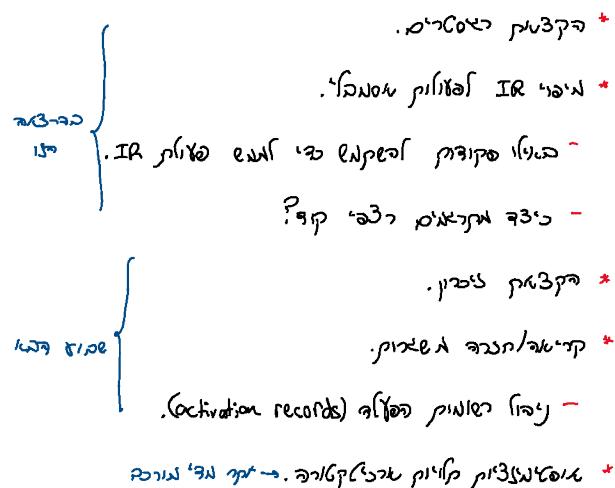


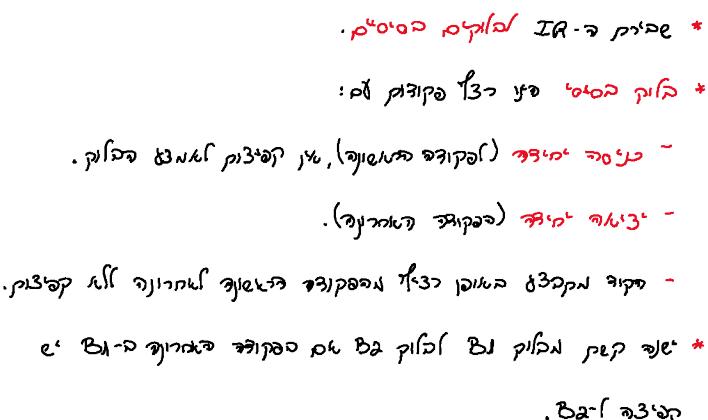
גרף בקורת הזירימה

12:18 AM Tuesday, October 6, 2020

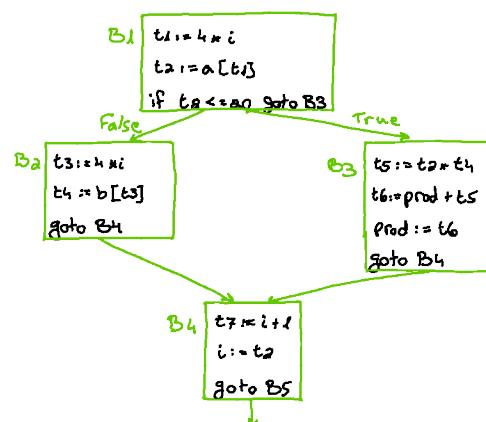
: מילוי ASM-ם IR-ן



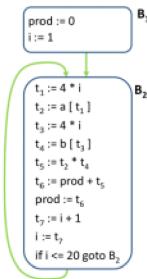
: Control Flow Graph - דיאגרם של זרימת הפקה



: ANTB - דיאגרם של זרימת הפקה



הנתקה ב-הנתקה



$G = \langle V, E \rangle$ אוסף מושגים *

ר' מילר ו- V -

ההכרה של הנתקה (ההכרה) E הנתקה של הנתקה E -

. בנות מונה B1-N

הנתקה של הנתקה

(איך מושג) three address statements ב-#37: Cfp *

איך מושג מ-ההכרה של הנתקה של הנתקה E : Cfp *

לעומת מושג של הנתקה של הנתקה

: OpenRef *

(מונח מושג של הנתקה של הנתקה) leaders → מושג של הנתקה של הנתקה

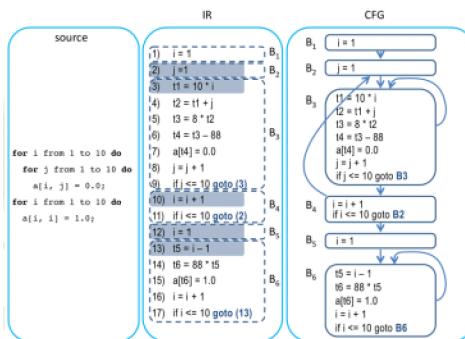
. leader מושג של הנתקה של הנתקה

. leader מושג של הנתקה של הנתקה

. leader מושג של הנתקה של הנתקה

איך מושג של הנתקה של הנתקה של הנתקה של הנתקה leaders מ-'

. מושג של הנתקה של הנתקה leaders מ-'



Simple Code Generation

11:03 AM Sunday, May 10, 2020

Simple Code Generation

* סעיפים

- נאנו, כוונתך מילא בפניהם.

