# COMPLETS

Dr. Carl Alphonce alphonce@buffalo.edu 343 Davis Hall



© 2021 Carl Alphonce - Reproduction



what does liveness and next use info looking like here?

t = a - b u = a - c v = t + u a = dd = v + u

Algorithm 8.7 [p. 528] Determining the liveness and next-use information for each statement in a basic block.

INPUT: A basic block B of three address instructions. Assume the symbol table initially shows all non-temporary variables in B as being live on exit. Not this instruction specifically, but instructions of the form

OUTPUT: At each statement i: x = y + z in B, we attach to i the liveness and next-use information for x, y, and z.

 $x = y \circ p z, x = \circ p y, or x = y.$ 

METHOD: We start at the last statement in B and scan backwards to the beginning of B. At each statement i: x = y + z in B do the following:

1) attach to statement i the information currently found in the symbol table regarding the next-use and liveness of x, y, and Z.

2) In the symbol table, set x to "not live" and "no next use".

3) In the symbol table, set y and z to "live" and the next uses of y and z to instruction i.

Next uses of y and z to instruction i.  $\odot$  2021 Carl Alphonce – Reproduction of this material is prohibited without the author's consent

## Example [p. 546]



INPUT: A basic block B of three address instructions. Assume the symbol table initially shows all non-temporary variables in B as being live on exit.

a	b	С	d	Ŀ	u	V
L	L	L	L			

### Example [p. 546]



We start at the last statement in B and scan backwards to the beginning of B. At each statement i:

x = y + zin B do the following:

 attach to statement i the information currently found in the symbol table regarding the next-use and liveness of x, y, and Z.
 In the symbol table, set x to "not

live" and "no next use".

3) In the symbol table, set y and z to "live" and the next uses of y and z to instruction i.

a	b	C	d	Ŀ	u	V
L	L	L	L			



We start at the last statement in B and scan backwards to the beginning of B. At each statement i:

x = y + zin B do the following:

 attach to statement i the information currently found in the symbol table regarding the next-use and liveness of x, y, and Z.
 In the symbol table, set x to "not live" and "no next use".

3) In the symbol table, set y and z to "live" and the next uses of y and z to instruction i.

a	b	С	d	Ŀ	u	V
L	L	L	D		L	L
					5	5

### Example [p. 546]



We start at the last statement in B and scan backwards to the beginning of B. At each statement i:

x = y + zin B do the following:

 attach to statement i the information currently found in the symbol table regarding the next-use and liveness of x, y, and Z,
 In the symbol table, set x to "not

live" and "no next use".

3) In the symbol table, set y and z to "live" and the next uses of y and z to instruction i.

a	b	C	d	Ŀ	u	V
L	L	L	D		L	L
					5	5



We start at the last statement in B and scan backwards to the beginning of B. At each statement i:

x = y + zin B do the following:

 attach to statement i the information currently found in the symbol table regarding the next-use and liveness of x, y, and Z.
 In the symbol table, set x to "not live" and "no next use".
 To the symbol table set y and z

3) In the sympol table, set y and z to "live" and the next uses of y and z to instruction i.

a	b	C	d	Ŀ	u	$\checkmark$
D	L	L	L		L	L
			4		5	5

					a	b	c	d	Ŀ	u	V
1:t		a		b							
2 <b>:</b> u	=	а	-	С							
3:v	=	t	+	U						L	2
4:a	=	d			L			D		-	
5:d	=	V	+	U				L			

We start at the last statement in B and scan backwards to the beginning of B. At each statement i:

x = y + zin B do the following:

 attach to statement i the information currently found in the symbol table regarding the next-use and liveness of x, y, and Z.
 In the symbol table, set x to "not

Live" and "no next use".

