Exercise

Code under analysis

```
char FixFilter(int L, char *E){
  int sum = 0;
  int length = 1 << L; // computes 2^L
  for (i = 0; i < length; i++){
    sum = sum + E[i];  // computes the sum of samples
  }
  sum = sum >> L; // computes the average = divides the sum by 2^L
  return sum;
}
```

Assumption: L <= 8

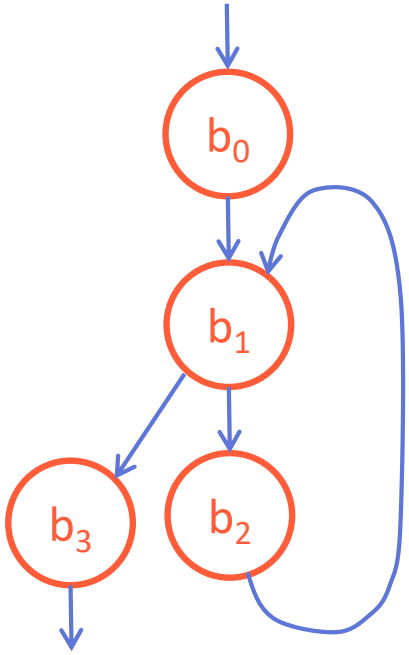
Code under analysis

```
char FixFilter(int L, char *E){
  int sum = 0;
  int length = 1 << L;
  for (i = 0; i < length; i++){
    sum = sum + E[i];
  }
  sum = sum >> L;
  return sum;
}
```

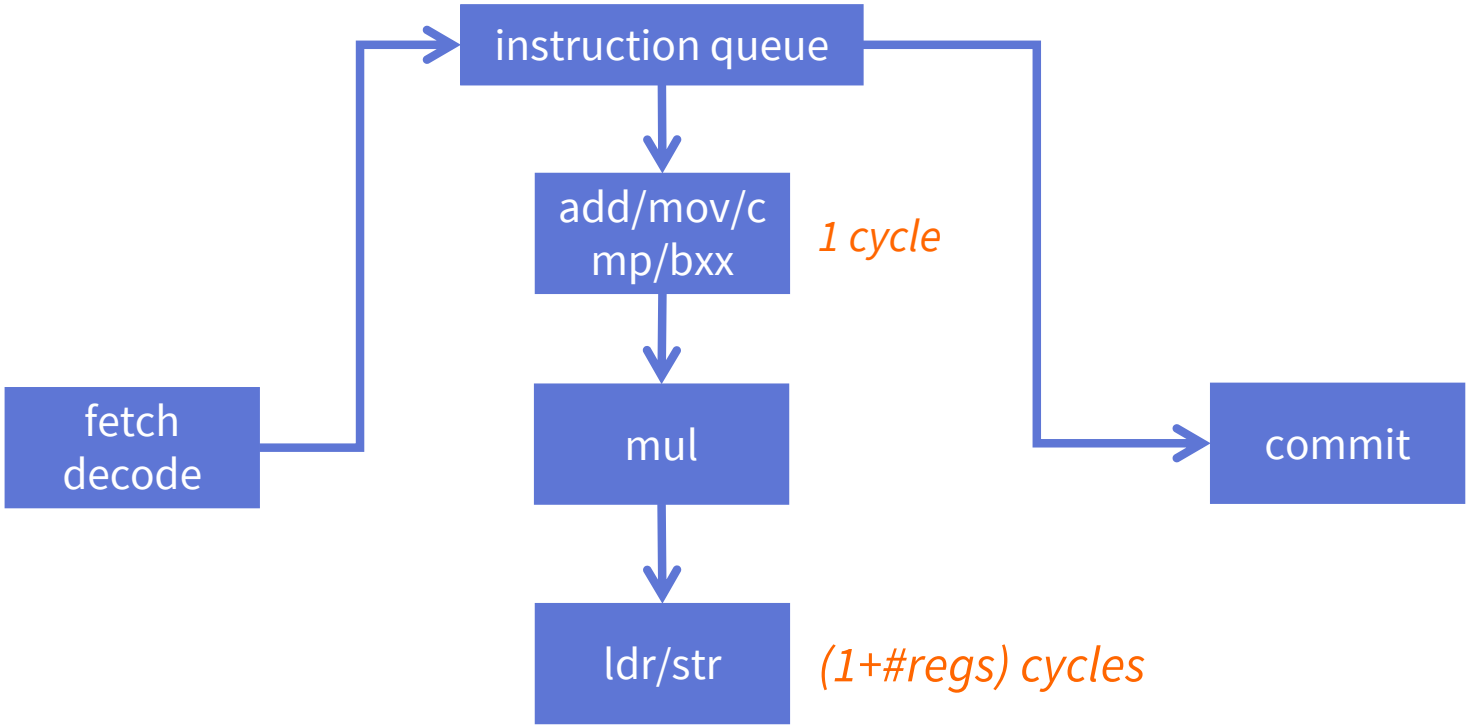
```
fixFilter:
      stmfd r13!,{r3-r6,r14}  (a)   // writes 5 registers
      mov   r3,#1             (b)
      mov   r3,r3, lsl r0     (c)
      mov   r6,r0             (d)
      mov   r0,#0             (e)
      mov   r4,#0             (f)
for_filter:
      cmp   r4,r3             (g)
      bcs   end_filter        (h)
      ldrb  r5,[r1,r4]        (i)   // reads one register
      add   r0,r0,r5          (j)
      add   r4,r4,#1          (k)
      b     for_filter        (l)
end_filter:
      mov   r0,r0,lsr r6      (m)
      ldmfd r13!,{r3-r6,r14}  (n)   // reads 5 registers
```

Build the CFG

```
fixFilter:
      stmfd r13!,{r3-r6,r14}  (a)
      mov   r3,#1             (b)
      mov   r3,r3, lsl r0     (c)   b0
      mov   r6,r0             (d)
      mov   r0,#0             (e)
      mov   r4,#0             (f)
for_filter:
      cmp   r4,r3             (g)   b1
      bcs   end_filter        (h)
      ldrb  r5,[r1,r4]        (i)
      add   r0,r0,r5          (j)
      add   r4,r4,#1          (k)   b2
      b     for_filter        (l)
end_filter:
      mov   r0,r0,lsr r6      (m)
      ldmfd r13!,{r3-r6,r14}  (n)   b3
```



Pipeline model



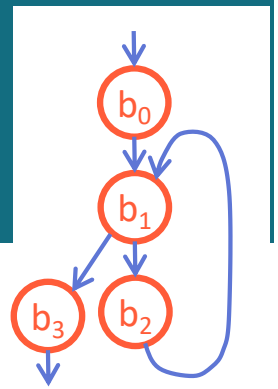
one instruction fetched each cycle (in the program order)

one instruction issued each cycle (in the program order)

one instruction committed each cycle (in the program order)

branches are predicted "not taken"

Execution cost of block b_1




```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)
      ldmfd   r13!,{r3-r6,r14}    (n)   b3
```

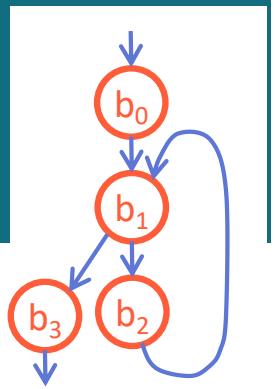
b_1 after b_0

F																		
A																		
Mu																		
Me																		
C																		

b_1 after b_2

F																		
A																		
Mu																		
Me																		
C																		

Execution cost of block b_1




```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```

t_0 = 13 cycles $t_{0\text{-}1}$ = 2 cycles

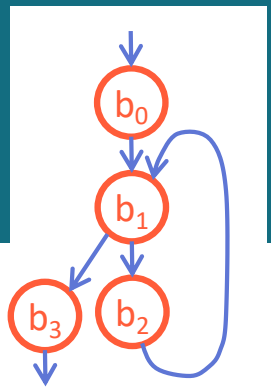
b_1 after b_0

F	a	b	c	d	e	f	g	h										
A			b	c	d	e	f	g	h									
Mu																		
Me		a	a	a	a	a	a											
C								a	b	c	d	e	f	g	h			

b_1 after b_2

F																		
A																		
Mu																		
Me																		
C																		

Execution cost of block b_1




```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```

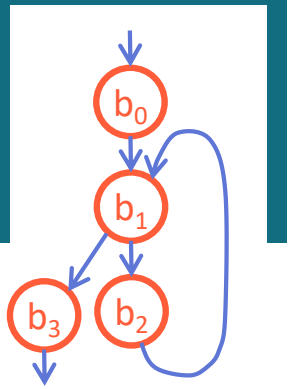
b_1 after b_0 — t_0 = 13 cycles — $t_{0\text{-}1}$ = 2 cycles

F	a	b	c	d	e	f	g	h										
A			b	c	d	e	f	g	h									
Mu																		
Me		a	a	a	a	a	a											
C								a	b	c	d	e	f	g	h			

b_1 after b_2 — $t_{2\text{-}1}$ = 3 cycles

F	i	j	k	l	m	n	g	h										
A				j	k	l		g	h									
Mu																		
Me		i	i															
C				i	j	k	l		g	h								

Execution cost of block b_2

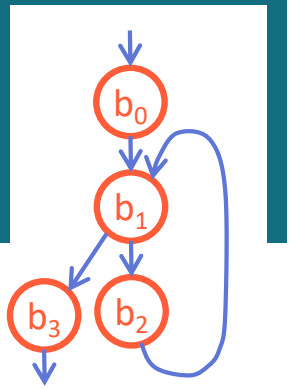



```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```

b_2 after b_1

F																		
A																		
Mu																		
Me																		
C																		

Execution cost of block b_2



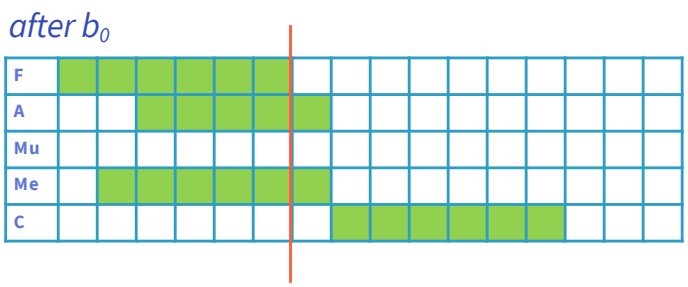

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```

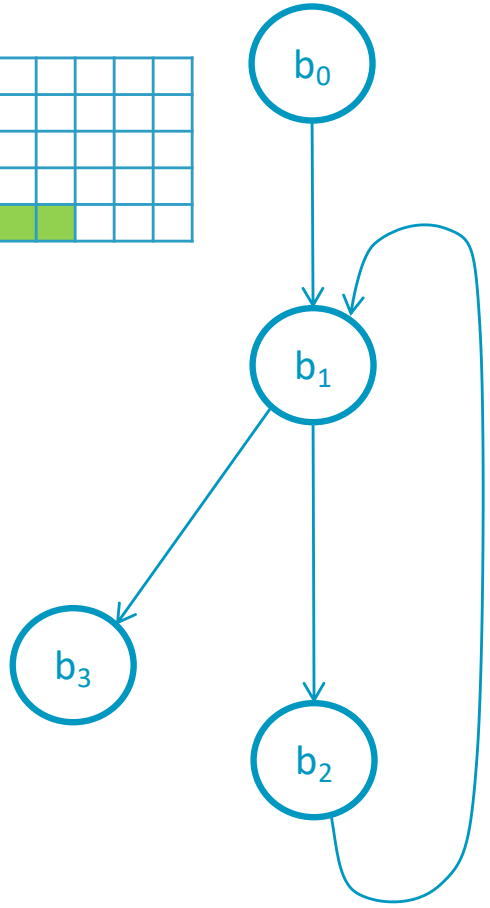
b_2 after b_1 — $t_{1\text{-}2}$ = 5 cycles

F	g	h	i	j	k	l												
A		g	h															
Mu						j	k	l										
Me				i	i													
C			g	h		i	j	k	l									

Using abstract interpretation

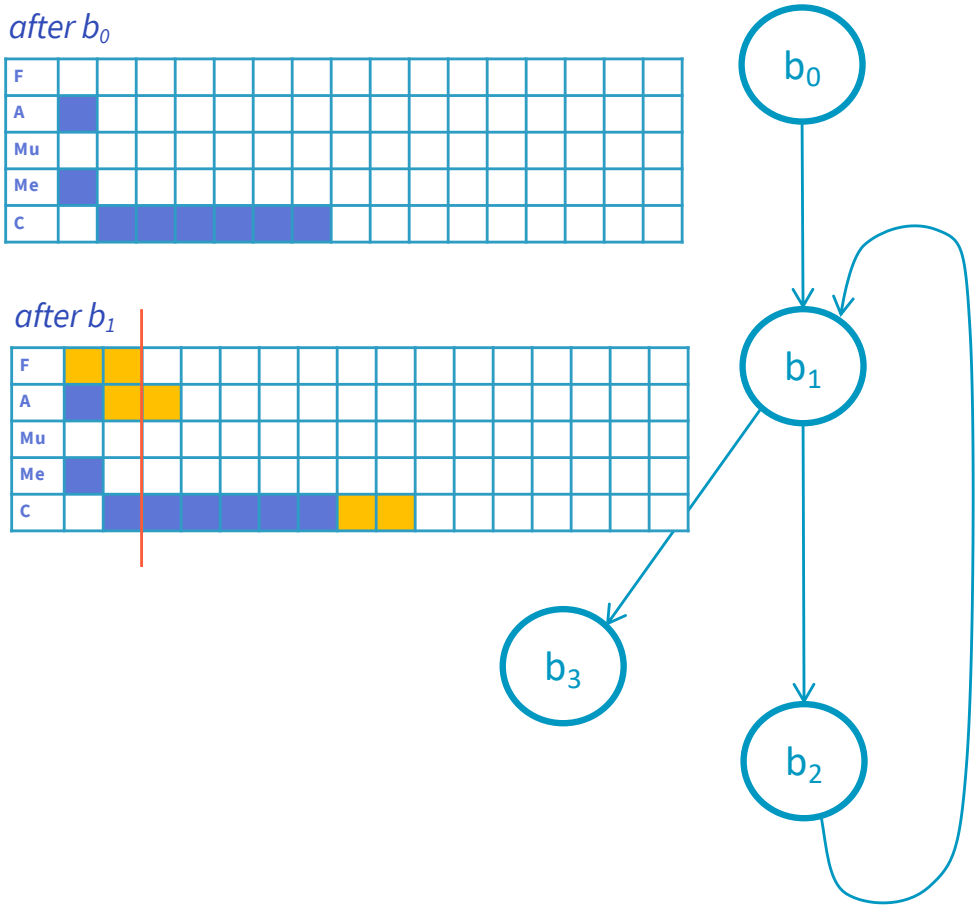
```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```