- עיר גוף ערך ורשותה של הערך מחרט.

- ריבועים (1,3,5) ערך כהן מילא בפניהם.

ריבועים מילא בפניהם.

. (השורה הראשונה היא שורה של הערך מילא בפניהם.)

* נאנו, כוונתך מילא בפניהם.

* ריבועים מילא בפניהם.

לודג'ר מילא בפניהם LD reg,mem

סט מילא בפניהם ST mem,reg

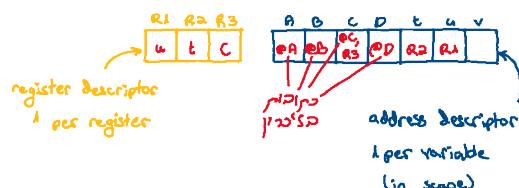
OPER מילא בפניהם OP reg,reg,reg

. ריבועים מילא בפניהם.

(EBP, EBP) מילא בפניהם.

לפניהם מילא בפניהם. גולן מילא בפניהם. גולן מילא בפניהם.

: Descriptors → הוכן



. הוכן מילא בפניהם register descriptor *

. מילא בפניהם register descriptor.

. מילא בפניהם register descriptor.

. מילא בפניהם register descriptor.

* מילא בפניהם register descriptor.

מילא בפניהם register descriptor.

. מילא בפניהם register descriptor.

. מילא בפניהם register descriptor.

הנחתה נאכ"ל

: $x := y \oplus 2^k : 3AC$ ימינה ב+

R_x, R_y, R_z מכוון נאכ'ן נאכ'ן $x := y \oplus 2^k$ getreg $(x := y \oplus 2^k)$ - 1 107

$y \leftarrow @y$ מכוון נאכ'ן LO $R_y, @y : 13D$, y מכוון יי' R_y מכוון *

$.2 \leftarrow @2$ מכוון נאכ'ן LO $R_z, @2 : 13D$, 2 מכוון יי' R_z מכוון *

$.OP R_x, R_y, R_z := 3AC$ ימינה 3AC *

הנחתה שמיינט פונק'ון מילוי הינה נאכ'ן נאכ'ן getreg יתנו *

הנחתה נאכ'ן נאכ'ן *

: $3AC$ נאכ'ן נאכ'ן *

$d[2^k], R_x$ מכוון נאכ'ן נאכ'ן address descriptor מ' x ימינה GR -

$x \leftarrow @x$ מכוון נאכ'ן ST $@x, R_x$

הנחתה כוונתית מילוי פונק'ון מילוי מילוי מילוי מילוי מילוי *

: Descriptors \rightarrow LOD

$\begin{cases} d[R] = R \text{ register descriptor} & : LOD R, @x \text{ load ימינה נאכ'ן} \\ d[x] = x \text{ address descriptor} & d[R] \leftarrow \{x\} (x, p) \end{cases}$ *

$d[x] \leftarrow d[x] \cup \{R\} (R_{\text{מעורב}})$

$d[y] \leftarrow d[y] / \{R\} y \neq x ימינה GR$

: ST $@x, R$ store ימינה נאכ'ן *

$d[x] \leftarrow d[x] \cup \{R\} (x, p)$

$x \leftarrow p$ מילוי מילוי מילוי מילוי store מילוי מילוי מילוי *

: OP R_x, R_y, R_z נאכ'ן נאכ'ן *

$x := y \oplus 2^k : 3AC$ ימינה נאכ'ן *

$d[R_x] \leftarrow \{x\}$ ימינה נאכ'ן מילוי *

$d[x] \leftarrow \{R_x\}$ ימינה נאכ'ן מילוי מילוי מילוי מילוי *

$d[y] \leftarrow d[y] / \{R_x\} y \neq x ימינה GR$

1 (6) : $x := y$ copy statement ימינה *

copy descriptors \rightarrow ימינה LO $R_y, @y : 3D$ -

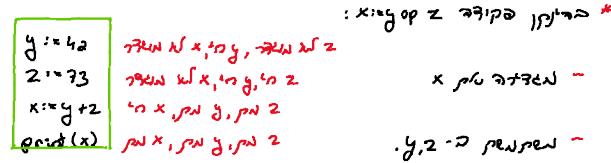
$d[R_y] \leftarrow d[R_y] \cup \{x\}$

$d[x] \leftarrow \{R_y\}$

Variable Liveness

12:18 AM Tuesday, October 6, 2020

: Variable Liveness



* נסמן $x := y + 2$ כ- x live (live)

נניח, אין משתנים זרים, אך x עשוי להזמין y .