3) In the symbol table, set y and z to "live" and the next uses of y and z to instruction i.

a	b	C	d	u	V
D	L	L	L	L	L
			4	5	5

We start at the last statement in B and scan backwards to the beginning of B. At each statement i:

				a	b	C	d	Ŀ	u	V	
l:t =	a		b								
2:u =	a		С								
3:v =	t	+	U						L	L	
4:a =	d			L			D				
5:d =	V	+	U				L				

	×	=	y +	z			
in	B	do	the	foll	lowi	ng	•

 attach to statement i the information currently found in the symbol table regarding the next-use and liveness of x, y, and Z.
 In the symbol table, set x to "not live" and "no next use".
 In the symbol table, set y and z

to "live" and the next uses of y and z to instruction i.

a	b	C	d	Ŀ	u	V
D	L	L	L	L	L	D
			4	3	3	

					a	b	с	d	Ŀ	u	V
1:t :		a		b							
2 <b>:</b> u :	-	a		С	D		L			L 3	
3 <b>:</b> v:	=	t	+	U						L	L
4:a :		d			L			D			
5:d :		V	+	U				L			

We start at the last statement in B and scan backwards to the beginning of B. At each statement i:

x = y + zin B do the following:

1) attach to statement i the information currently found in the symbol table regarding the next-use and liveness of x, y, and Z.

2) In the symbol table, set x to "not live" and "no next use".

3) In the symbol table, set y and z to "live" and the next uses of y and z to instruction i.

a	b	C	d	Ŀ	u	V
D	L	L	L	L	L	D
			4	3	3	

				a	Ь	С	d	Ŀ	u	V	
1:t =	а		b								
2:u =	а	-	С	D		L			L		
3:v =	t	+	U						L	L	
4:a =	d			L			D		~	5	
5:d =	V	+	U				L				

We start at the last statement in B and scan backwards to the beginning of B. At each statement i:

x = y + zin B do the following:

 attach to statement i the information currently found in the symbol table regarding the next-use and liveness of x, y, and Z.
 In the symbol table, set x to "not live" and "no next use".
 To the symbol table, set y and z

to "live" and the next uses of y and z to instruction i.

a	b	С	d	Ŀ	u	V
L	L	L	L	L	D	D
2		2	4	3		

We start at the last statement in B and scan backwards to the beginning of B. At each statement i:

	a	b	С	d	Ŀ	u	V	
1:t = a - b	L 2	Ļ			L 3			
2:u = a - c	D		L			L 3		
3:v = t + u						L	L	
4:a = d	L			D				
5:d = v + u				L				

x = y + zin B do the following:

1) attach to statement i the information currently found in the symbol table regarding the next-use and liveness of x, y, and Z. 2) In the symbol table set x to "not

2) In the symbol table, set x to "not live" and "no next use".

3) In the symbol table, set y and z to "live" and the next uses of y and z to instruction i.

a	b	С	d		u	$\vee$
L	L	L	L	L	D	D
2		2	4	3		

#### Example [



We start at the last statement in B and scan backwards to the beginning of B. At each statement i:

					a	þ	с	d	Ŀ	u	V
L:t		а		b	L 2	L			L 3		
2 <b>:</b> u	=	а	-	С	D		L			L 3	
8:v	=	t	+	U						L	L
l:a	=	d			L			D			
5:d	=	V	+	U				L			

x = y + zin B do the following:

 attach to statement i the information currently found in the symbol table regarding the next-use and liveness of x, y, and Z.
 In the symbol table, set x to "not live" and "no next use".
 In the symbol table, set y and z

to "live" and the next uses of y and z to instruction i.

a	b	C	d	Ŀ	u	V
L	L	L	L	D	D	D
1	1	2	4			

Updating register descriptors (RD) and address descriptors (AD)

1. LD R, x

(a) set RD of R to only x

(b) Add R to AD of x

(c) Remove R from the AD of any variable other than x 2. ST x, R

(a) Add &x to AD of x

3. OP RX, RY, RZ for 
$$x = y \text{ op } z$$

(a) set RD of Rx to only x

(b) set AD of x to only Rx (&x not in AD of x !)

(c) Remove Rx from the AD of any variable other than x 4. "When we process a copy statement x = y, after generating the Load for y into register Ry, if needed, and after managing descriptors as for all Load statement (per rule 1):" [p. 545]

(a) Add x to the RD of Ry (b) set AD of x to only Ry

R1	R2	R3	a	b	C	d	Ŀ	u	V

Register descriptor

Assume just 3 registers are available for the sake of this example.