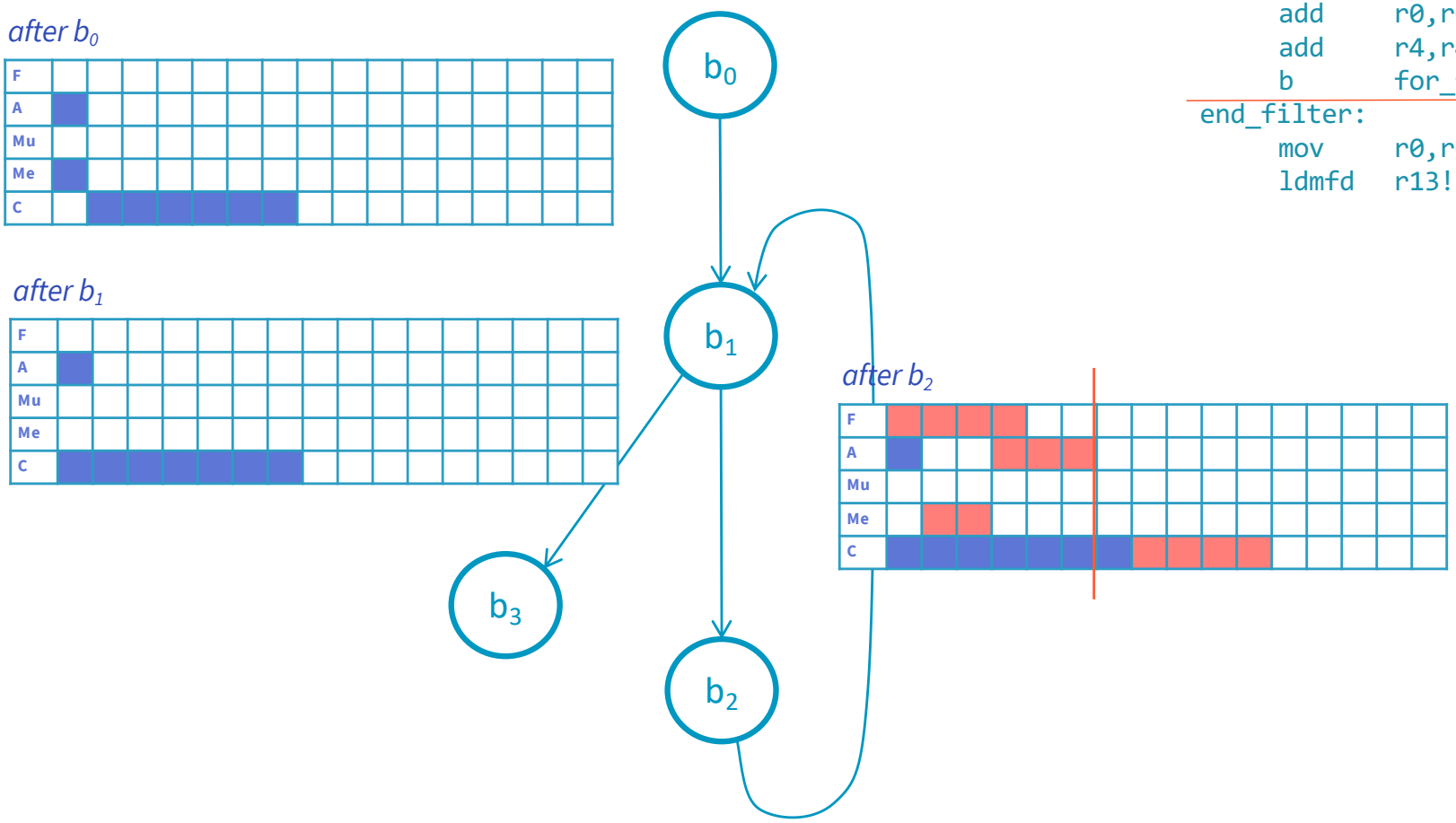
Using abstract interpretation

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```



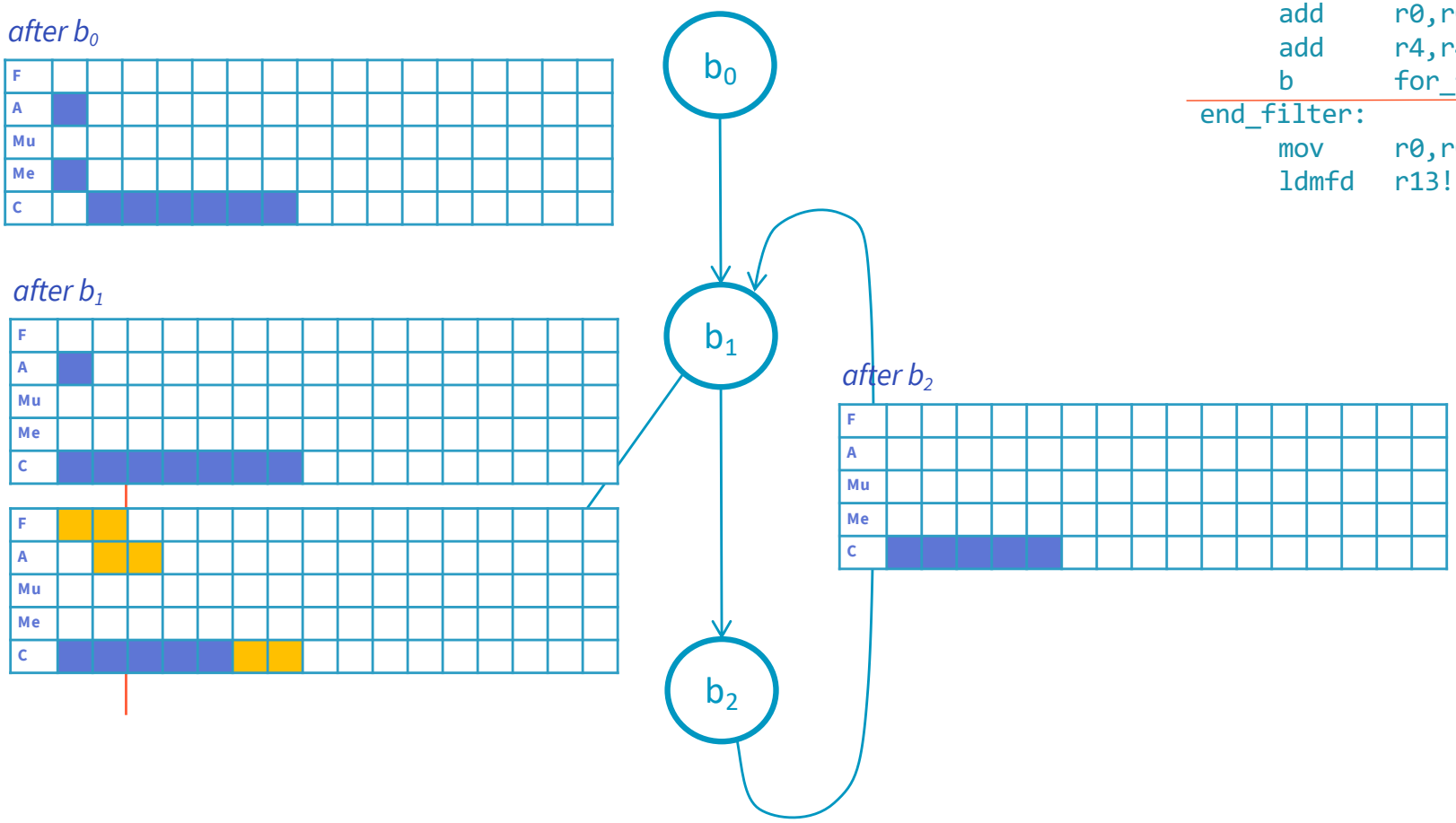
Using abstract interpretation

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```



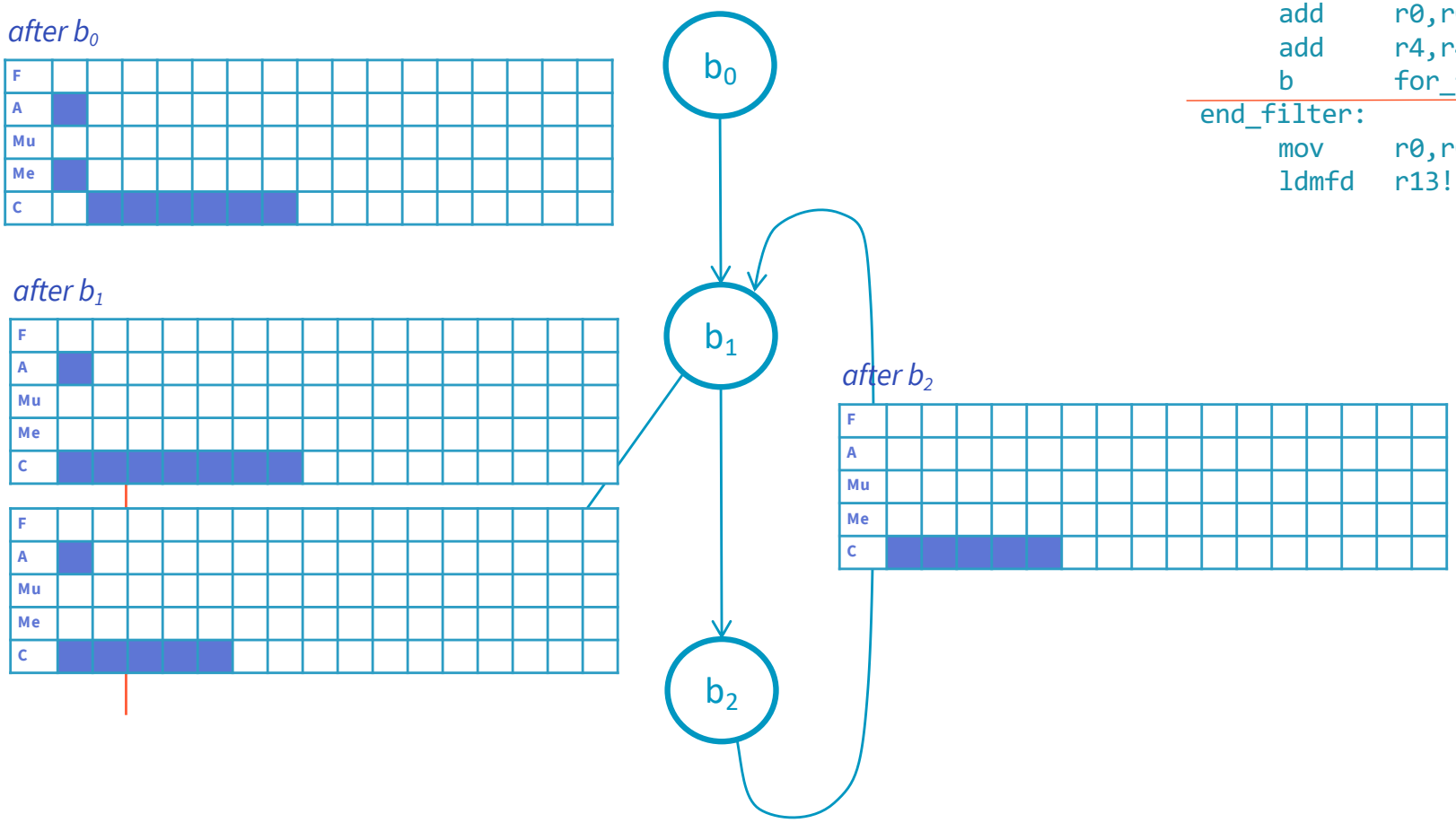
Using abstract interpretation

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```



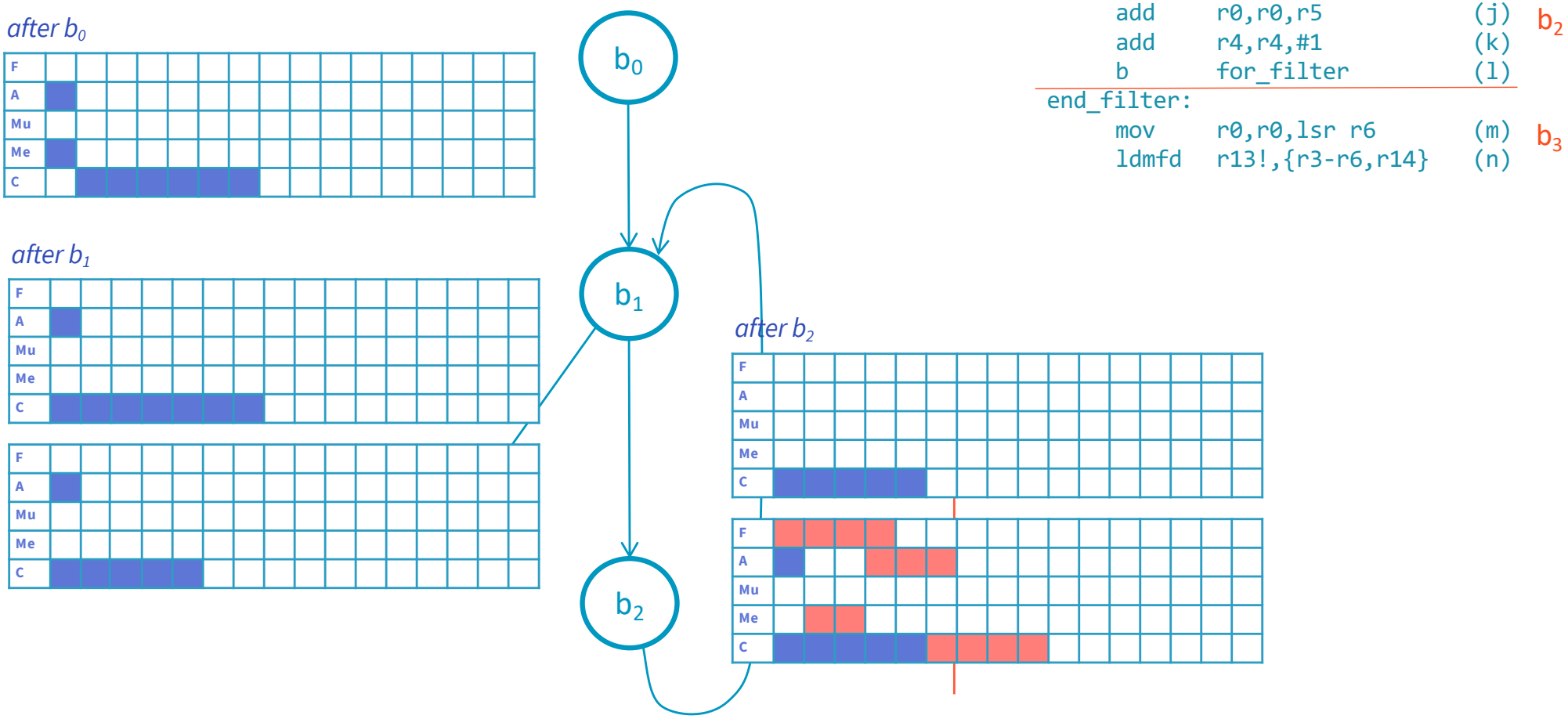
Using abstract interpretation

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```



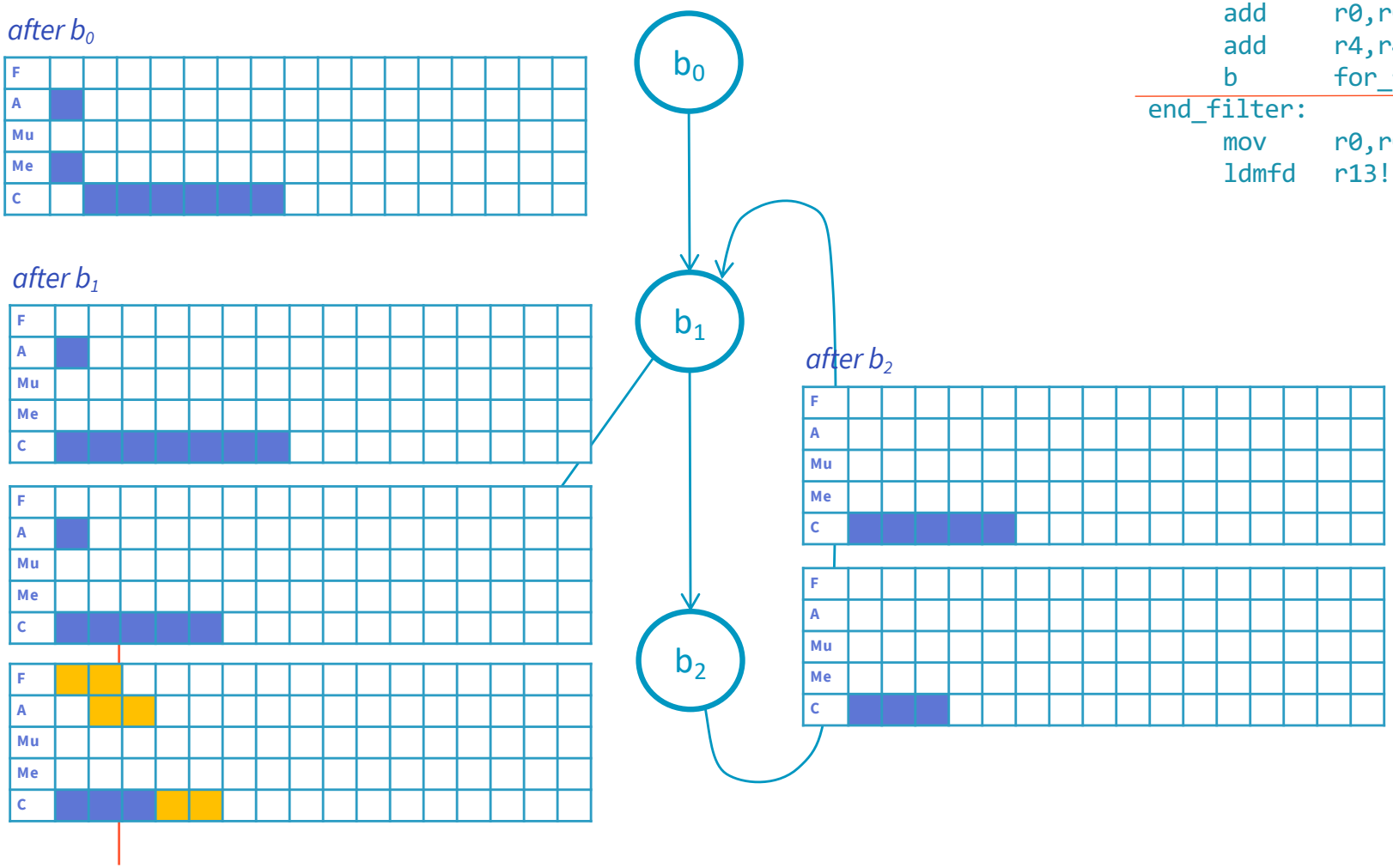
Using abstract interpretation

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```



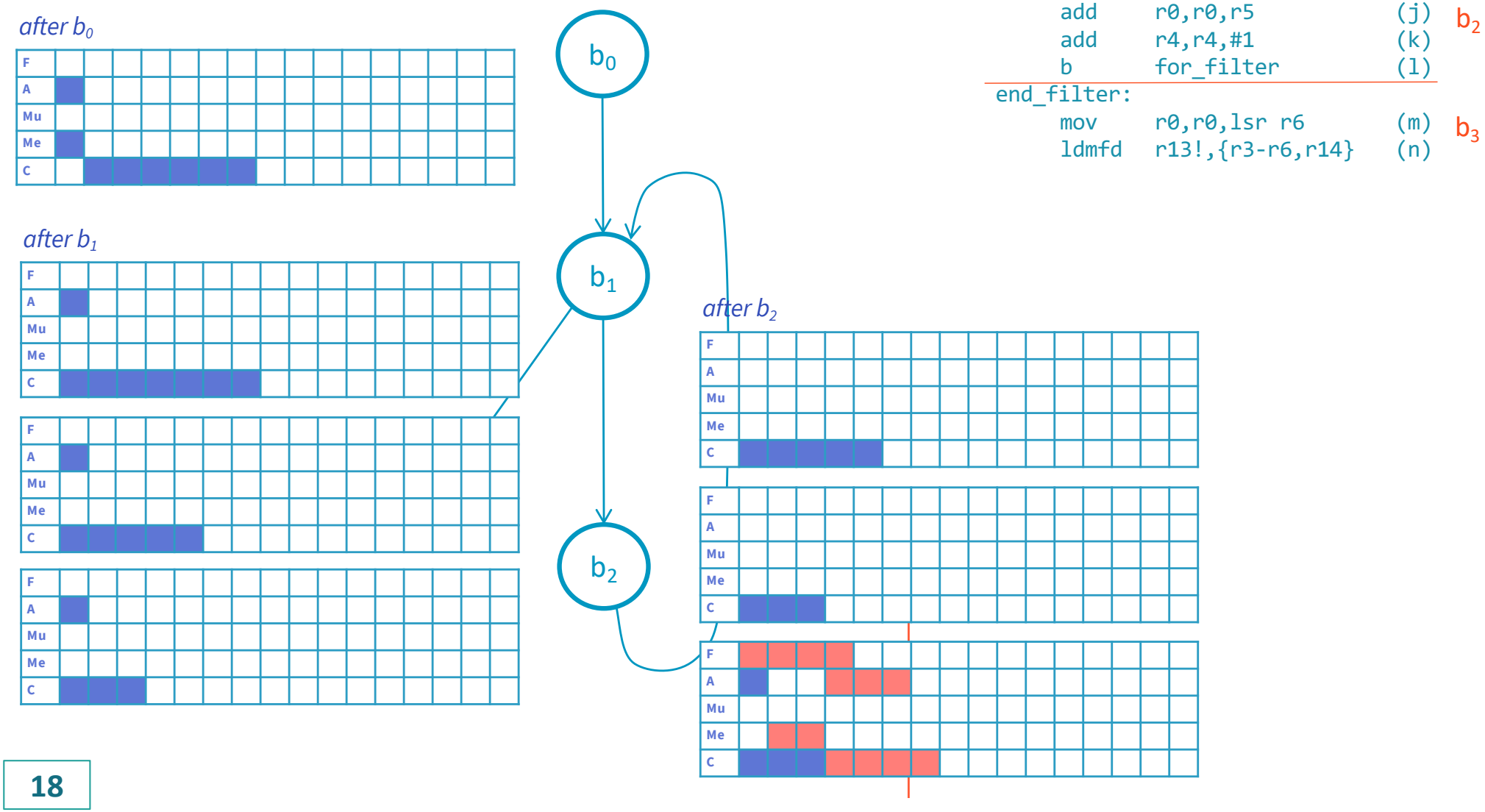
Using abstract interpretation

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```



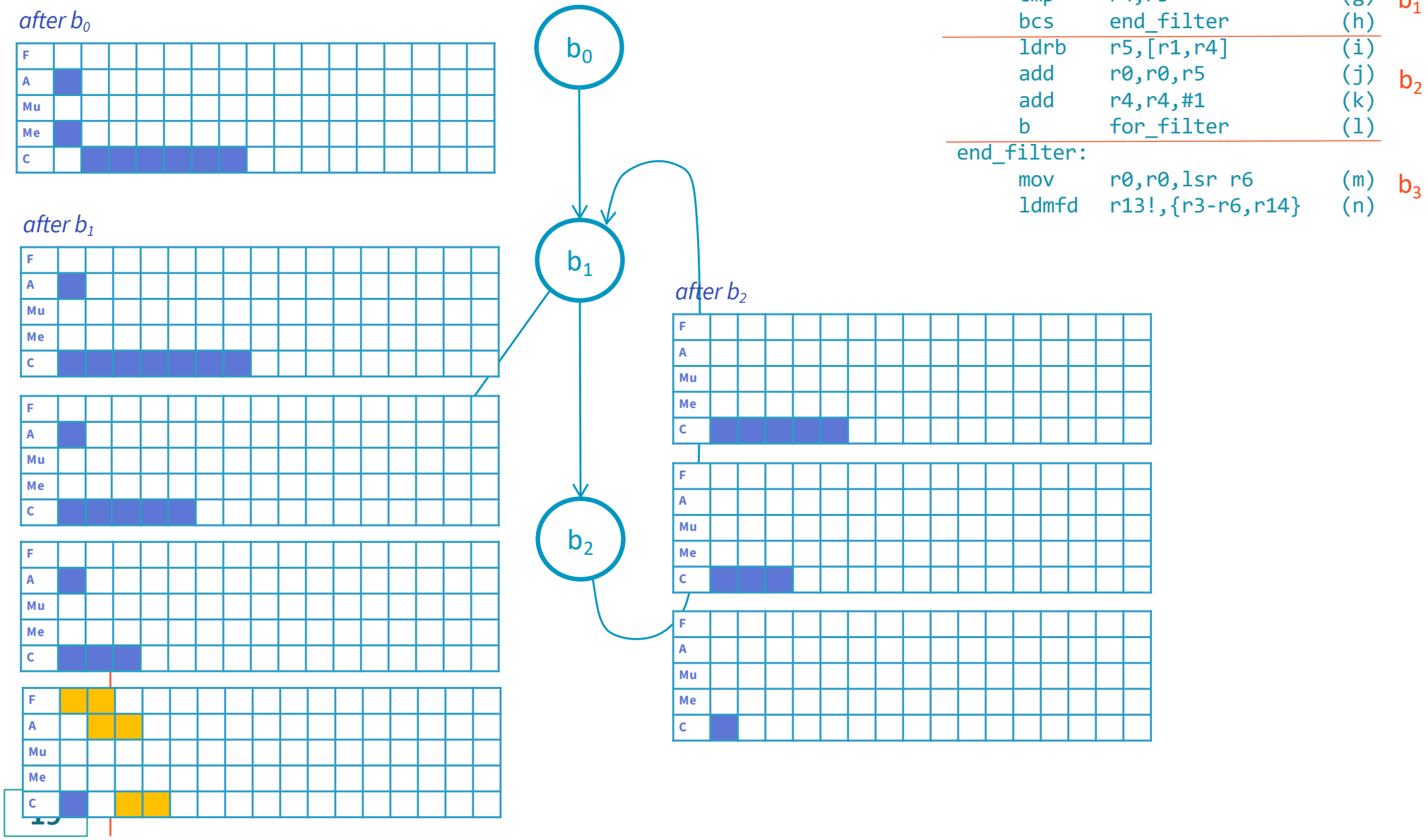
Using abstract interpretation

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```



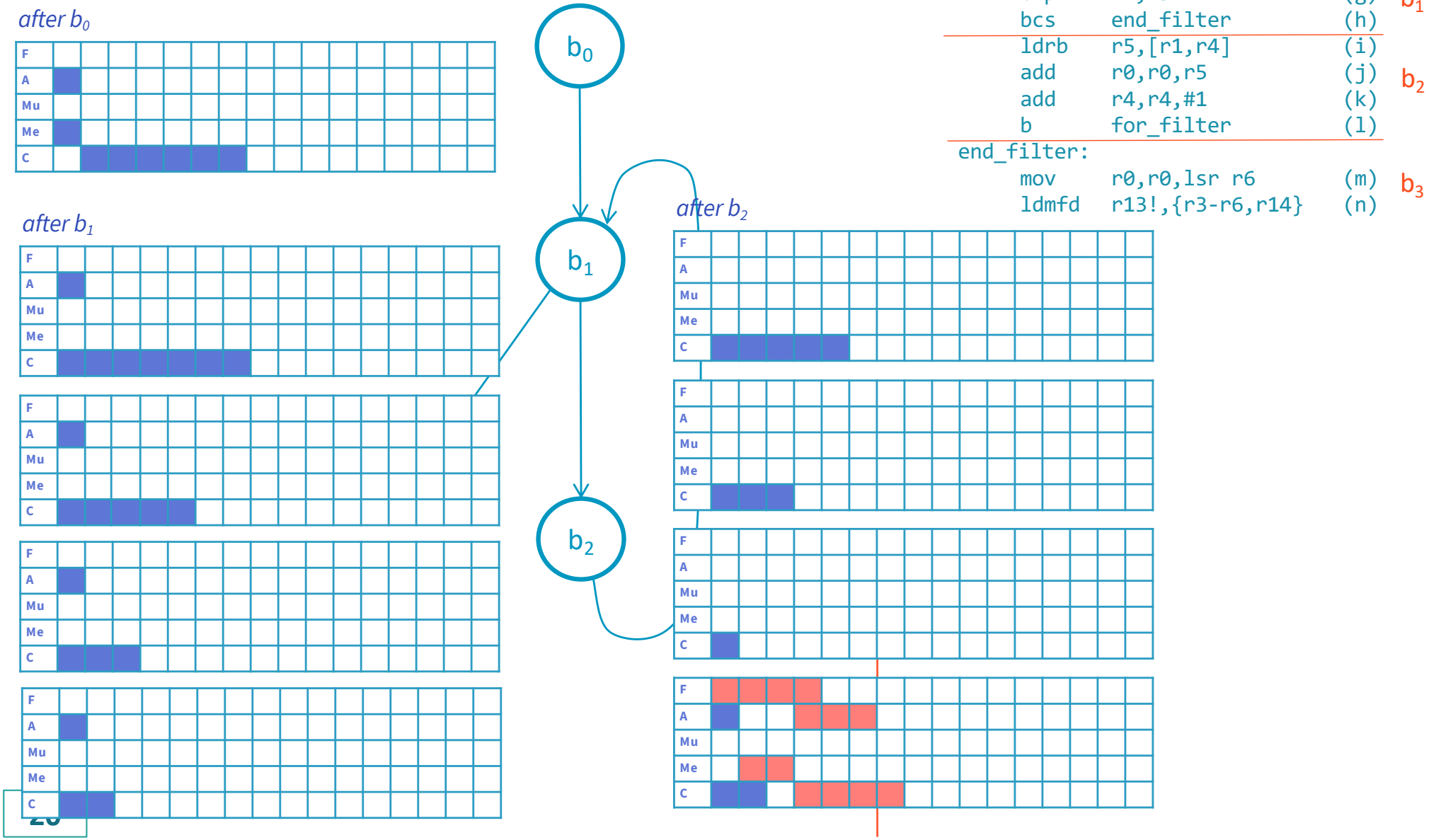
Using abstract interpretation

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}   (a)