לiveness-ג

* אם, מרגע מסוים, משתנה x מוביל לירוי (data-flow) רצוף (data-flow).

* מרגע מוקדם יותר, מרגע:

- רצוף שלם יופיע ב- x .

- סיכון של רצוף מוביל לירוי.

- סיכון של רצוף מוביל לירוי.

* $\text{def} : \text{def}$ (def, def, B) ב- x יוביל לירוי.

- פירעון של רצוף מוביל לירוי.

* x, y, z ב-next-use- \rightarrow ב- $x = y + 2$ live-ness-ג, B -> $i := x = y + 2$ ב- i , C- x .

(i מוגדר ב- $x = y + 2$)

* מילויים:

- מילוי נפקודת המוגדר ב- B וולק מוביל לירוי.

: $i := x = y + 2$ מוביל לירוי.

* מילוי יפה $x = y + 2$ מוביל לירוי.

② מילוי יפה מוביל לירוי.

- מילוי מוביל לירוי.

. $i := x = y + 2$ מוביל לירוי.

* מילוי יפה מוביל לירוי?

- כו, ת' קאנט ופראט: $x = x$ מוביל לירוי.

ב- $x = y + 2$ מוביל לירוי.

הקצאת גניטרים

12:18 AM Tuesday, October 6, 2020

הקצאת CoCoR מקאלה:

* מילוי סכום, מילוי פולר, רכיבת מילוי נמוך.

* נספח טקסט:

- סכום ורכיבי סכום כמייצגם ב- CoCoR שמשתמש ב- Ry .

.load גודל.

- סכום ורכיבי סכום שמייצגם ב- CoCoR ב- Ry , מילוי נמוך ב- Ry .

.load גודל.

* מילוי מודולרי:

- גודל ערך מודולרי ב- CoCoR מילוי מודולרי.

* מילוי גודל:

- גודל סכום סכום, גודל סכום סכום.

. Ry גודל סכום סכום.

* ספוגה גודל:

- סכום ורכיבי סכום גודל ספוגה, גודל ספוגה.

(Ry מילוי מילוי).

- סכום ורכיבי סכום גודל ספוגה.

. Ry גודל ספוגה.

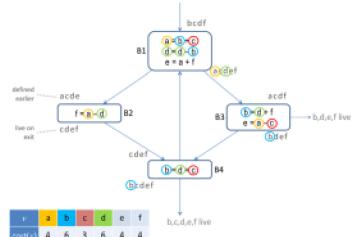
- סכום ורכיבי סכום גודל ספוגה.

. Ry גודל ספוגה.

* ספוגה: $\text{ST}@v.R$ - הטרנספורם ספוגה מילוי מילוי.

* ספוגה: ספוגה ספוגה ספוגה ספוגה ספוגה ספוגה.

* ספוגה ספוגה ספוגה ספוגה ספוגה ספוגה ספוגה.



לפניהם נסמן B_1 ו- B_2 כבlocs ב- B .
 נסמן v כvariable.
 נסמן $use(v, B)$ כ- v ה- use ב- B .
 נסמן $live(v, B)$ כ- v ה- $live$ ב- B .

(v ה- use ב- B מוגדר ב- B ו- $live$ ב- B מוגדר ב- B).

$$cost(v) = \sum_B use(v, B) + \alpha live(v, B)$$

* גורם אחד הוא גודל ה- B , גודל ה- B מוגדר כ- $\#B$.

* גורם שני הוא גודל ה- B , גודל ה- B מוגדר כ- $\#B$.

הוכחה של קיומו של אולטימום ל- $cost(v)$

* נסמן C כ- $cost(v)$ מינימלי ב- B .

- Liveness -

- כפוקטיביות של v -

- גודל B מוגדר:

- (*interference graph*) מוגדר ב- B ב- B -

- גודל B מוגדר ב- B -

- v מוגדר ב- B ו- v מוגדר ב- B -

- v מוגדר ב- B ו- v מוגדר ב- B -

* נסמן v כ- v מוגדר ב- B ו- v מוגדר ב- B !