#### Address descriptor

Variables t, u, and v are compilergenerated temporary variables.

$$t = a - b$$
  
 $u = a - c$   
 $v = t + u$   
 $a = d$   
 $d = v + u$ 

R1	R2	R3	a	b	С	d	Ŀ	u	V
			a	Ь	C	d			

$$t = a - b$$
  
 $u = a - c$   
 $v = t + u$   
 $a = d$   
 $d = v + u$ 

At start of block, assume the values of variables a, b, c, and d are in main memory.

Variables t, u, and v are compilergenerated temporary variables.

R1	R2	R3	a	b	С	d	Ŀ	u	V
			a	Ь	С	d			

t = a - b

LD R1, a LD R2, b SUB R2, R1, R2

	a	b	С	d	Ŀ	u	V
1:t = a - b	L 2	L			L		
2:u = a - c	D		L			L 3	
3:v = t + u						L	L
4:a = d	L			D			
5:d = v + u				L			

R1	R2	R3	a	b	с	d	Ŀ	u	V
			a	Ь	с	d			

t = a - b

LD R1, a LD R2, b SUB R2, R1, R2

- 1. LD R, x
  - (a) Set RD of R to only x
  - (b) Add R to AD of x
  - (c) Remove R from the AD of any var other than x
- 2. ST x, R
  - (a) Add &x to AD of x
- 3. OP Rx, Ry, Rz for x = y op z
  - (a) Set RD of Rx to only x
  - (b) Set AD of x to only Rx (&x not in AD of x !)
  - (c) Remove Rx from the AD of any var other than x
- 4. x = y (after generating LD Ry y (if needed) and using rule 1) (a) Add x to the RD of Ry
  - (b) Set AD of x to only Ry



R1	R2	R3	a	þ	C	d	Ŀ	u	V	
			a	þ	С	d				
t = a	– b		LD R1, LD R2, SUB R2	a b 2, R1,	R2					
R1	R2	R3	a	b	c	d	Ŀ	u	V	
a	Ŀ		a, R1	b	С	d	R2			
<ul> <li>No registers are in use - pick the first two available for a and b.</li> <li>Choose to put t in R2 because b is not used again in this block.</li> <li>1. LD R, x <ul> <li>(a) Set RD of R to only x</li> <li>(b) Add R to AD of x</li> <li>(c) Remove R from the AD of any var other than x</li> </ul> </li> <li>1. LD R, x <ul> <li>(a) Set RD of R to only x</li> <li>(b) Add R to AD of x</li> <li>(c) Remove R from the AD of any var other than x</li> </ul> </li> <li>2. ST x, R <ul> <li>(a) Add &amp;x to AD of x</li> <li>(b) Set AD of Rx to only x</li> <li>(c) Remove R from the AD of any var other than x</li> </ul> </li> </ul> <li>4. x = y (after generating LD Ry y (if needed) and using rule 1 <ul> <li>(a) Add x to the RD of Ry</li> <li>(b) Set AD of x to only Ry</li> </ul> </li>										
				1:1	t = a	– b	L 2	L		

R1	R2	R3	a	Ь	c	d	Ŀ	u	V
			a	b	C	d			
t = a	– b		LD R1, LD R2, SUB R2	a b , R1,	R2				
R1	R2	R3	a	b	с	d	Ŀ	u	V
a	Ŀ		a, R1	b	с	d	R2		
u = a	- C		LD R3, SUB R1	, c , R1,	R3 2. 51 3. 0F 4. x	<pre>D R, x (a) Set RD of R (b) Add R to AD (c) Remove R fro f x, R (a) Add &amp;x to AI P Rx, Ry, Rz for (a) Set RD of Rx (b) Set AD of x (c) Remove Rx fr = y (after gene (a) Add x to the (b) Set AD of x</pre>	to only x of x om the AD of a O of x r x = y op z k to only x to only Rx (& rom the AD of erating LD Ry e RD of Ry to only Ry	ny var other tha ox not in AD of x any var other th y (if needed) and	n x !) an x d using rule 1)
							a b	c d t	u v
				2 <b>:</b> ι	<b>J = a</b>	— C	D	L	L 3