      mov     r3,#1              (b)
      mov     r3,r3, lsl r0      (c)   b0
      mov     r6,r0              (d)
      mov     r0,#0              (e)
      mov     r4,#0              (f)
for_filter:
      cmp     r4,r3              (g)   b1
      bcs     end_filter         (h)
      ldrb    r5,[r1,r4]         (i)
      add     r0,r0,r5           (j)   b2
      add     r4,r4,#1           (k)
      b       for_filter         (l)
end_filter:
      mov     r0,r0,lsr r6       (m)   b3
      ldmfd   r13!,{r3-r6,r14}   (n)
```



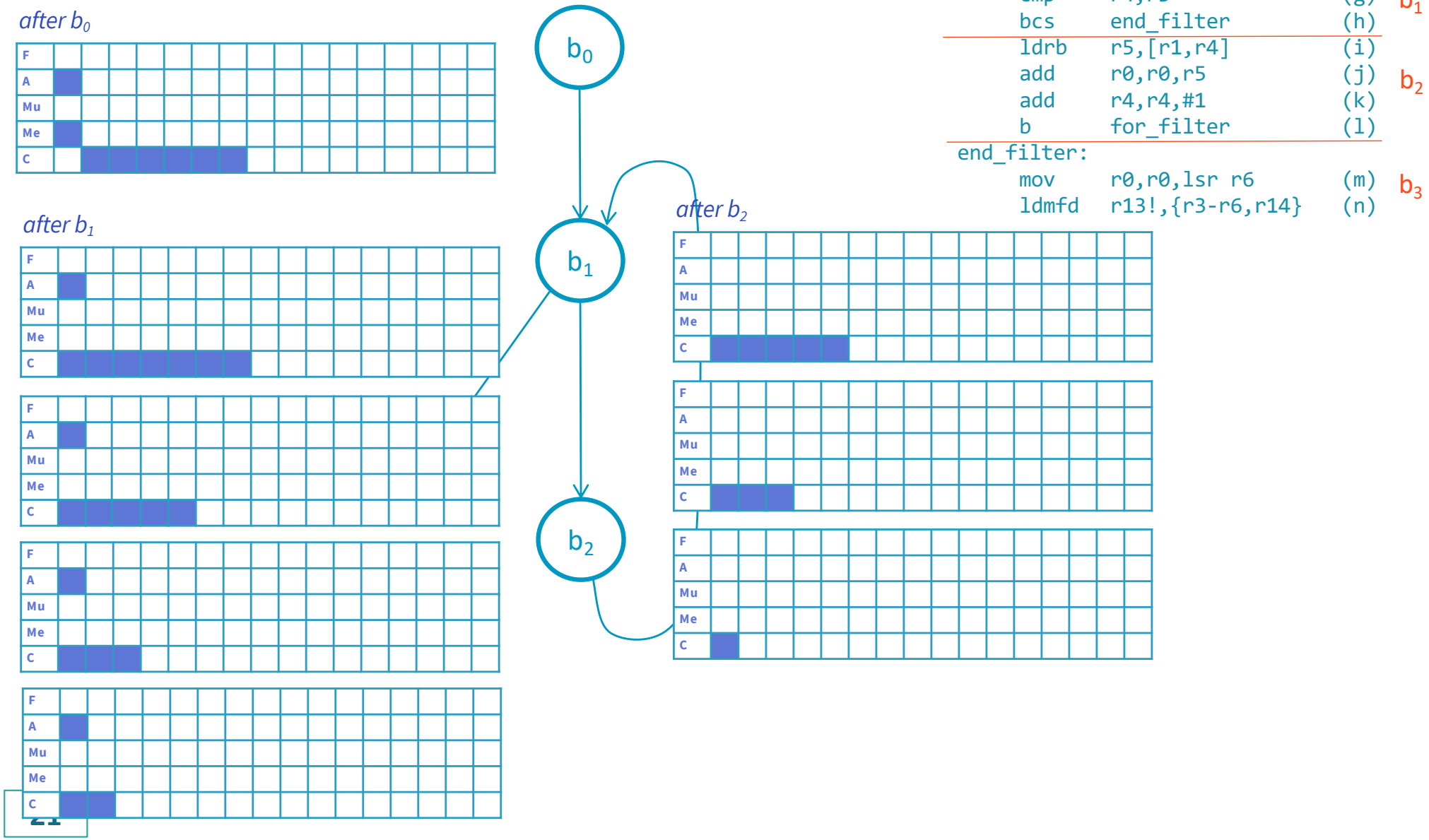
Using abstract interpretation

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```



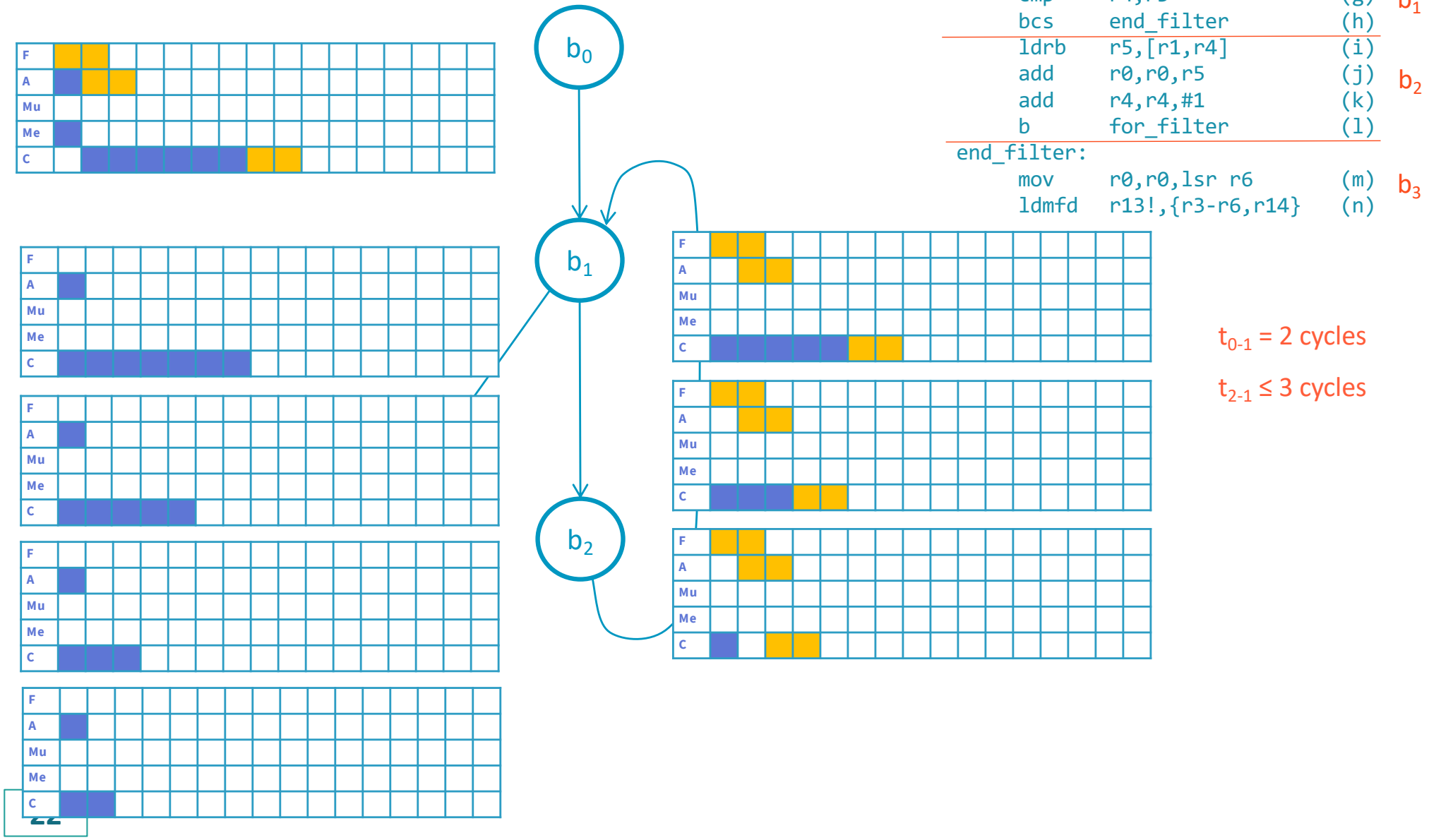
Using abstract interpretation

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```



Using abstract interpretation

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```

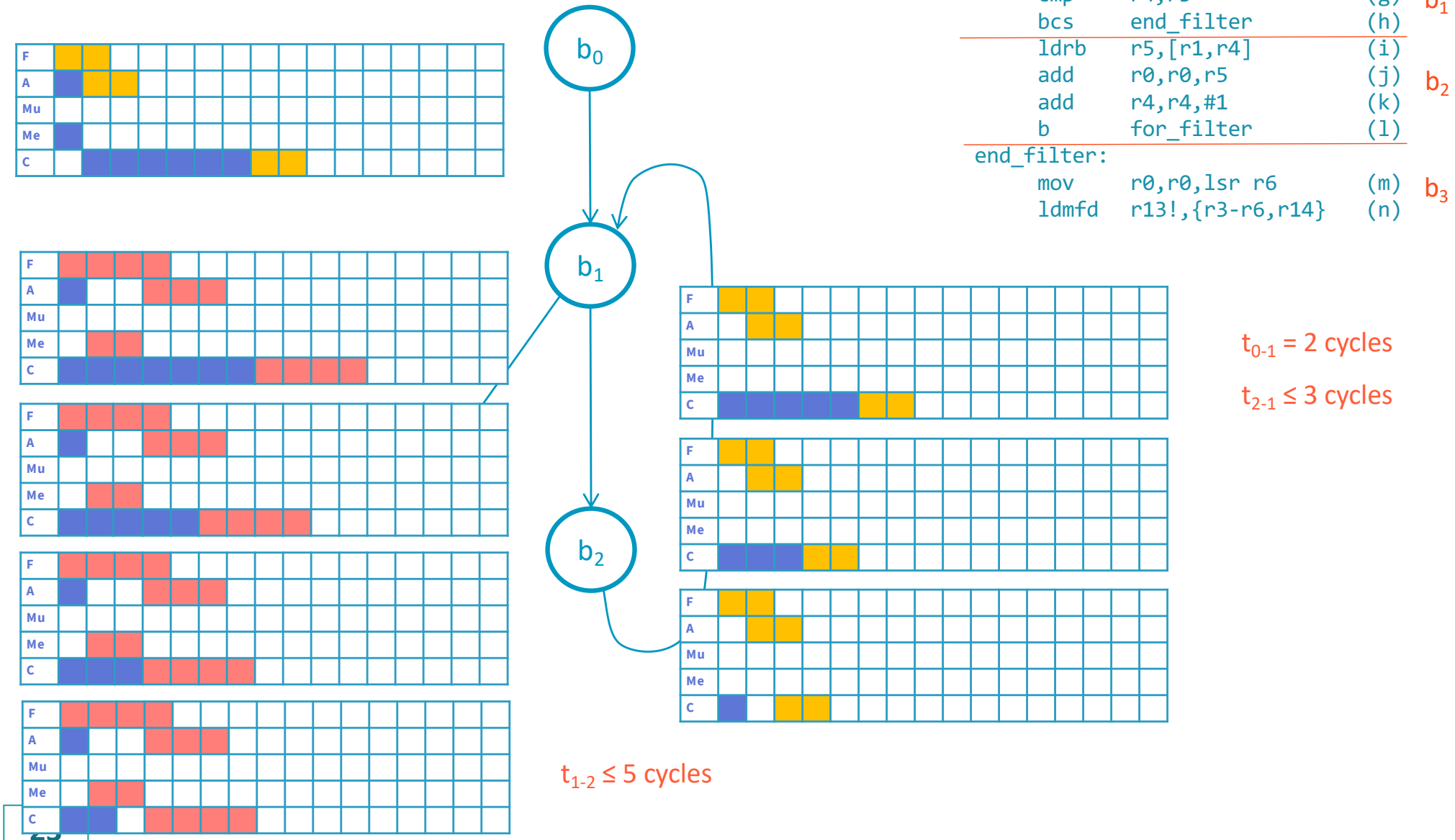


$t_{0\text{-}1}$ = 2 cycles

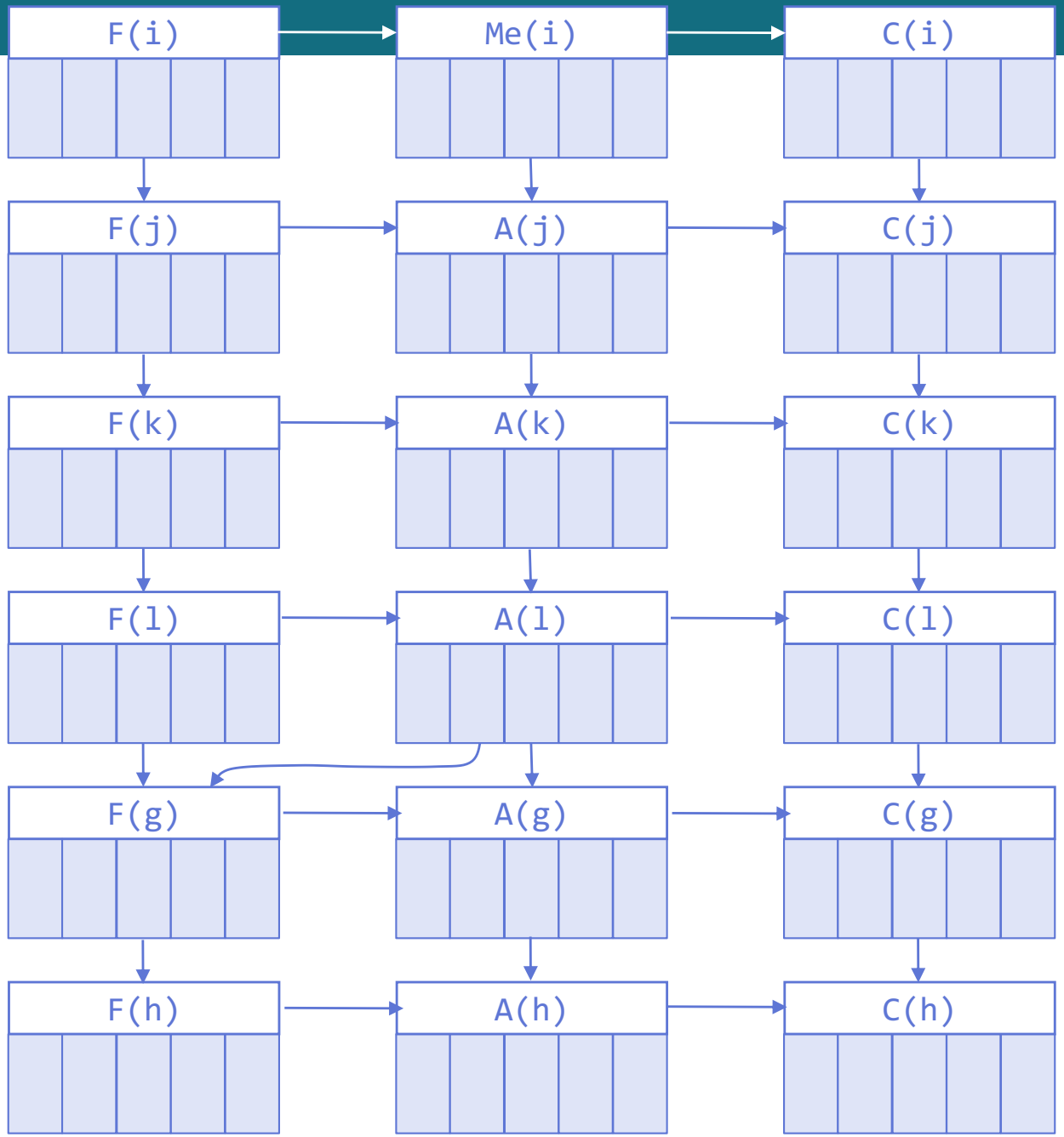
$t_{2\text{-}1} \leq$ 3 cycles

Using abstract interpretation