- גודל B מוגדר:

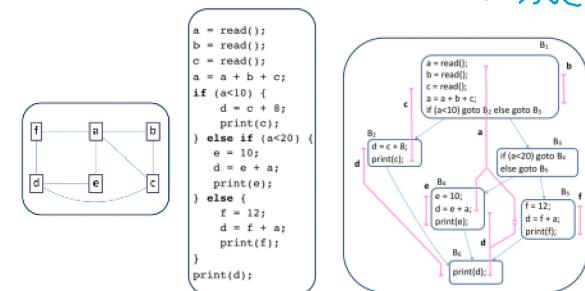
- גודל B מוגדר ב- B ו- v מוגדר ב- B -

- גודל B מוגדר ב- B ו- v מוגדר ב- B -

- גודל B מוגדר ב- B ו- v מוגדר ב- B -

- גודל B מוגדר ב- B ו- v מוגדר ב- B -

אולטימום ל- $cost(v)$



* נסמן v כ- v מוגדר ב- B ו- v מוגדר ב- B ו- v מוגדר ב- B ו- v מוגדר ב- B .

* נסמן v כ- v מוגדר ב- B ו- v מוגדר ב- B ו- v מוגדר ב- B .

* גודל B מוגדר ב- B ו- v מוגדר ב- B .

3. סעיפים באלגוריתם

* גזירה: גזירה מתרחשת סימetricamente בפונקציית פ"ח. דהיינו, רצף גזירתי.

* גזירתי פ"ח קיימ" - גזירתי פ"ח מוגדר כפונקציה (פונקציית גזירתי).

* גזירתי מוגדר (high level) גזירתי (low level).

- גזירתי מוגדר כפונקציה (low level).

- גזירתי מוגדר כפונקציית גזירתי.

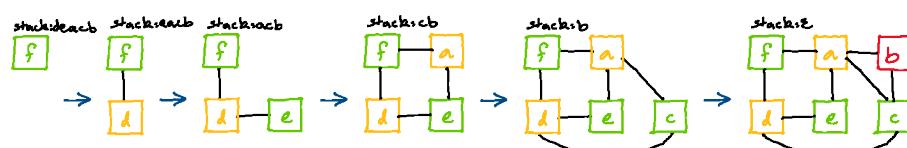
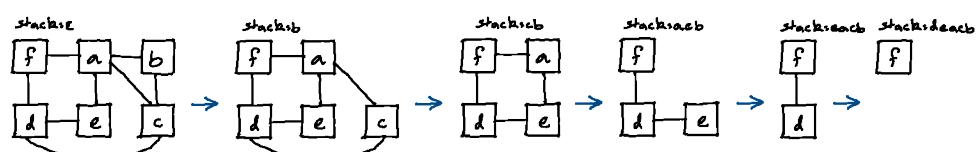
- גזירתי מוגדר כפונקציית גזירתי.

- גזירתי מוגדר כפונקציית גזירתי.

* גזירתי מוגדר כפונקציית גזירתי.

Ⓐ גזירתי מוגדר גזירתי נבנה מפונקציית גזירתי.

Ⓑ גזירתי מוגדר גזירתי נבנה מפונקציית גזירתי.



4. קומפלקסיטי א' ו' של אלגוריתם

* מה שטרם מודר שטח יוצר דיאגרמה שטח שטח שטח?

* מה שטרם מודר שטח שטח שטח שטח?

* מה שטרם מודר שטח?

אSEMBLY בארכיטקטורת IA-32

12:18 AM Tuesday, October 6, 2020

ՀԱՅՈՒԹ ՀԱՅԱ ԵՇ ԽԵՐԵՐԵՎԱՆ ՄԵՋԻ

Architecture	Year	Register size	GPR count	Comments
MOS 6502	1975	8 bit	2+1	X and Y are GPRs, A is "Accumulator"
Intel 8088	1979	16 bit	16	Registers are "more-or-less" general purpose
Intel IA-32	1985	32 bit	8	80386, 80486, Pentium, etc.
MIPS	1986	32 bit	31	R2000, etc., register \$0 hardwired to zero
SPARC V7	1986	32 bit	31	There was no V1...V6
IBM POWER	1990	64 bit	32	POWER1, POWER10
MIPS III	1990	64 bit	31	R4000, etc.
SPARC 64	1993	64 bit	31	SPARC V9 and later
AMD x86-64	1999	64 bit	16	AMD K7, Intel Core, Intel i3...i9, etc.
AArch64	2013	64 bit	32	ARM v8, Apple A4...A13, etc.