R1	R2	R3	a	b	С	d	Ŀ		u		V	
<b>_</b>		And the second se	a is already in R1, so no load									
τ = a	- D	<b>a</b> r.	heeded. t is used later, so don't overwrite R2. Load c into R3.									
R1	R2 Put result into R1 since a is not needed again											
a	Ŀ		9	in	this Dlo	ck.						
u = a	- C		LD R3, SUB R1	, c L, R1,	R3							
R1	R2	R3	a	Ь	С	d	Ŀ		U,		V	
u	Ŀ	С	a	Ь	C, R3	d	R2		R1			
1. LD R, x (a) Set RD ( (b) Add R to (c) Remove I	of R to only x o AD of x R from the AD o	of any var other	than x				a b	c	d	ji ka	u	V
<ol> <li>ST x, R         <ul> <li>(a) Add &amp;x <sup>-</sup></li> <li>OP Rx, Ry, R:</li></ul></li></ol>	to AD of x z for x = y op of Rx to only >	Z	5 1	2:	u = a	– C	D	L			L	
(b) Set AD (c) Remove I 4. x = y (after (a) Add x to (b) Set AD	Rx to only R Rx from the AD generating LD o the RD of Ry of x to only <u>Ry</u>	(&x not in AD o of any var other Ry y (if needed) /	than x and using rule :	1) this ma	aterial is p	rohibited	withou	it the	auth	vor's	3 con	sent

Г

R1	R2	R3	a	Ь	с	d	Ŀ	u	V
			a	b	с	d			
t = a	- b		LD R1, LD R2, SUB R2	a b , R1,	R2				
R1	R2	R3	a	b	C	d	Ŀ	u	V
a	Ŀ		a, R1	Ь	C	d	R2		
u = a	- C		LD R3, SUB R1	с , R1,	R3				
R1	R2	R3	a	Ь	C	d	Ŀ	u	$\checkmark$
u	ŀ	С	a	b	c, R3	d	R2	R1	
v = t	+ u		ADD R3	, R2,	R1		a b e	c d ł	u v
				3:	v = t	+ U			L L 5 5

R1	R2	R3	a	b	с	d	Ŀ	u	V	
			a	Ь	с	d				
t = a	– b		LD R1 LD R2 SUB R2	, a , b 2, R1,	R2					
R1	R2	R3	Mertilite Containing and Containing					UL	V	
a	Ŀ	tana	d u are	already	y in regi	isters -	no loo	ds		
u = a	u = a - Perform addition, putting the result into R3; c is no									
R1	R2	Kar						u	V	
u	Ŀ	c	a	Ъ	c, R3	d	R2	R1		
v = t	v = t + u ADD R3, R2, R1									
R1	R2	R3	a	Ь	C	d	Ŀ	u	V	
u	Ŀ	V	a	Ь	с	d	R2	R1	R3	

R1	R2	R3	a	b	C	d	Ŀ	u	V
u	Ŀ	$\checkmark$	a	b	с	d	R2	R1	R3

3:

rule 1)

Same state as at end of previous slide

1.	LD R, X
	(a) Set RD of R to only x
	(b) Add R to AD of x
	(c) Remove R from the AD of any var other than x
2.	ST x, R
	(a) Add &x to AD of x
3.	OP Rx, Ry, Rz for $x = y$ op z
	(a) Set RD of Rx to only x
	(b) Set AD of x to only Rx (&x not in AD of x !)
	(c) Remove Rx from the AD of any var other than x
4.	x = y (after generating LD Ry y (if needed) and using
	(a) Add x to the RD of Ry
	(a) Fact $AD$ of x to only $By$
	(b) Set AD OT X to only Ky