```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```



Using execution graphs

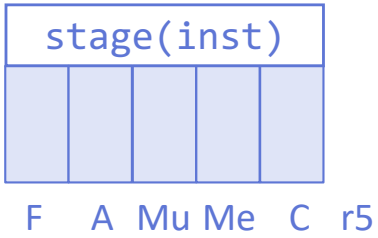



```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```

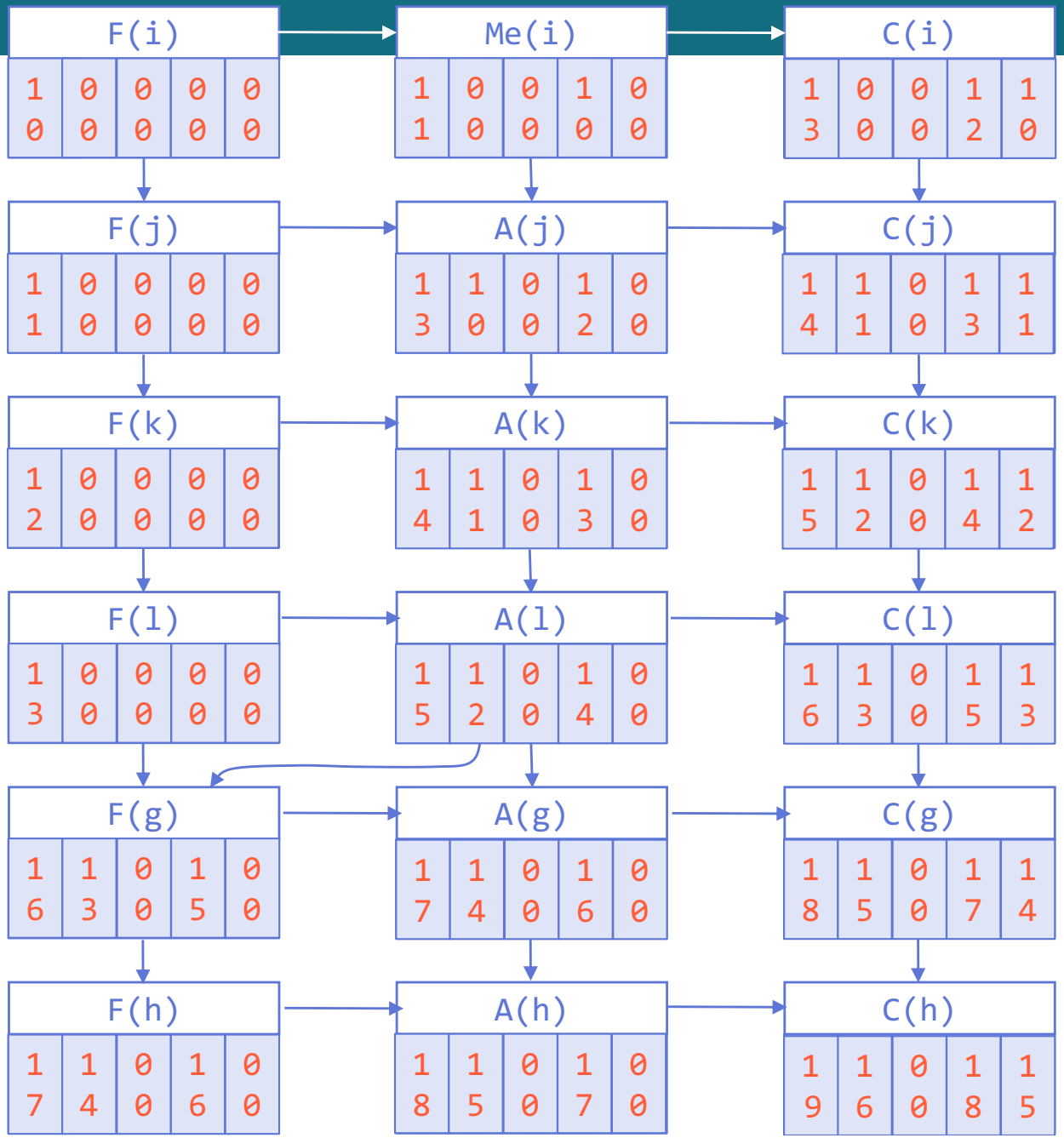
Resources:

pipeline stages: F, A, Mu, Me, C

registers: r5



Using execution graphs

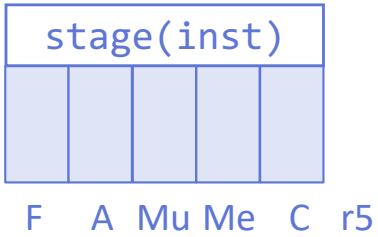



```
fixFilter:
      stmfd   r13!,{r3-r6,r14}    (a)
      mov     r3,#1               (b)
      mov     r3,r3, lsl r0       (c)   b0
      mov     r6,r0               (d)
      mov     r0,#0               (e)
      mov     r4,#0               (f)
for_filter:
      cmp     r4,r3               (g)   b1
      bcs     end_filter          (h)
      ldrb    r5,[r1,r4]          (i)
      add     r0,r0,r5            (j)   b2
      add     r4,r4,#1            (k)
      b       for_filter          (l)
end_filter:
      mov     r0,r0,lsr r6        (m)   b3
      ldmfd   r13!,{r3-r6,r14}    (n)
```

Resources:

pipeline stages: F, A, Mu, Me, C

registers: r5



Using execution graphs

C(l)				
16	13	00	15	13

$t_{C(l)} = \max(t_F+6 , t_A+3 , t_{Me}+5 , t_C+3)$

C(h)				
19	16	00	18	15

$t_{C(h)} = \max(t_F+9 , t_A+6 , t_{Me}+8 , t_C+5)$

F	A	Mu	Me	C

$t_{C(h)} - t_{C(l)} = \max(9-6 , 6-3 , 8-5 , 5-3)$

$t_{2\text{-}1} \leq 3$ cycles

ILP formulation

$$\max T = 13\,x_0 + 2\,x_{0\text{-}1} + 5\,x_{1\text{-}2} + 3\,x_{2\text{-}1} + 8\,x_{1\text{-}3}$$

$x_0 = 1$

$x_0 = x_{0\text{-}1}$

$x_1 = x_{0\text{-}1} + x_{2\text{-}1}$

$x_1 = x_{1\text{-}2} + x_{1\text{-}3}$

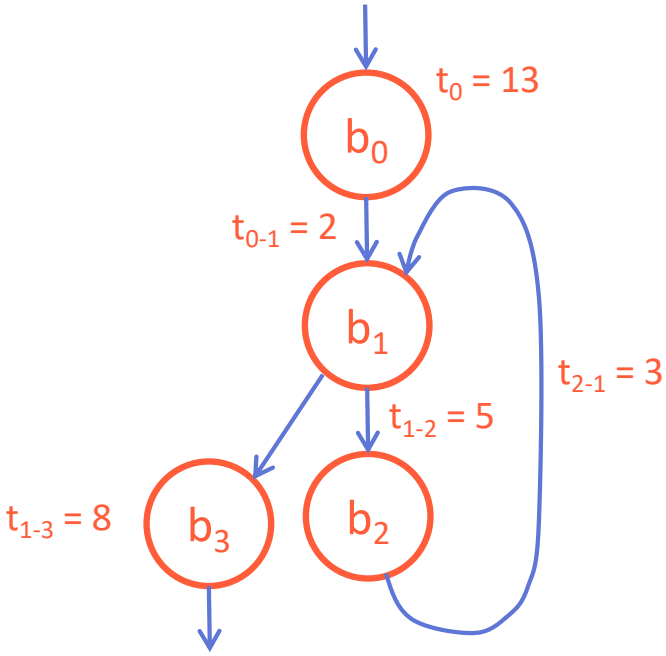
$x_2 = x_{1\text{-}2}$

$x_2 = x_{2\text{-}1}$

$x_3 = x_{1\text{-}3}$

$x_3 = 1$

$x_{2\text{-}1} \leq 256$



Instruction cache analysis