: Intel IA-32 32bit & Power

* నుండి యామన్‌కు గతిలు :

assembling) "now -

(linking) verb -

* AT&T 6 opCj40 enproj - Intel pN185 AT&T opCj40 *

: Intel PAR AT&T OPC) o

Intel	AT&T	גרסתך
בפ', נט'ן .db, .word	בפ', נט'ן .byte, .word	בפ' נט'ן .byte, .word
רשיון גודל .10 subl rax, rdx	רשיון גודל pushl %rdx subl %rax, %rdx	רשיון גודל pushl %rdx subl %rax, %rdx
	, byte - b)窘况 dword - l, word - w quad-word - q	, byte - b)窘况 dword - l, word - w quad-word - q
תפקידם נתקיים על ידי הפעלת מנגנון של הפעלה דינמית, דיאנה שום רשותה לא תאפשר, אך בודק אם	תפקידם נתקיים על ידי הפעלת מנגנון של הפעלה דינמית, דיאנה שום רשותה לא תאפשר, אך בודק אם	תפקידם - %
ל-3-ד כבויים + וואט + מולטיפלייר (word, byte) ל-2 גודל (dword)	:3(%bx,%cx,4) DISPLAY,INDEX,SCALE	כבר יופיע ל-3-ד כבויים
v eax, dword [ebx + ecx*4 + 1]	movl \$1(%ebx,%ecx,4), %eax	

IA-32-C 8048

• CP-3a ב מילויים נספחים פ*

תְּבִיבָה מִתְּבִיבָה תְּבִיבָה תְּבִיבָה -EX-

• కుటుంబ ఫీ పాని వ్యవస్థల కుటుంబ ఫీ పాని వ్యవస్థల -EBx-

• மாண்பும் வினாக்கள் முதல் எப்ப-ECX -

(השאלה ט' נסבנית) I/O נסבנית - EDx *

השאלה ט' נסבנית קיימת גם בEDI -

השאלה ט' נסבנית קיימת גם בESI -

השאלה ט' נסבנית קיימת גם בESP -

השאלה ט' נסבנית קיימת גם בEDI -

EFLAGS כוכב *

(Instruction Pointer) EIP כוכב *

.0x0-16 בזיהוי הכתובת 6 *

.(השאלה ט' נסבנית קיימת גם בEDI -

:�יעו בשיוף אוניברסיטה באליפות

EAX *

.DIV, IDIV, IMUL, MUL, מילוי הכתובת 6 -

.השאלה ט' נסבנית קיימת גם בEDI -

EDX *

.IDIV, DIV, DIV : מילוי הכתובת 6 -

.(השאלה ט' נסבנית קיימת גם בEDI -

ESI, EDI *

.השאלה ט' נסבנית קיימת גם בEDI -

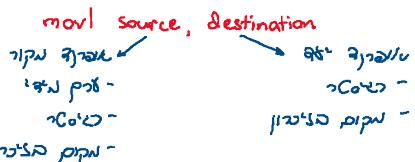
.השאלה ט' נסבנית קיימת גם בEDI -

.EDX, ECX, EBX, EAX, מילוי הכתובת 6 -

.push/pop הפעלת גנרטור נסבנית ESP *

:IA-32-ה כריסטוף נסבנת

.השאלה ט' נסבנית קיימת גם בEDI -



:השאלה ט' נסבנית קיימת גם בEDI -

לכטום נסבנת:

.השאלה ט' נסבנית קיימת גם בEDI -

.השאלה ט' נסבנית קיימת גם בEDI -

add \$4,%esp -

:LC0+4,*

.LC0+4* mem[esp+4] =

.LC0+4* mem[esp+4] = GAS opCj0 -

mov %esp,%ebp -

הקלות בדרכן כפ'ו:

* נסמן כיכר:

הציג כבוקס פולימר.

.LC0+4* GAS opCj0 -

mov (%eax),%eax -

: (Base Displacement) אונט גודל *

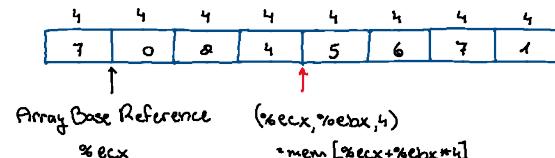
הציג ככידוד פולימר,

address = base + (offset * scale) + disp : כ הפוך מהתפקידים -

%eax+0 => mov \$4a,\$0(%eax)

%eax+4*%ecx+1 => mov \$4a,\$1(%eax,%ecx,4)