	a	b	С	d	Ŀ	u	V
v = + + u						L	L
						6	5

this material is prohibited without the author's consent

R1	R2	R3	a	b	c	d	Ŀ	ч	V
u	Ŀ	V	a	b	С	d	R2	R1	R3

a = d LD R2, d



this material is prohibited without the author's consent

R1	R2	R3	a	b	с	d	Ŀ		u	V	g
u	Ŀ	$\checkmark$	a	b	C	d	R2		R1	R	3
a = d			LD R2,	, d							
R1	R2	R3	a	Ь	С	d	Ŀ		u	V	9
u	a,d	V	R2	b	С	d,R2			R1	R	3
						K US WE					
1. LD R, x (a) Set RD (b) Add R t	of R to only x o AD of x						a b	с	d	t u	$\checkmark$
(c) Remove 2. ST x, R (a) Add &x 3. OP Rx, Ry, R	R from the AD o to AD of x z for $x = y$ op	t any var other t z	:han x	4 • 2	a = 0		1		79		
(b) Set AD (c) Remove 4. x = y (after (a) Add x t	<pre>(a) Set AD of x to only Rx (&amp;x not in AD of x !) (c) Remove Rx from the AD of any var other than x 4. x = y (after generating LD Ry y (if needed) and using rule 1) (a) Add x to the RD of Ry</pre>										
(b) Set AD	of x to only Ry			CRUD PRIM		municen	with the	. Crie	MULINO		SERVE

R1	R2	R3	a	b	С	d	Ŀ	u	V
u	Ŀ	V	a	b	с	d	R2	R1	R3
a = d			LD R2,	d					
R1	R2	R3	a	Ь	С	d	Ŀ	u	V
u	a,d	V	R2	b	с	d,R2		R1	R3
d = v	+ U		ADD R1	., R3,	R1				

1	. LD R, x
	(a) Set RD of R to only x
	(b) Add R to AD of x
	(c) Remove R from the AD of any var other than x
2	. ST x, R
	(a) Add &x to AD of x
3	. OP Rx, Ry, Rz for $x = y$ op z
	(a) Set RD of Rx to only x
	<pre>(b) Set AD of x to only Rx (&amp;x not in AD of x !)</pre>
	(c) Remove Rx from the AD of any var other than x
4	<pre>. x = y (after generating LD Ry y (if needed) and using rule 1)</pre>
	(a) Add x to the RD of Ry
	(b) Set AD of x to only Ry

a b c d t u v 5:d = v + u  $\downarrow$  L

this material is prohibited without the author's consent

R1	R2	R3	a	b	C	d	Ŀ	u	V		
u	Ŀ	V	a	þ	C	d	R2	R1	R3		
a = d			LD R2,	, d							
R1	R2	R3	a	þ	с	d	Ŀ	u	V		
u	a,d	V	R2	þ	C	d,R2		R1	R3		
d = v + u ADD R1, R3, R1 $(a = v + u)$											
R1	R2	R3	a	þ	с	d	Ŀ	u	✓		
d	a	V	R2	þ	С	R1			R3		
	u and v are in										
Canv	registers, so no loads needed. Cannot destroy a (exists only in R2) without a b c d t u v										
storu	storing back to memory, so use R1 for result. Move d to R1 from R2.										

© 2021 Carl Alpha

ertal is prohibited without the author's consent

R1	R2	R3	a	Ь	с	d	Ŀ	u	V	
u	Ŀ	V	a	Ь	C	d	R2	R1	R3	
a = d LD R2, d										
R1	R2	R3	a	Ь	с	d	Ŀ	u	V	
u	a,d	V	R2	Ь	с	d,R2		R1	R3	
d = v + u ADD R1, R3, R1										
R1	R2	R3	a	Ь	с	d	Ŀ	u	V	
d	a	V	R2	Ь	C	R1			R3	
exit			ST a, ST d,	R2 R1						

R1	R2	R3	a	b	¢	d	Ŀ	u	V			
u	Ŀ	V	a	b	С	d	R2	R1	R3			
a = d												
values of R1 and R2 are stored back to memory (d and a												
respectively). Value of R3 can be lost - it is a temporary of R3 only this block.												
d = v + u												
R1	R2	R3	a	b	c	d	Ŀ	U	∕ ✓			
d	a	V	R2	Ь	с	R1			R3			
exit ST a, R2 ST d, R1												
R1	R2	R3	a	b	с	d	Ŀ	u	V			
d	a	✓	a,R2	b	C	d,R1			R3			

© 2021 Carl Alphonce - Reproduction of this material is prohibited without the author's consent