```
fixFilter:
L0➔S0   stmfd   r13!,{r3-r6,r14}   (a)
        mov     r3,#1               (b)
L1➔S1   mov     r3,r3, lsl r0       (c)
        mov     r6,r0               (d)
L2➔S0   mov     r0,#0               (e)
        mov     r4,#0               (f)
for_filter:
L3➔S1   cmp     r4,r3               (g)
        bcs     end_filter          (h)
L4➔S0   ldrb    r5,[r1,r4]          (i)
        add     r0,r0,r5            (j)
L5➔S1   add     r4,r4,#1            (k)
        b       for_filter          (l)
end_filter:
L6➔S0   mov     r0,r0,lsr r6        (m)
        ldmfd   r13!,{r3-r6,r14}    (n)
```

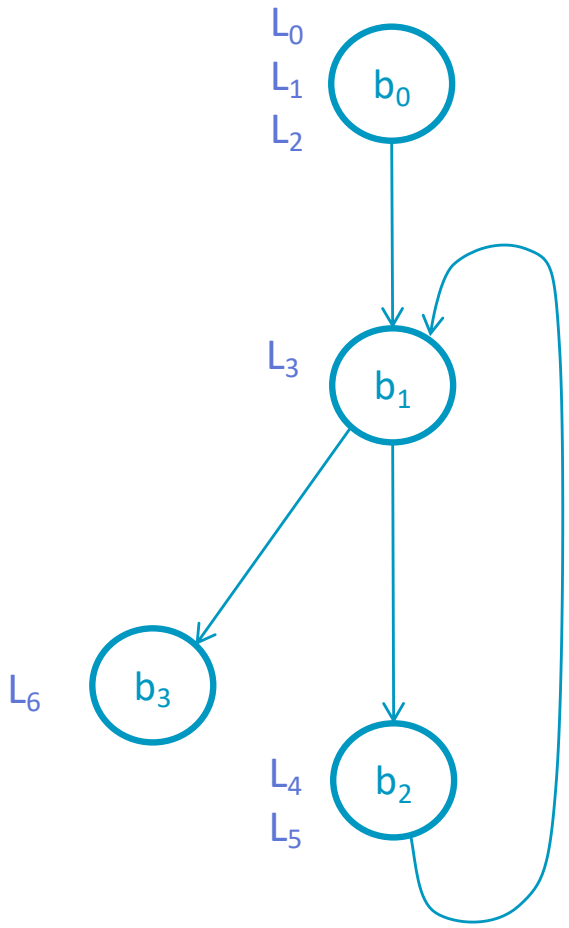
b_0 b_1 b_2 b_3

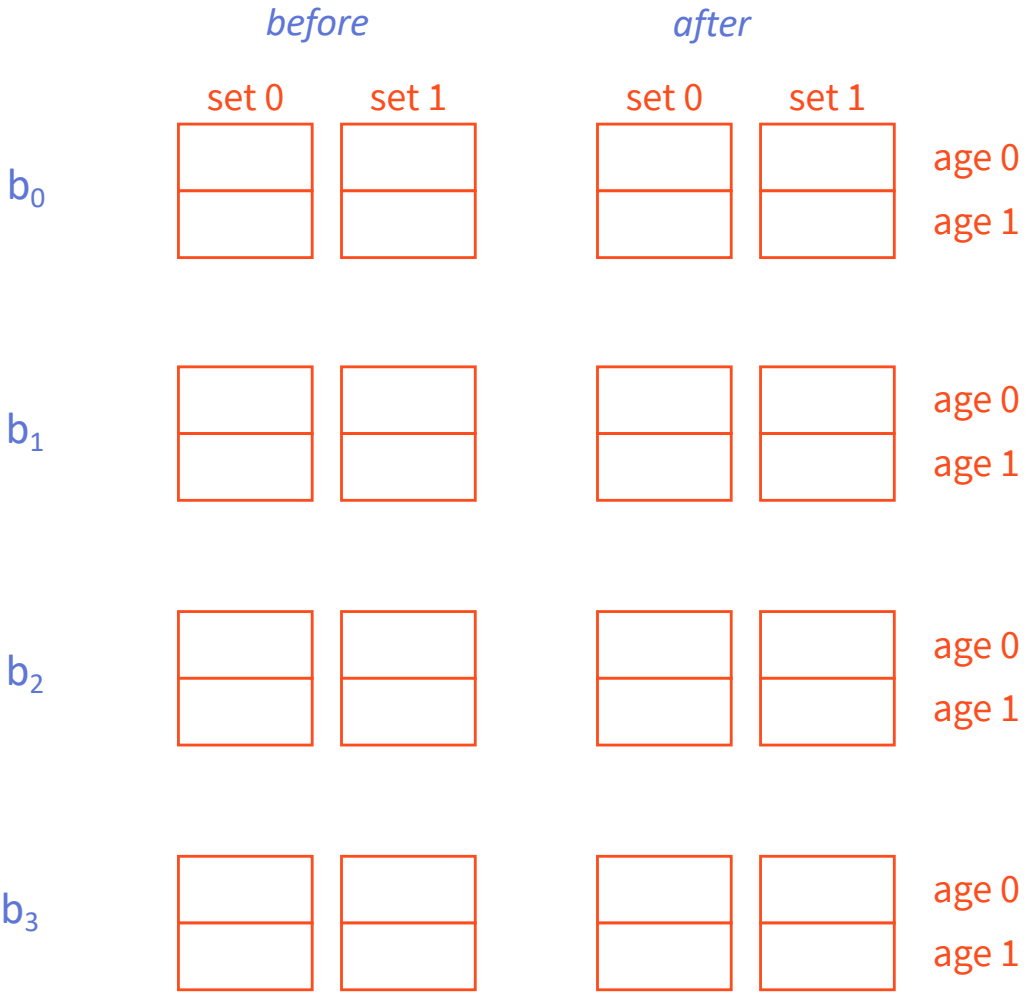
Assumptions:

- 32-byte 2-way set associative cache
- 8-byte cache lines (memory blocks)
- 4-byte instruction codes
- function stored at address 0

Cache analysis by Abstract Interpretation

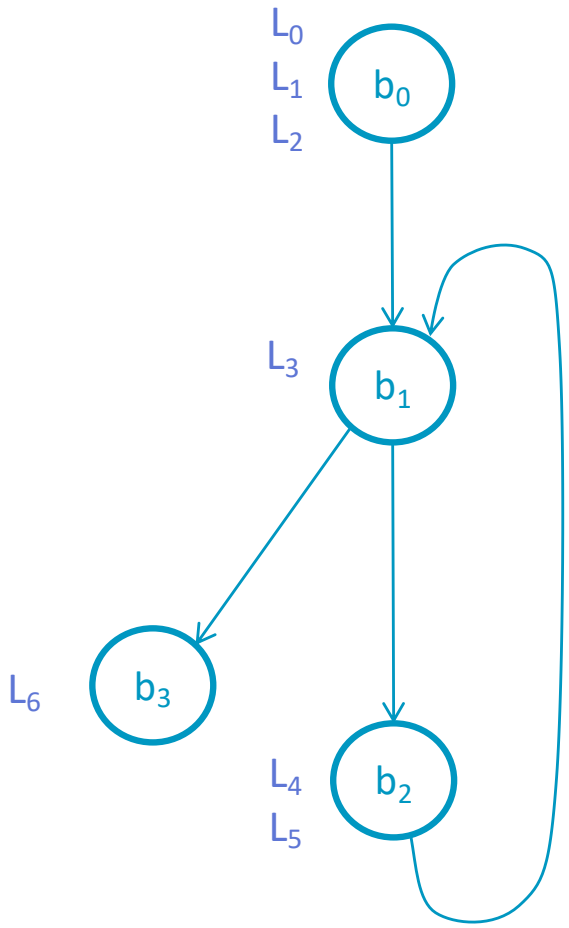
MUST analysis

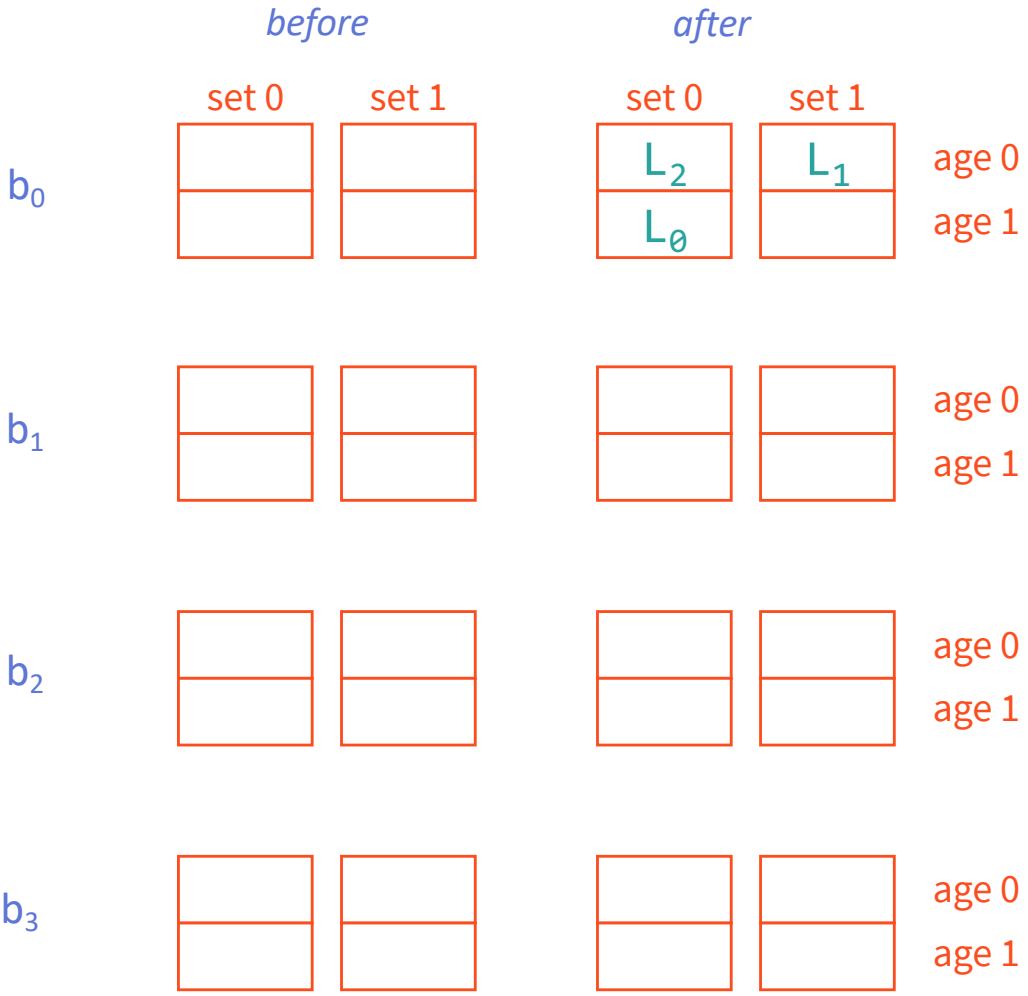