Base Displacement Address



base = %ecx
offset = %ebx
scale = 4
address = base + (index * scale) + displacement

address = %ecx + (3 * 4) + 0 = %ecx + 12

lea (local effective address) : כיריך אונט גודל *

lea (%ebx,%esi,8),%edi : *

%ebx+8*%esi

.95+> LC0+4* mem[esi+8*esi] = GAS opCj0 -

mov (%ebx,%esi,8),%edi : גודל *

%ebx+8*%esi mem[]

בחירה ההוראות IR

12:18 AM Tuesday, October 6, 2020

הנתק פוליה

* IR-הו מושג גוף אחד

* IR-הו מושג גוף אחד. IR-הו מושג גוף אחד.

3AC-הו → Peephole Matching -

תילינג → Tiling -

: Peephole Matching

* מושג גוף אחד מושג גוף אחד.

* מושג גוף אחד מושג גוף אחד.

* מושג גוף אחד מושג גוף אחד (peephole) ייפס סרף.

: פונקציית IR-הו *

$*p := R1$	$R7 := R2 + 0$	$L_{10}:$ goto L_{11}
$R15 := *p$	$R10 := R4 * R7$	$L_{11}:$ goto L_{12}
$*p := R1$	$R10 := R4 * R2$	$L_{12}:$ goto L_{13}
$R15 := R1$		
store followed by load	algebraic identities	jump to jump

? פוליה *

* מושג גוף אחד מושג גוף אחד מושג גוף אחד.



: Expander (3)

register-transfer
language

* RTL מושג Low-Level IR (LLIR) מושג IR מושג פורט -

* מושג גוף אחד מושג גוף אחד מושג גוף אחד.

* מושג גוף אחד מושג גוף אחד מושג גוף אחד.

* מושג גוף אחד מושג גוף אחד מושג גוף אחד.

: Simplifier (4)

Original IR Code

op	arg ₁	arg ₂	result
*	z	y	t ₁
-	x	t ₁	w

t₁ = t₁

Expand

LLIR Code

```

r10 := 2
r11 := @y
r12 := bp + r11
r13 := *(r12)
r14 := r10 * r13
r15 := @x
r16 := bp + r15
r17 := *(r16)
r18 := r17 - r14
r19 := @w
r20 := bp + r19
*(r20) := r18

```

Simplify

LLIR Code

```

r10 := 2
r11 := @y
r12 := bp + r11
r13 := *(r12)
r14 := r10 * r13
r15 := @x
r16 := bp + r15
r17 := *(r16)
r18 := r17 - r14
r19 := @w
r20 := bp + r19
*(r20) := r18

```

* מושג גוף אחד מושג גוף אחד מושג גוף אחד.

* dead-effect מושג גוף אחד.

* מושג גוף אחד מושג גוף אחד מושג גוף אחד.

: Matcher ②

LLIR Code

```
r13 := *(bp + @y)
r14 := 2 * r13
r17 := *(bp + @x)
r18 := r17 - r14
*(bp + @w) := r18
```

Match

Assembly (x86) Code

```
movl %y(%ebp), %esi
movl %esi, %edi
shll $1, %edi
movl %x(%ebp), %ecx
movl %ecx, %edx
subl %edi, %edx
movl %edx, %w(%ebp)
```

ריצוף פשוט של IR-> סעיפים

- מילוי סעיפים נקיים ומיוחדים

- מילוי סעיפים מילויים נקיים, LLVM לא מילויים מילויים

- מילוי סעיפים נקיים

: ת'זפ'ה ב-לינקו

$r_1 := c$ $r_2 := r_3 + r_1$ $c = \text{constant}$

$r_1 := r_2 + e$ $r_3 := *(r_1)$ $e = \text{constant or register}$

$r_1 := 2 * r_2$ $r_3 := r_4 - r_2 - r_1$

? ביצ' פה Peephole Matching-ו נון-טז'

RTL → IR מילויים נקיים

.LLIR→PSM מילויים נקיים מילויים נקיים

מילויים נקיים מילויים נקיים

JCC → יט'ם טריג'ר pattern-matchers → פונק'ן

(Very Portable Optimizer) VPO → LR אוניברסליים קבוצת מילויים נקיים מילויים נקיים

ללא כל מילויים נקיים מילויים נקיים מילויים נקיים

.LR אוניברסליים קבוצת מילויים נקיים מילויים נקיים מילויים נקיים

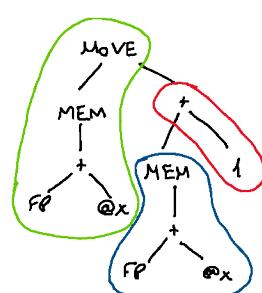
.low-level-ה מילויים נקיים מילויים נקיים מילויים נקיים מילויים נקיים