Cache analysis by Abstract Interpretation

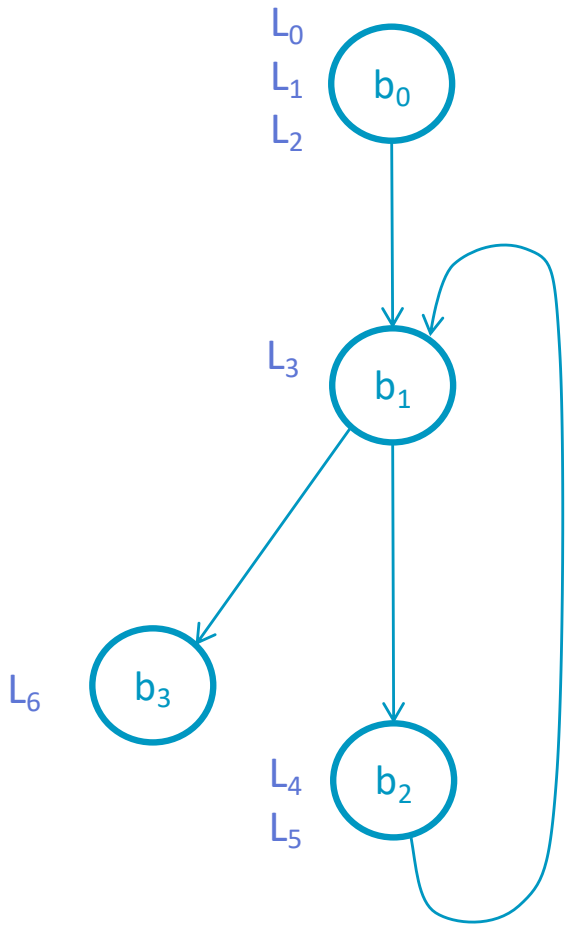
MUST analysis

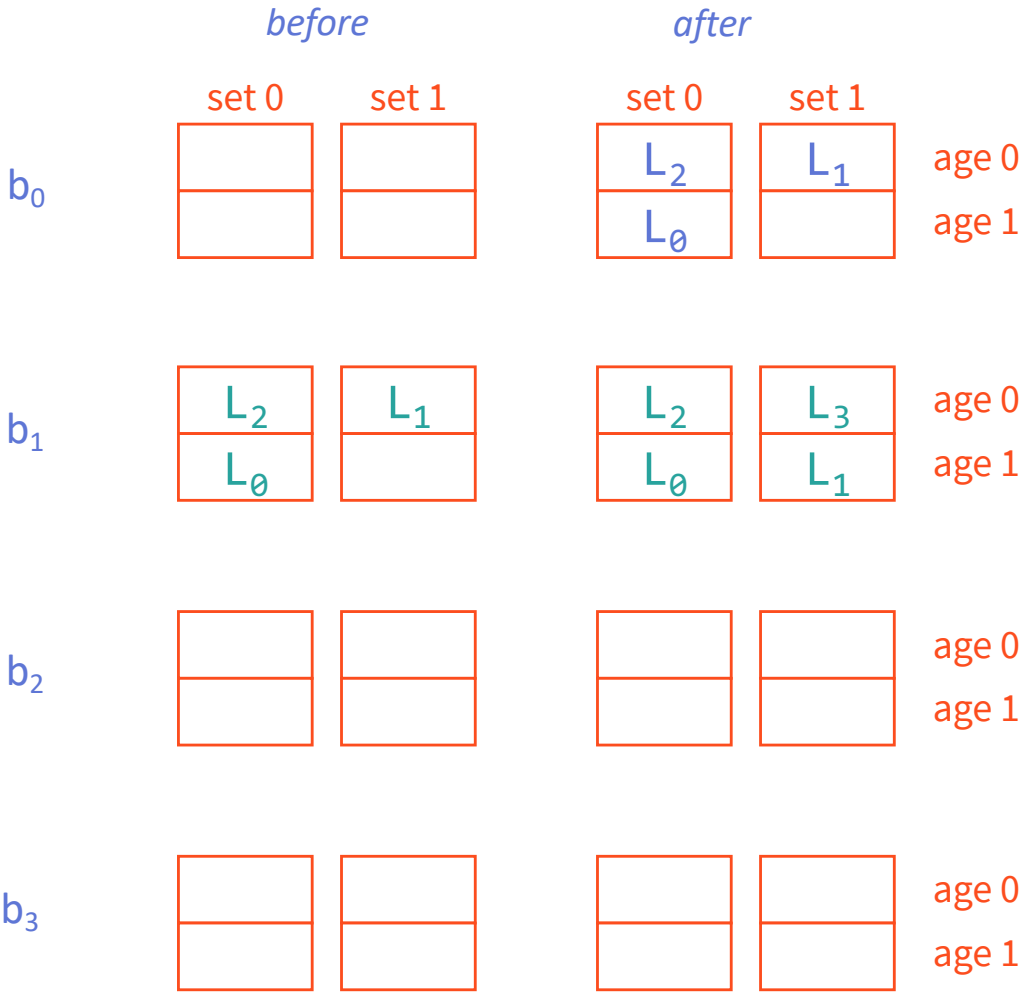




Cache analysis by Abstract Interpretation

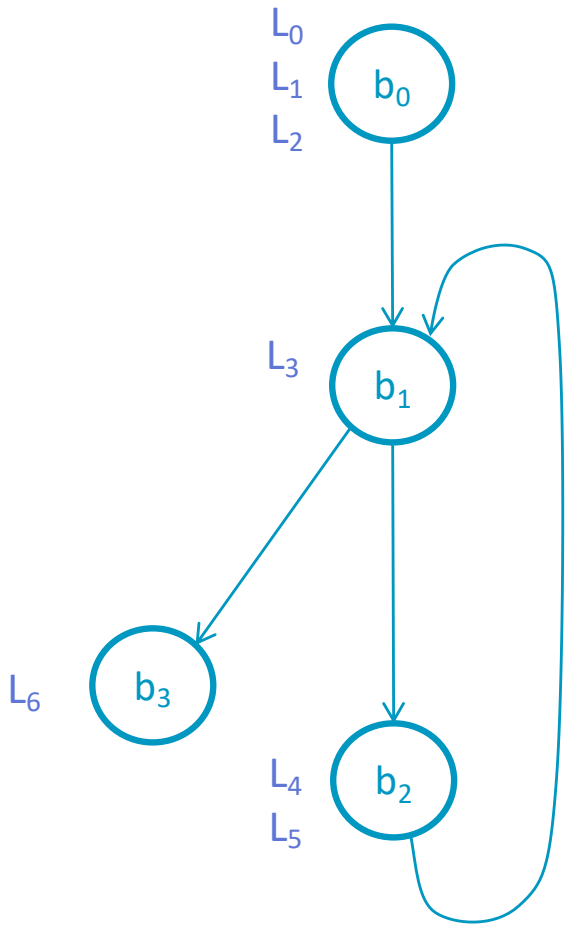
MUST analysis

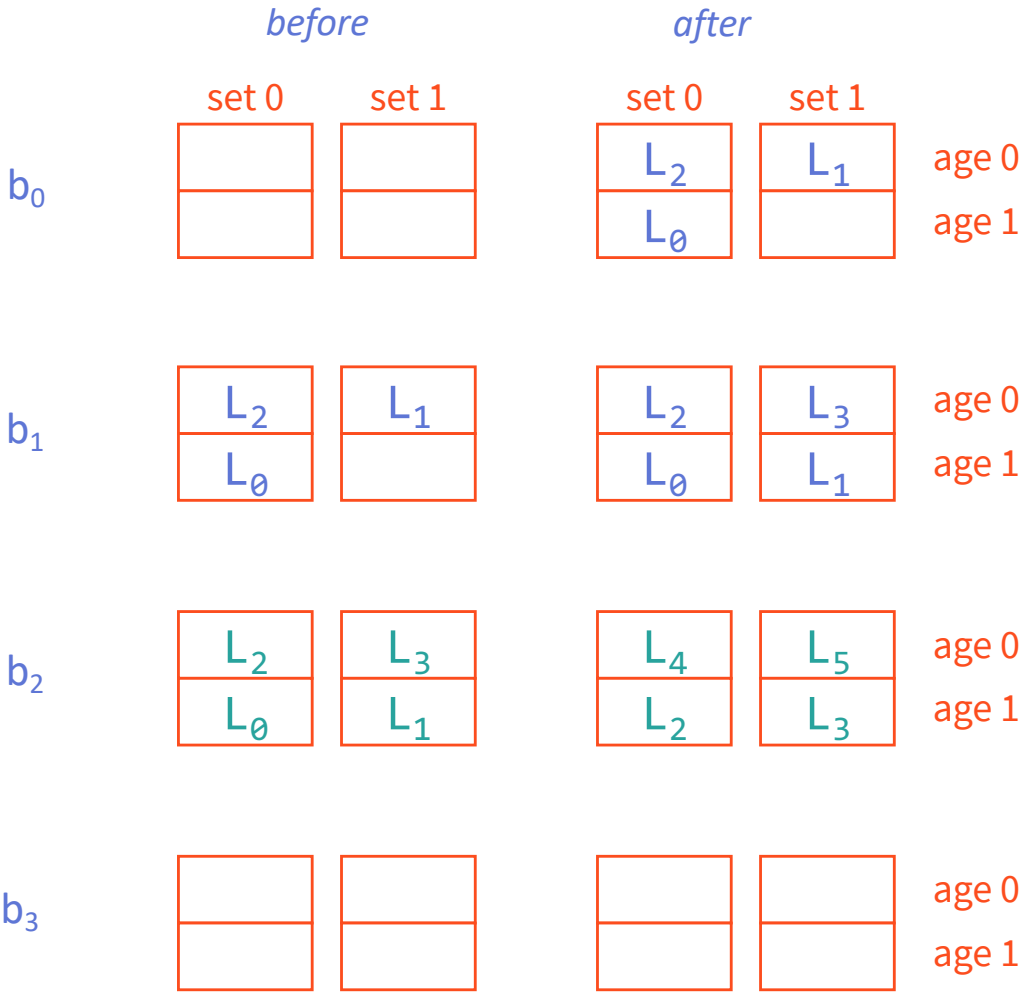




Cache analysis by Abstract Interpretation

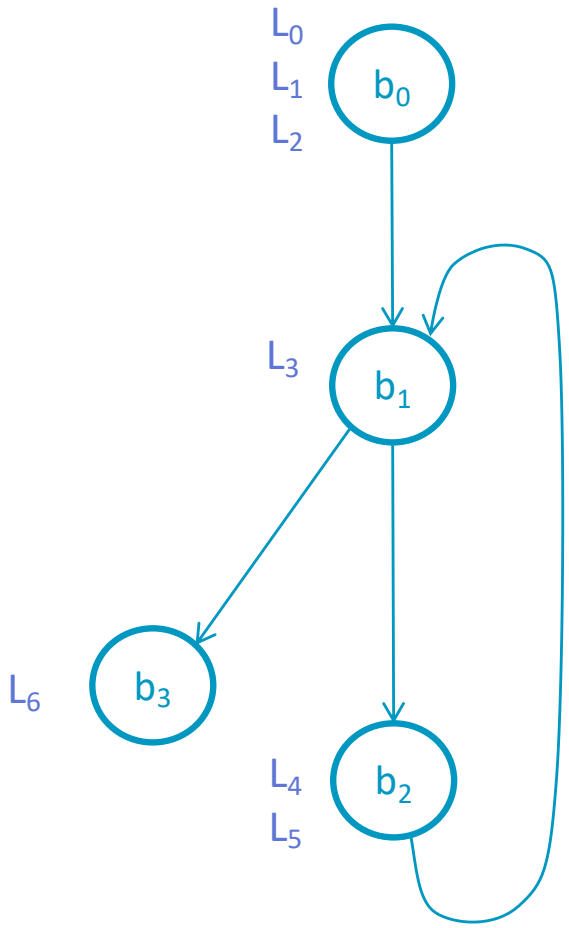
MUST analysis

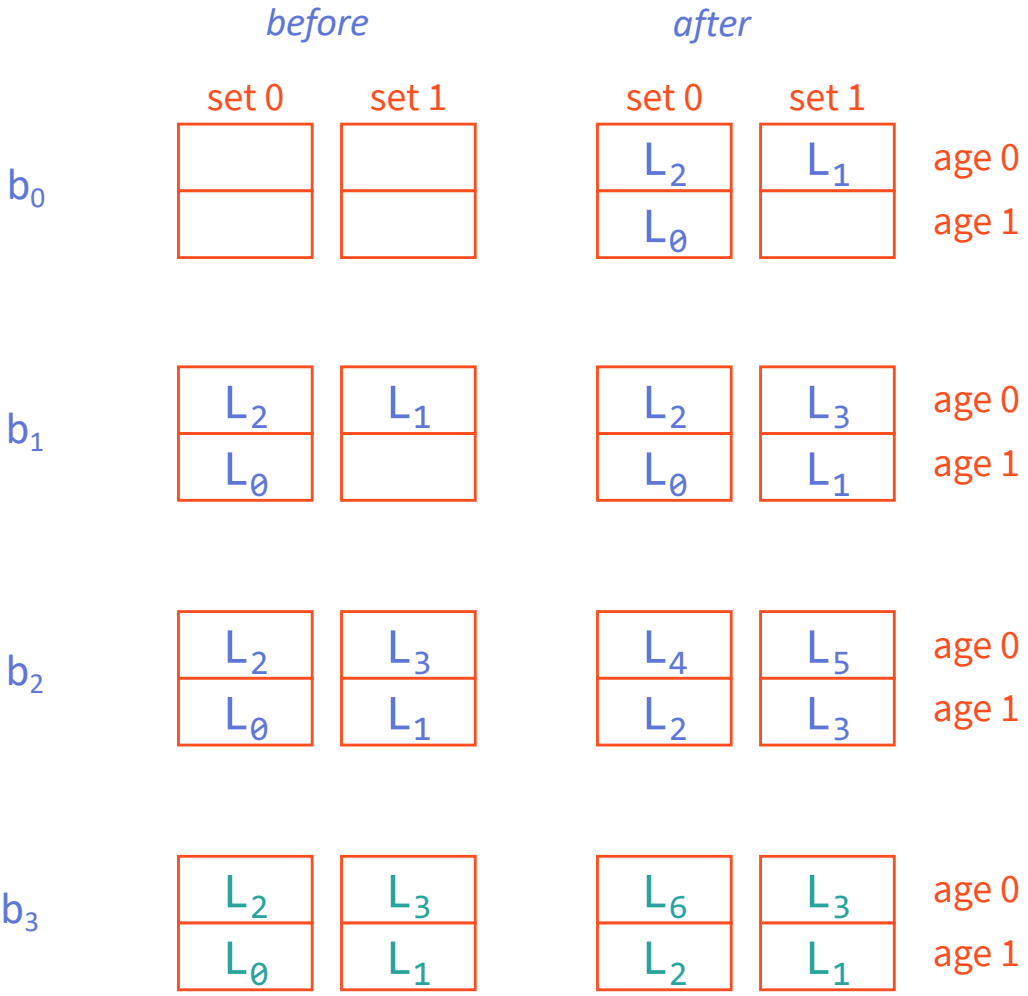




Cache analysis by Abstract Interpretation

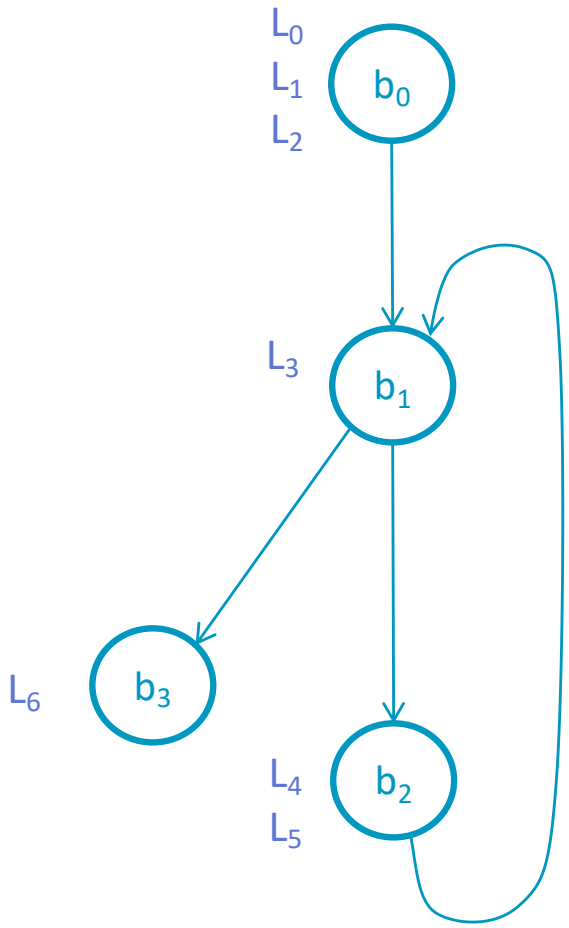
MUST analysis

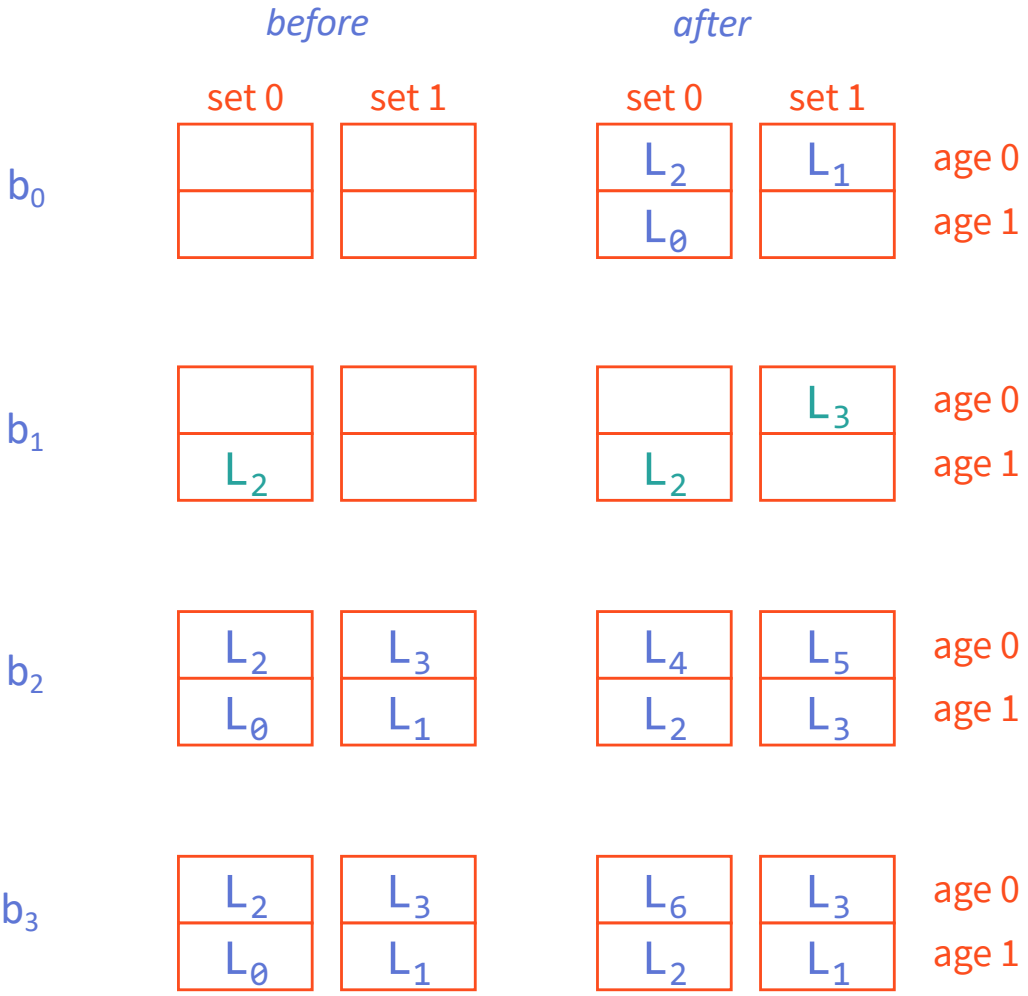




Cache analysis by Abstract Interpretation

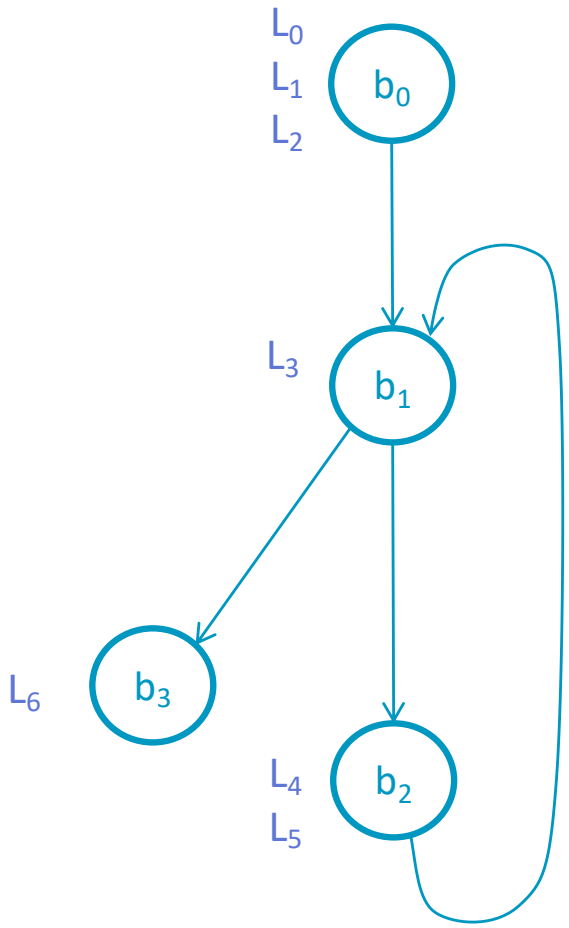
MUST analysis

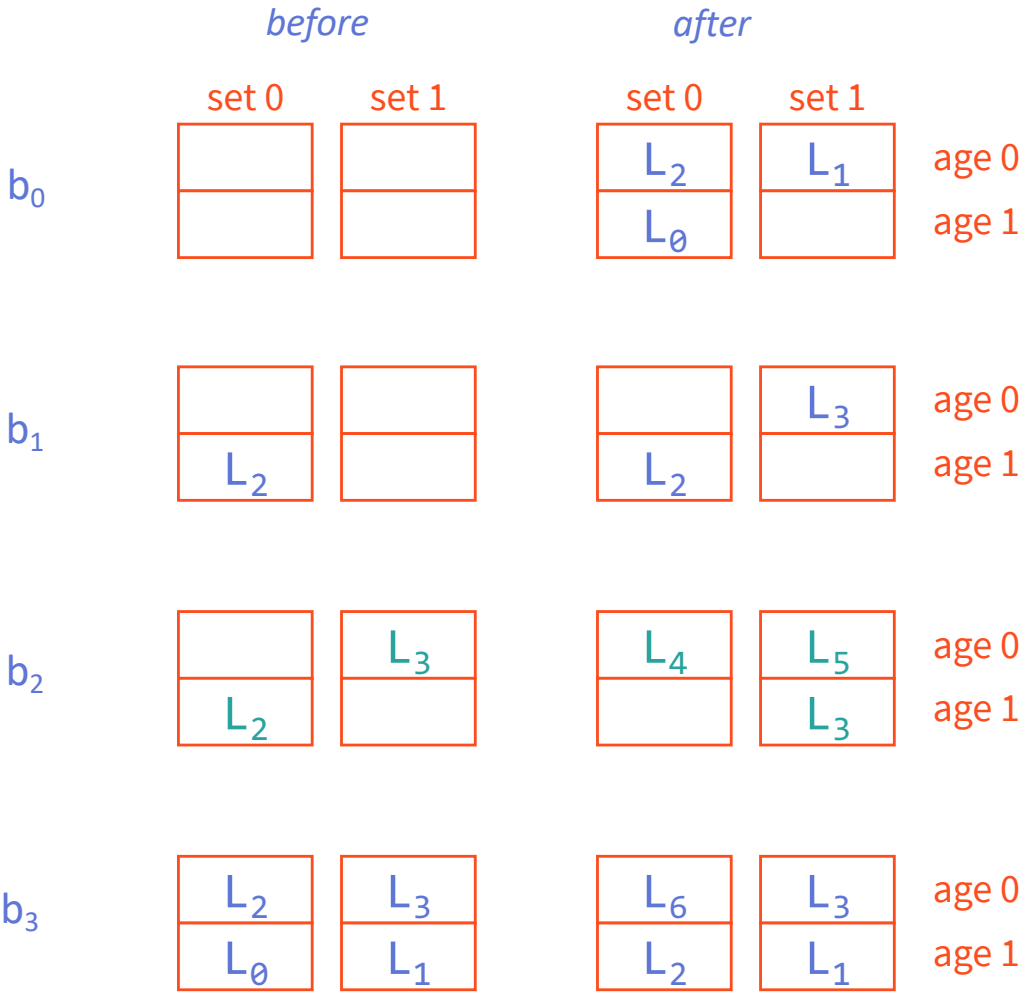




Cache analysis by Abstract Interpretation

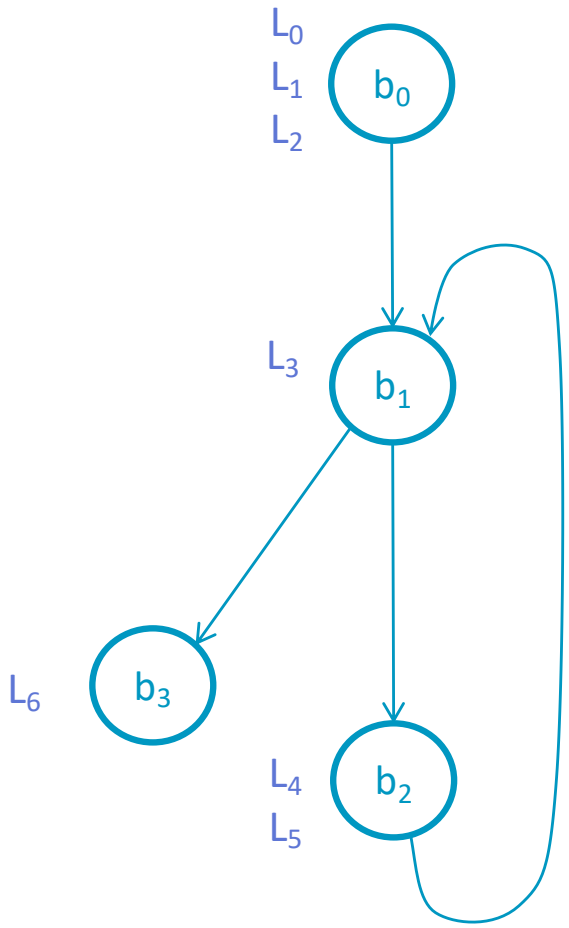
MUST analysis

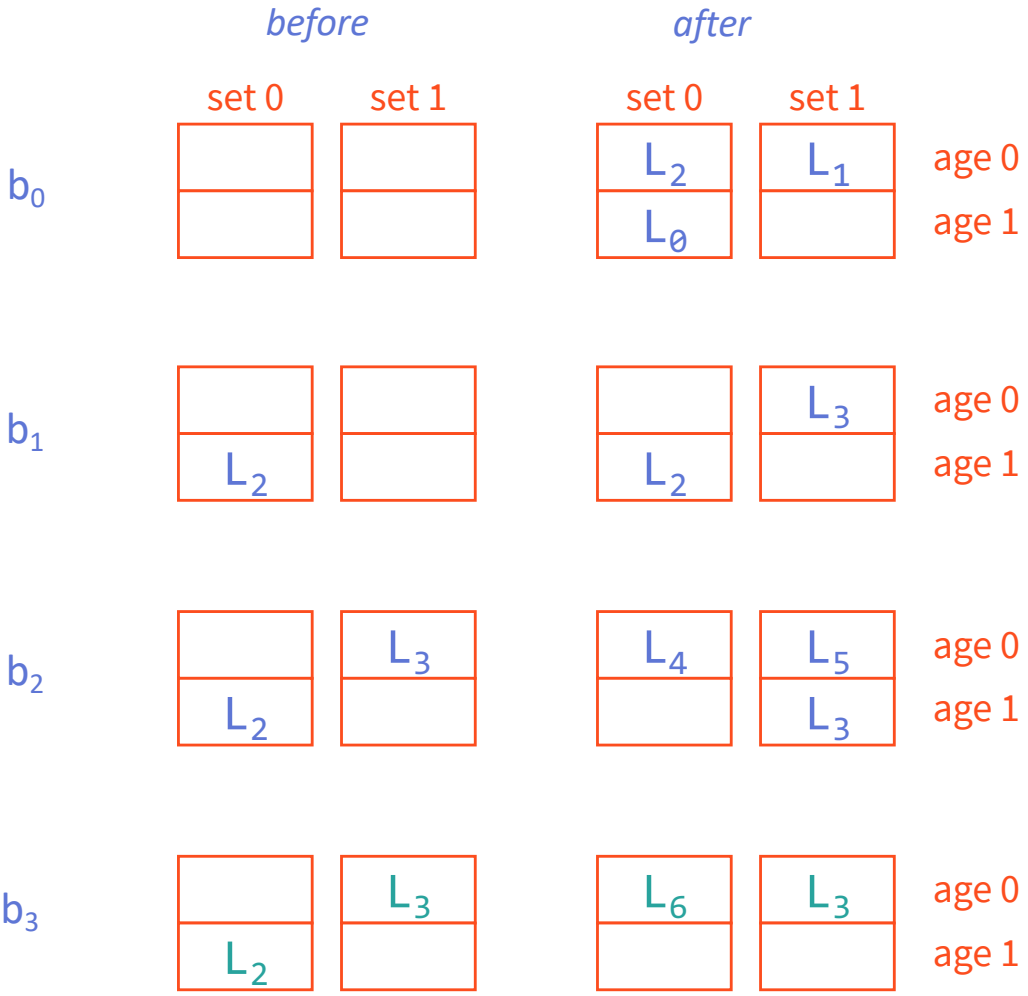




Cache analysis by Abstract Interpretation

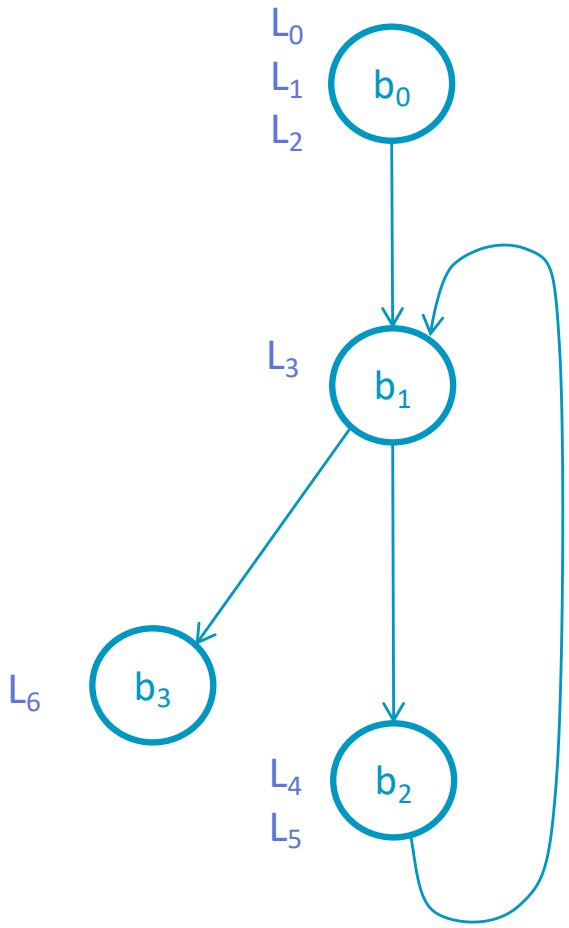
MUST analysis

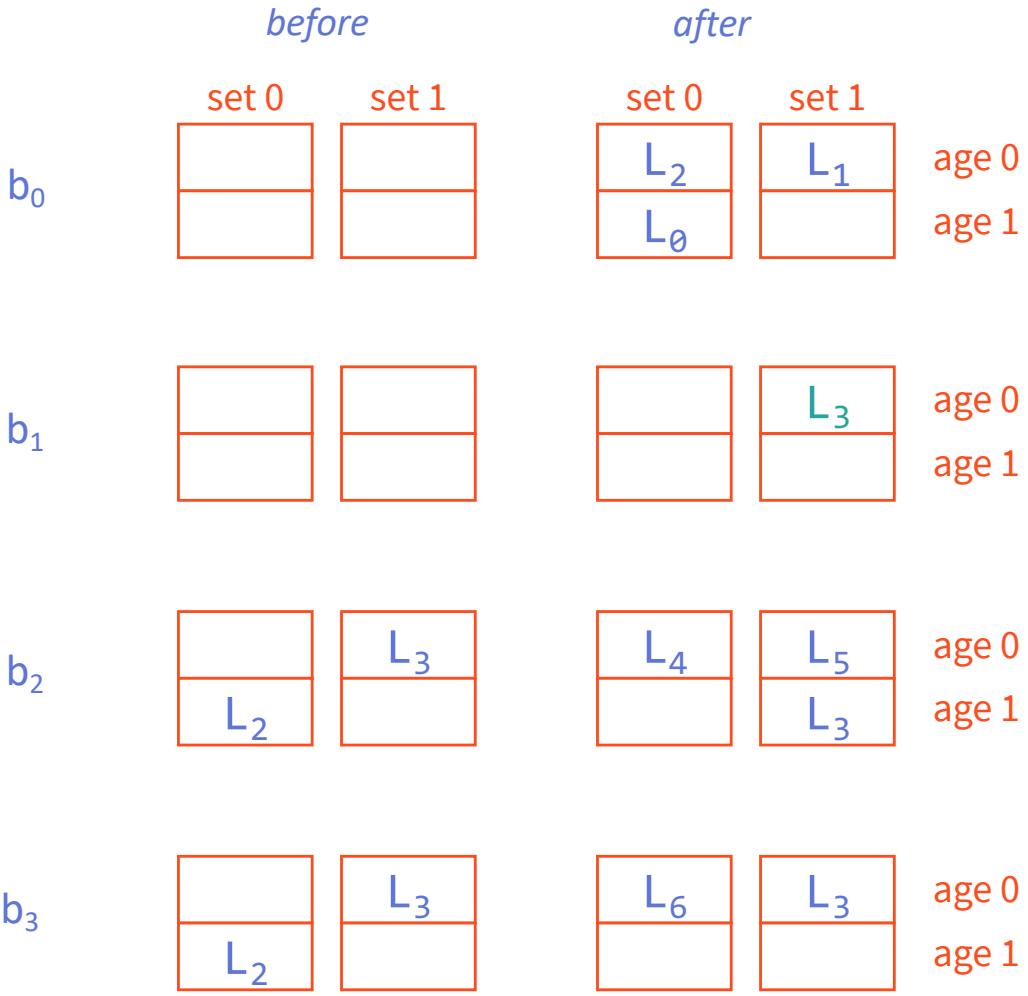




Cache analysis by Abstract Interpretation

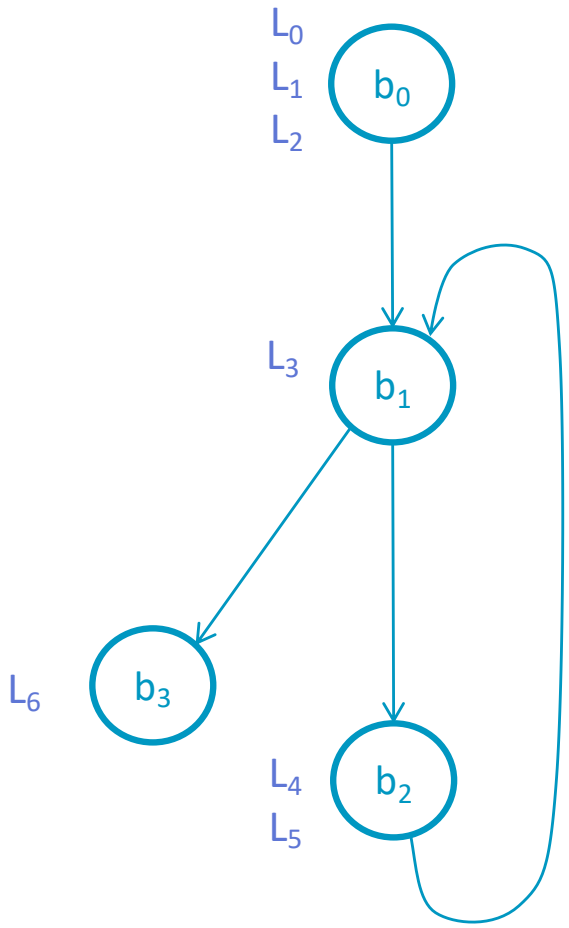
MUST analysis

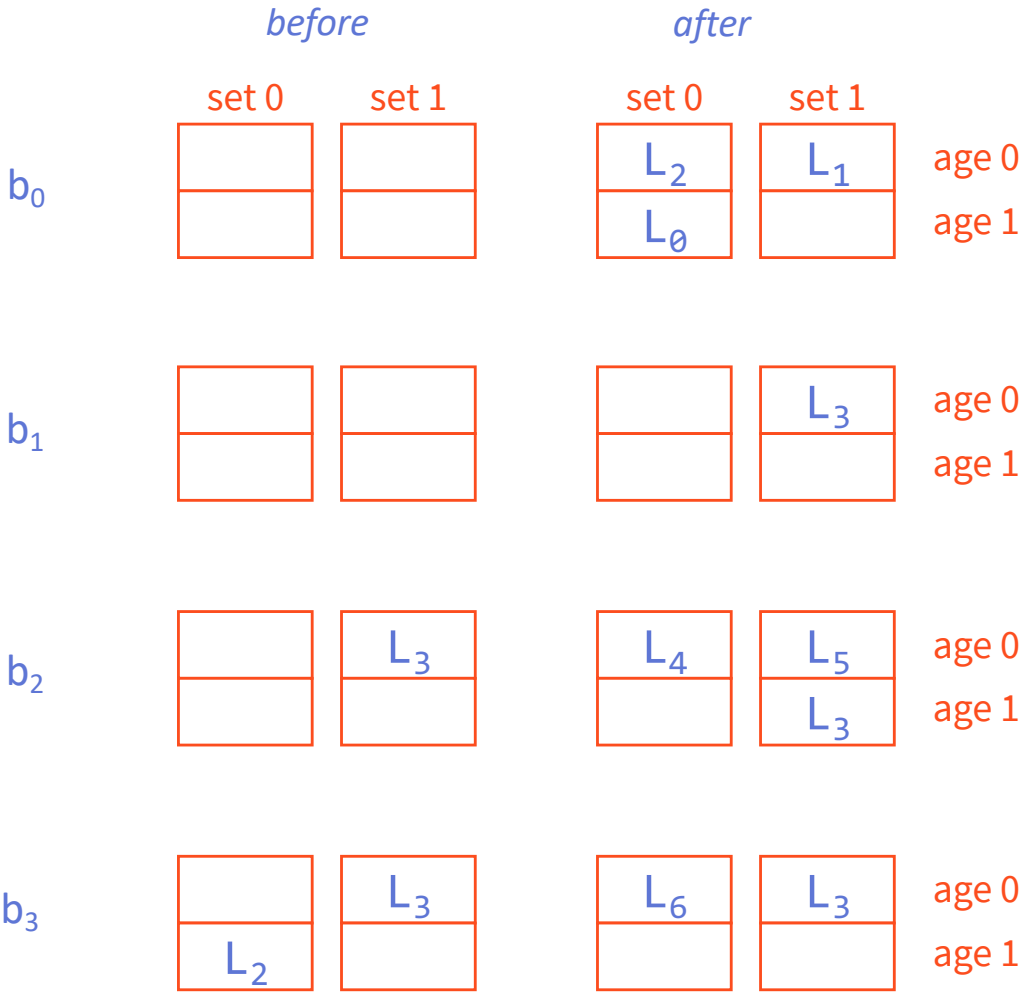




Cache analysis by Abstract Interpretation

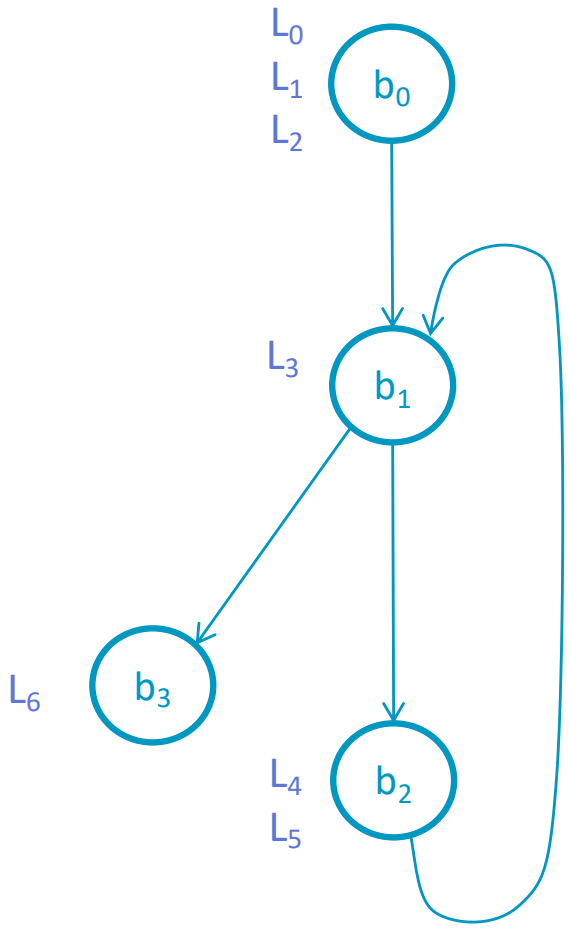
MUST analysis



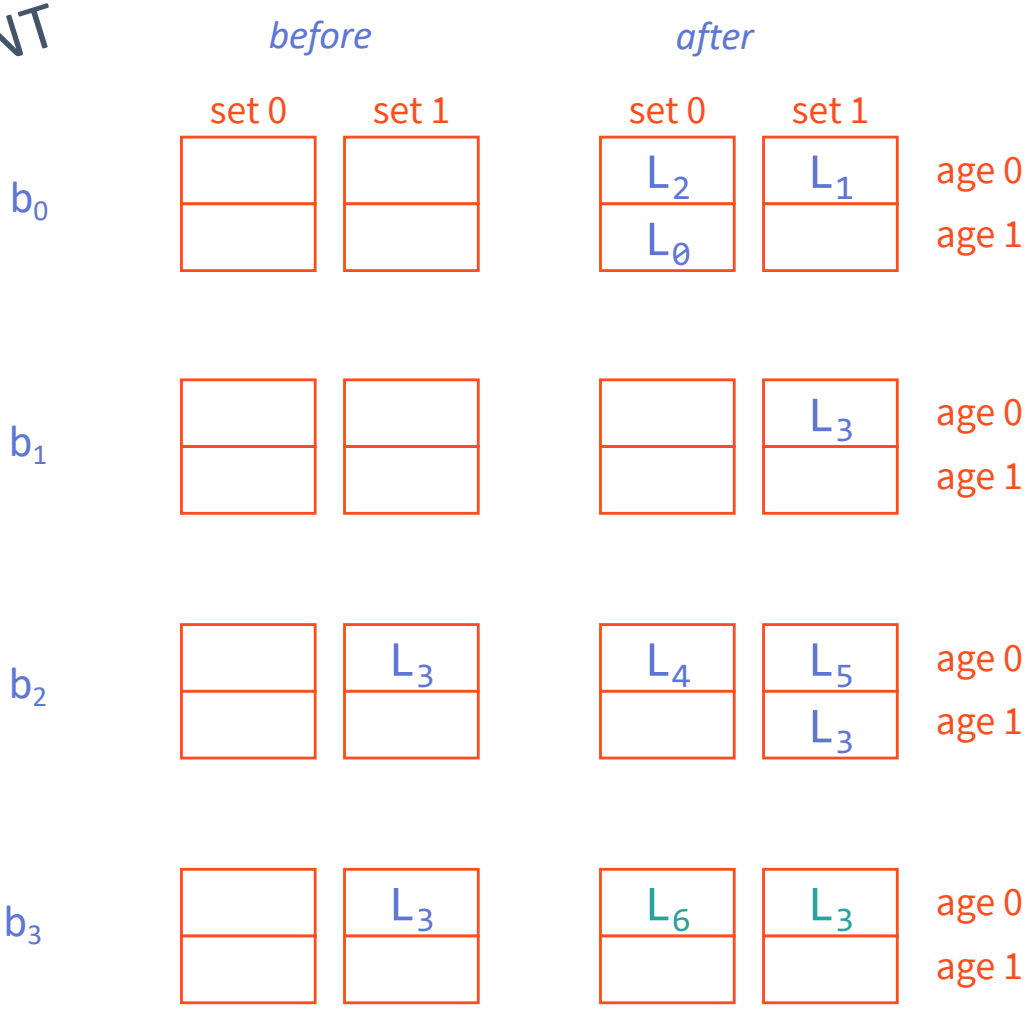


Cache analysis by Abstract Interpretation

MUST analysis

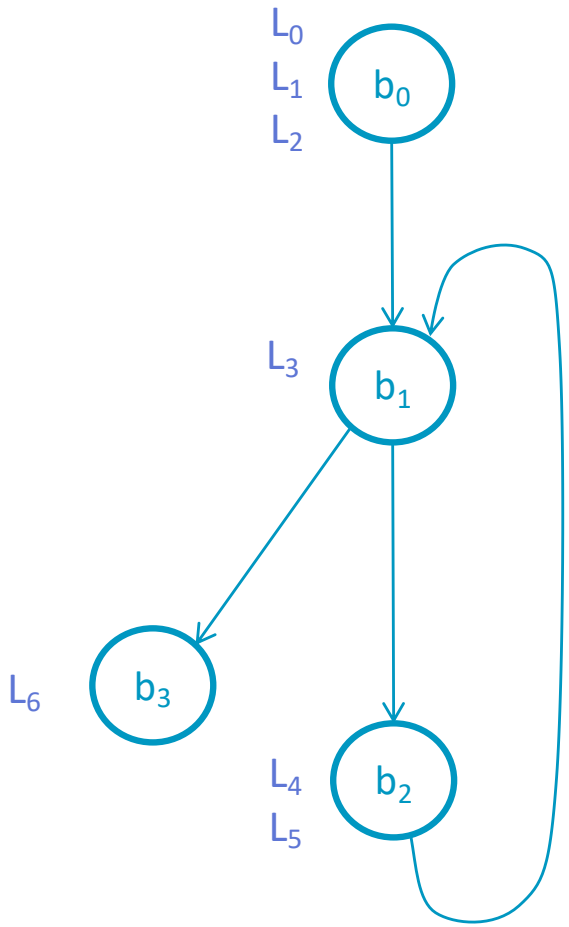


FIX POINT



Cache analysis by Abstract Interpretation

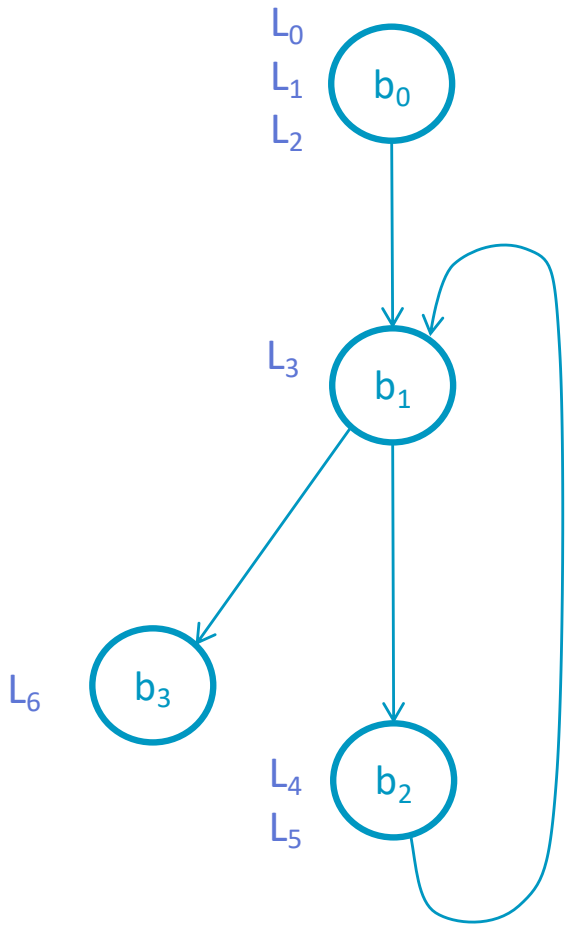
MAY analysis

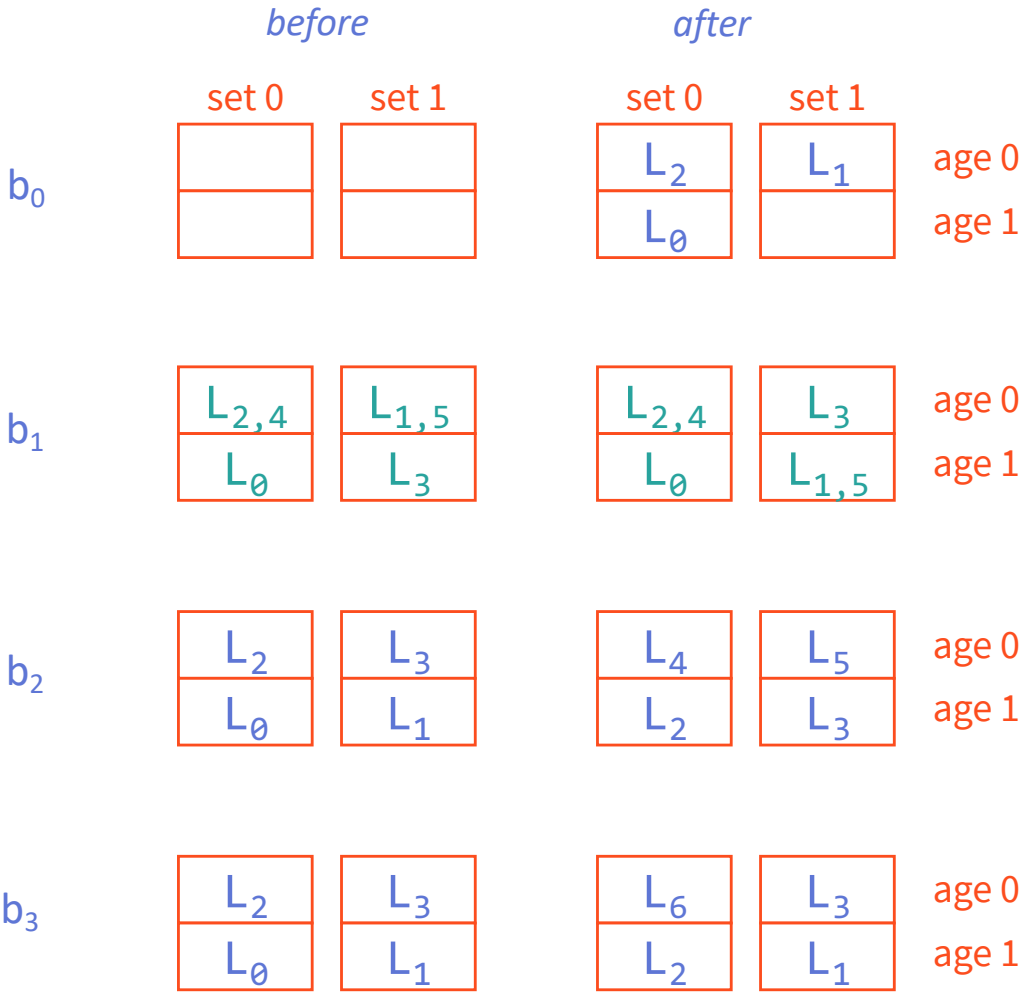


	before		*after*		
	set 0	set 1	set 0	set 1	
b_0			L_2 / L_0	L_1	age 0 / age 1
b_1	L_2 / L_0	L_1	L_2 / L_0	L_3 / L_1	age 0 / age 1
b_2	L_2 / L_0	L_3 / L_1	L_4 / L_2	L_5 / L_3	age 0 / age 1
b_3	L_2 / L_0	L_3 / L_1	L_6 / L_2	L_3 / L_1	age 0 / age 1

Cache analysis by Abstract Interpretation

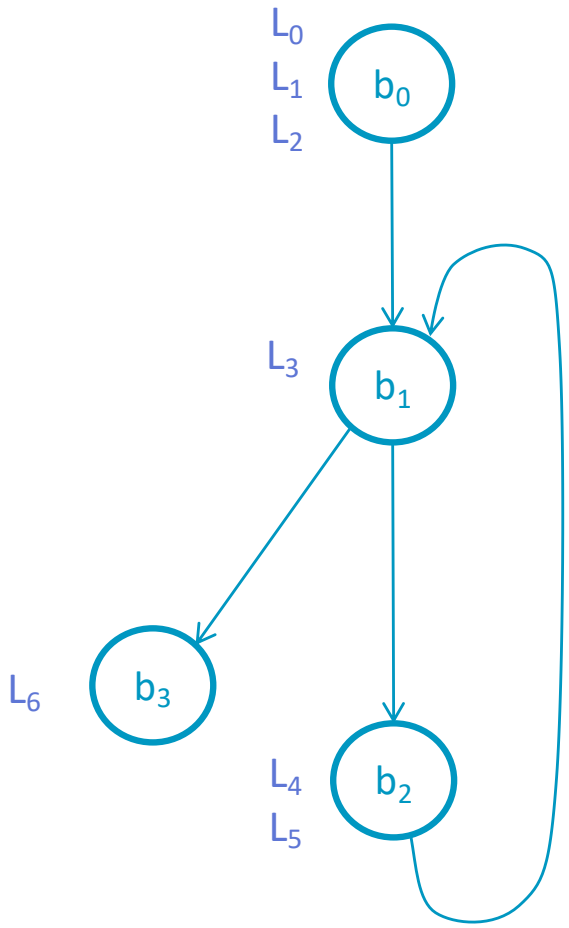
MAY analysis





Cache analysis by Abstract Interpretation