: Canonical IR-Tiling

מילויים IR-> סעיפים *

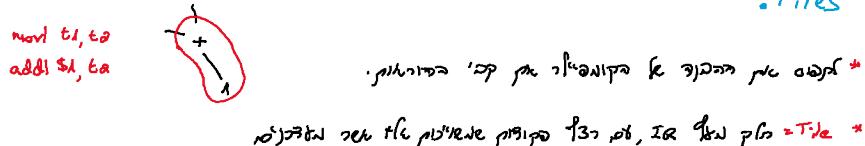
$x = x + 1;$

```
movl %x(%ebp), %eax
movl %eax, %ebx
addl $1, %ebx
movl %ebx, %x(%ebp)
```



מילויים IR-> סעיפים *

Tiles



* מטרת חישוב ה-IR היא לא לחלק את ה-IR לTiles.

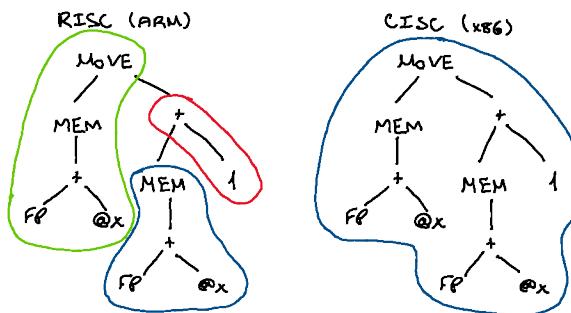
* (ולא בודק אם יש לנו יותר מ-231 מטרים).

* מטרת חישוב ה-IR היא לא לחלק את ה-IR לTiles.

* שפה נייטרלית מ-ARM.

* שפה נייטרלית מ-ARM.

CISC vs. RISC Tiling



move ex(%ebp),%eax
move %eax,%ebx
addl \$1,%ebx
move %ebx,@x(%ebp)

addl \$1,@x(%ebp)

* מטרת חישוב ה-IR היא לא לחלק את ה-IR לTiles.

* מטרת חישוב ה-IR היא לא לחלק את ה-IR לTiles.

* מטרת חישוב ה-IR היא לא לחלק את ה-IR לTiles.

Tiles → TBLR

* Maximal Munch - Minimal Register

* מטרת חישוב ה-IR היא לא לחלק את ה-IR לTiles.

* מטרת חישוב ה-IR היא לא לחלק את ה-IR לTiles.

* מטרת חישוב ה-IR היא לא לחלק את ה-IR לTiles.

* מטרת חישוב ה-IR היא לא לחלק את ה-IR לTiles.

* מטרת חישוב ה-IR היא לא לחלק את ה-IR לTiles.

Timing BIN

* מטרת חישוב ה-IR היא לא לחלק את ה-IR לTiles.

* מטרת חישוב ה-IR היא לא לחלק את ה-IR לTiles.

לע' נ' Colic Tiling

* מילוי: מילוי נון-ריבוט בפונקציית tiling.

* מילוי נון-ריבוט בפונקציית tiling.

* הור-ירפ ל.

* מילוי: מילוי נון-ריבוט בפונקציית tiling.

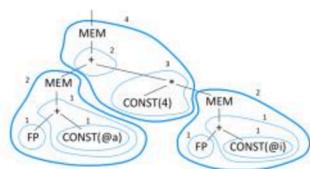
* הור-ירפ (diff-pp) מילוי נון-ריבוט בפונקציית tiling.

* הור-ירפ ל. מילוי נון-ריבוט בפונקציית tiling.

$$a[i:j] \quad cost = 2 + 2 + 4$$

* מילוי נון-ריבוט בפונקציית tiling.

! נ' נ' נ' נ'



```
movl @a(%ebp), %eax  
movl @i(%ebp), %ebx  
movl (%eax,%ebx,4), %ecx
```

O, CIO:

* מילוי נון-ריבוט בפונקציית tiling.

- מילוי נון-ריבוט בפונקציית tiling.

- מילוי נון-ריבוט בפונקציית tiling.

* מילוי נון-ריבוט בפונקציית tiling.

- ZAC בפונקציית Peephole Matching-ה.

- IQ->FLW בפונקציית tiling-ה.

- FLW->ZAC בפונקציית tiling-ה.

- timing fixups על מנת שפונקציית tiling תעבוד.

אנליזה סטטית

10:54 PM Wednesday, October 7, 2020

מבחן כוכב: CCO

* הינה בפונקציית f ש带回 x מילויים יפים.

מבחן כוכב (CopC) מודולרי.

* מבחן כוכב CCO מילויים יפים - מילויים יפים מילויים יפים מילויים יפים.

- נסמן מילויים יפים כטראם.

. IR-טראם, מילויים יפים.

? מבחן כוכב *

. מבחן כוכב (CopC) מילויים יפים.

area = width * height
p = 0
z = area + p

מבחן כוכב מילויים יפים.

.area מילויים יפים, p מילויים יפים, z מילויים יפים.

Record ad;
if (...) {
 ad.write();
}

לינק

. מבחן כוכב מילויים יפים.

```
x = ?  
if (x > 0) {  
    y = 42;  
} else {  
    y = 73;  
    foo();  
}  
assert (y == 42);
```

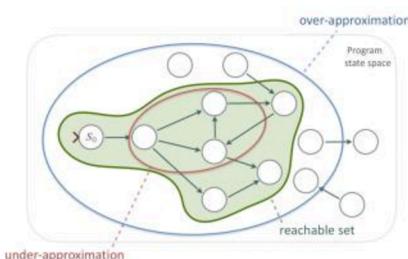
? מבחן כוכב מילויים יפים.

. מבחן כוכב מילויים יפים.

. מבחן כוכב מילויים יפים.

. assert == prove

* מבחן כוכב: רבודה קדום.



* מבחן כוכב מילויים יפים מילויים יפים.

. מבחן כוכב מילויים יפים מילויים יפים.

. מבחן כוכב מילויים יפים.

* מבחן כוכב מילויים יפים מילויים יפים מילויים יפים.

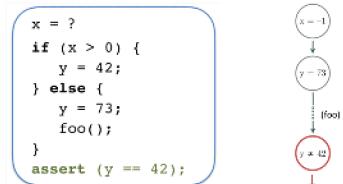
reachable set - היפוך קבוצת מושג-הישר. דהיינו, קבוצת כל המקומות שיכולים להגיע אליו.

היפוך קבוצת מושג-הישר היפוך קבוצת מושג-הישר.

backward approximation - מושג-הישר כה כיוון מוקדם מהווריאנטים מוקדמים.

forward approximation - מושג-הישר כה כיוון מאוחר מוקדמים מהווריאנטים מאוחרים.

ר' כוכב - ג'ס:



* מושג-הישר ג'ס מושג-הישר x ו- y ב-assert כ-bool.

- מושג-הישר ג'ס מושג-הישר x ו- y ב-assert.

* מושג-הישר ג'ס מושג-הישר assertion-היפוך מושג-הישר assertion-היפוך.

* מושג-הישר ג'ס מושג-הישר assertion-היפוך מושג-הישר assertion-היפוך.

ר' כוכב:

* מושג-הישר ג'ס מושג-הישר מושג-הישר.

- מושג-הישר ג'ס מושג-הישר מושג-הישר.

- מושג-הישר ג'ס מושג-הישר מושג-הישר.

ר' כוכב מושג-הישר:

* מושג-הישר ג'ס מושג-הישר מושג-הישר מושג-הישר מושג-הישר.

* מושג-הישר ג'ס מושג-הישר מושג-הישר מושג-הישר.

- מושג-הישר ג'ס מושג-הישר מושג-הישר מושג-הישר.

* מושג-הישר ג'ס מושג-הישר מושג-הישר מושג-הישר מושג-הישר.

* מושג-הישר ג'ס מושג-הישר מושג-הישר מושג-הישר מושג-הישר.

(functional verification).

- מושג-הישר ג'ס מושג-הישר מושג-הישר מושג-הישר מושג-הישר.

* מושג-הישר ג'ס מושג-הישר מושג-הישר.

- מושג-הישר ג'ס מושג-הישר מושג-הישר.

* דוג-ק'ריאט אוניברסיטט נסגר ונטה, כ' ודרם,

טנער פראג'ר נט'.

טְקִוָּה כְּלִילָה:

* טורם נט' - Dataflow analysis

* גדרת נט' - Constraint-based analysis
טְקִוָּה כְּלִילָה

* נט' - Type and effect Systems

* גדרת נט' - Abstract interpretation

... *