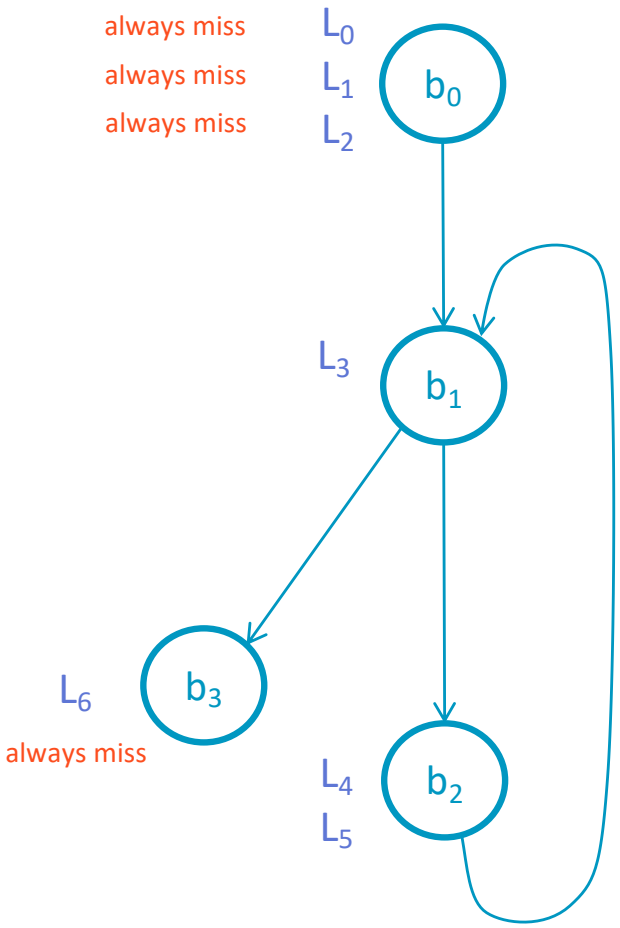
MAY analysis



	before set 0	before set 1	after set 0	after set 1	
b_0			L_2	L_1	age 0
			L_0		age 1
b_1	$L_{2,4}$	$L_{1,5}$	$L_{2,4}$	L_3	age 0
	L_0	L_3	L_0	$L_{1,5}$	age 1
b_2	$L_{2,4}$	L_3	L_4	L_5	age 0
	L_0	$L_{1,5}$	$L_{2,0}$	L_3	age 1
b_3	L_2	L_3	L_6	L_3	age 0
	L_0	L_1	L_2	L_1	age 1

Cache analysis by Abstract Interpretation

MAY analysis

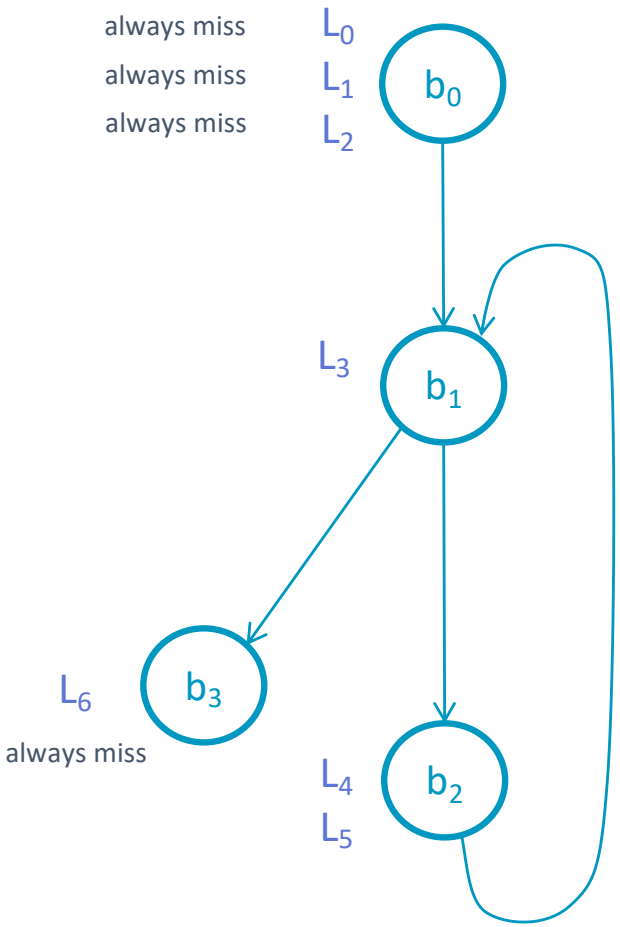


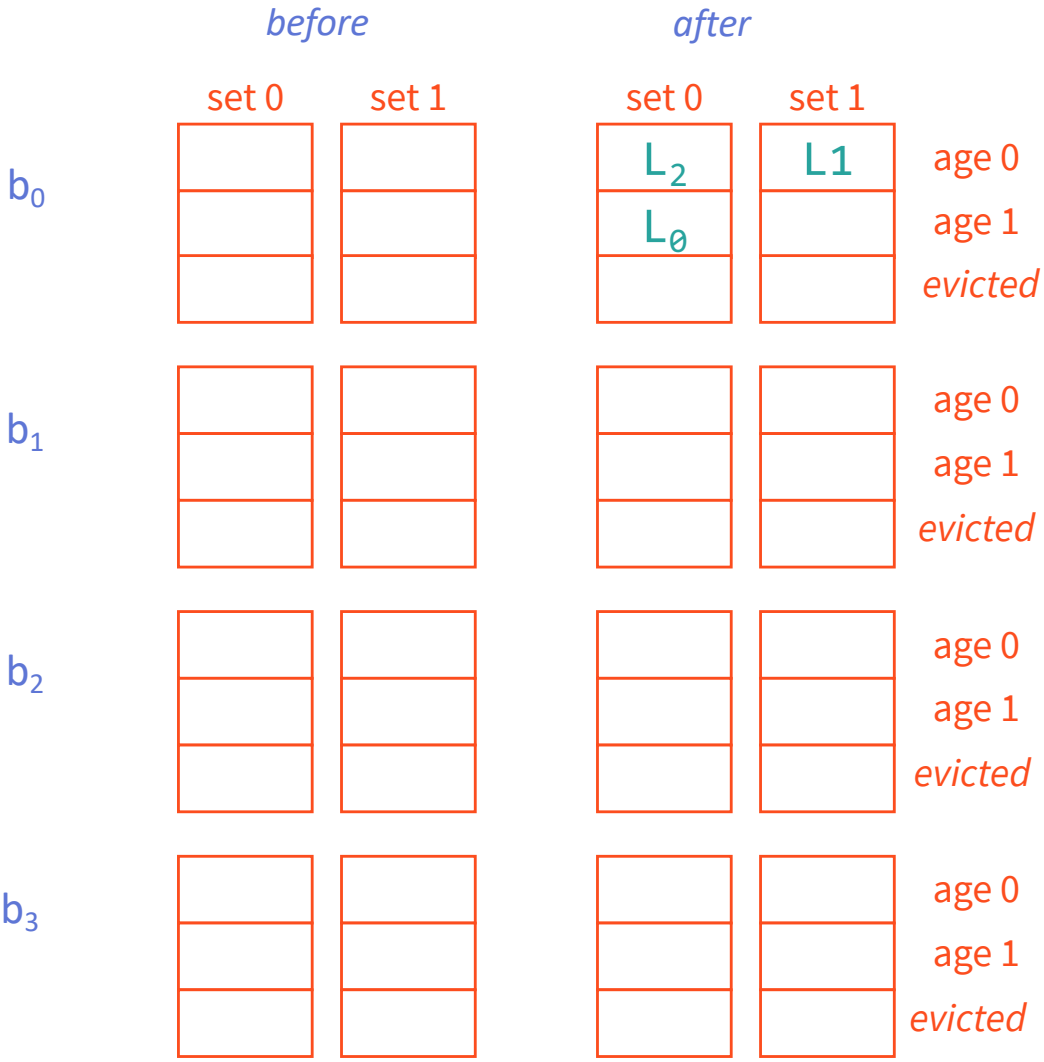
FIX POINT

	before		*after*		
	set 0	set 1	set 0	set 1	
b_0			L_2	L_1	age 0
			L_0		age 1
b_1	$L_{2,4}$	$L_{1,5}$	$L_{2,4}$	L_3	age 0
	L_0	L_3	L_0	$L_{1,5}$	age 1
b_2	$L_{2,4}$	L_3	L_4	L_5	age 0
	L_0	$L_{1,5}$	$L_{2,0}$	L_3	age 1
b_3	$L_{2,4}$	L_3	L_6	L_3	age 0
	L_0	$L_{1,5}$	$L_{2,4}$	$L_{1,5}$	age 1

Cache analysis by Abstract Interpretation

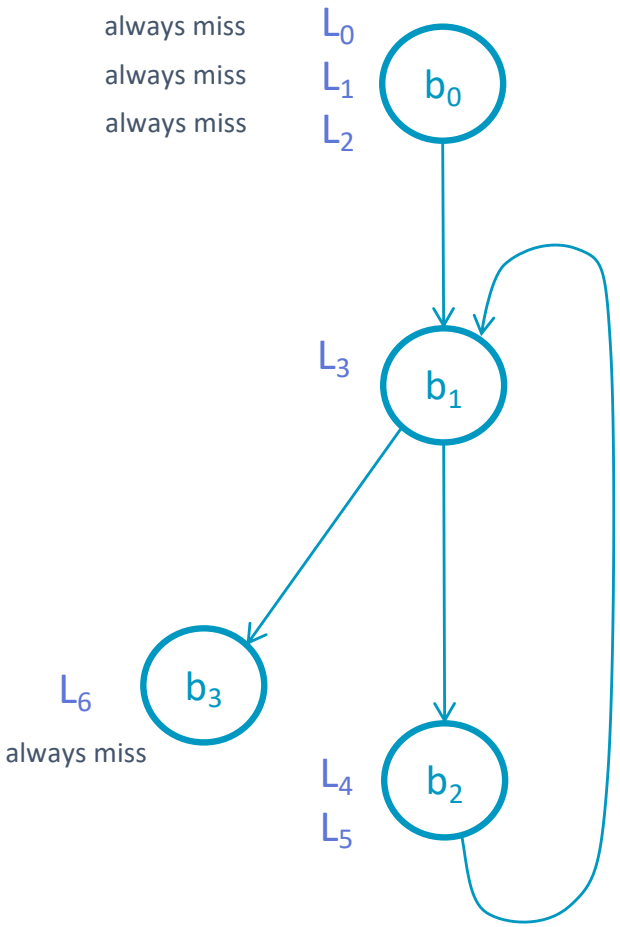
PERSISTENCE analysis

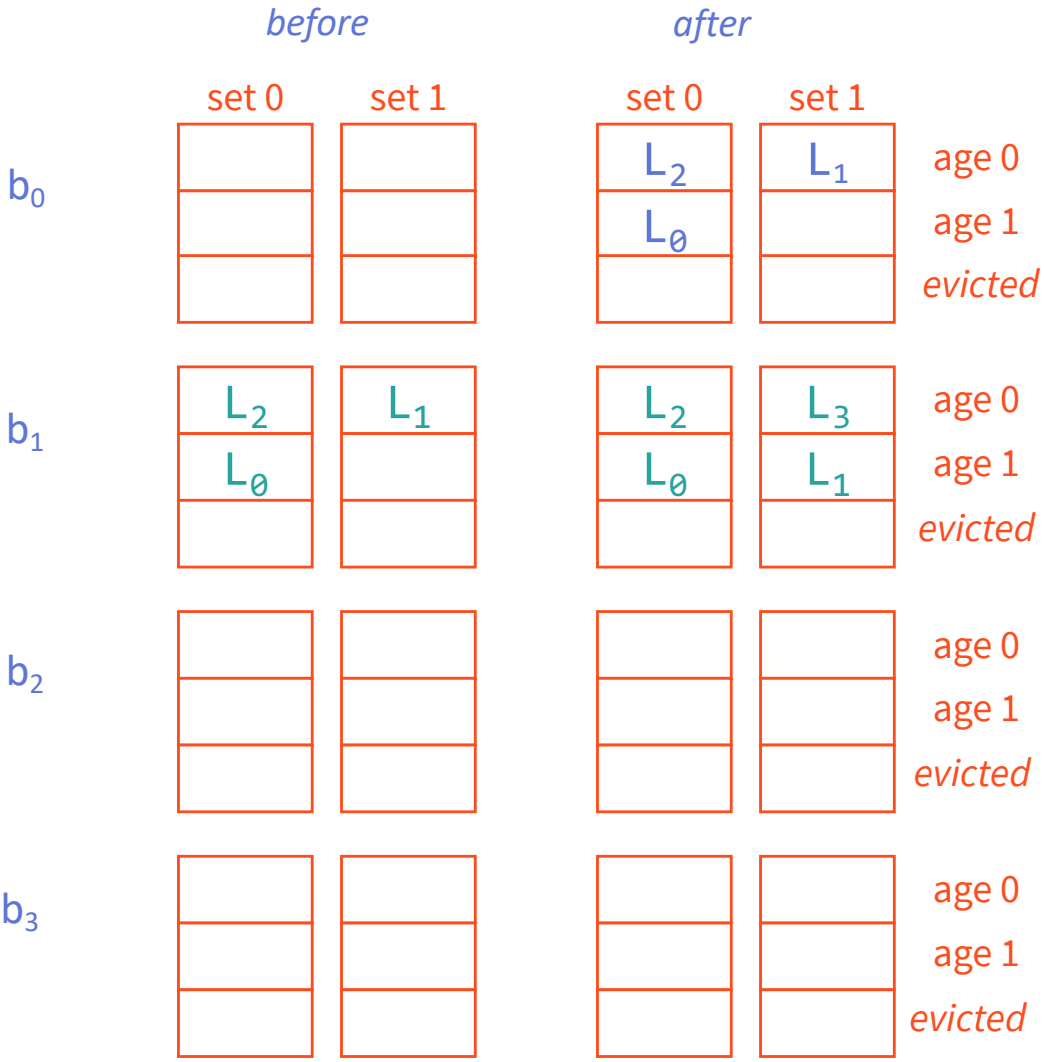




Cache analysis by Abstract Interpretation

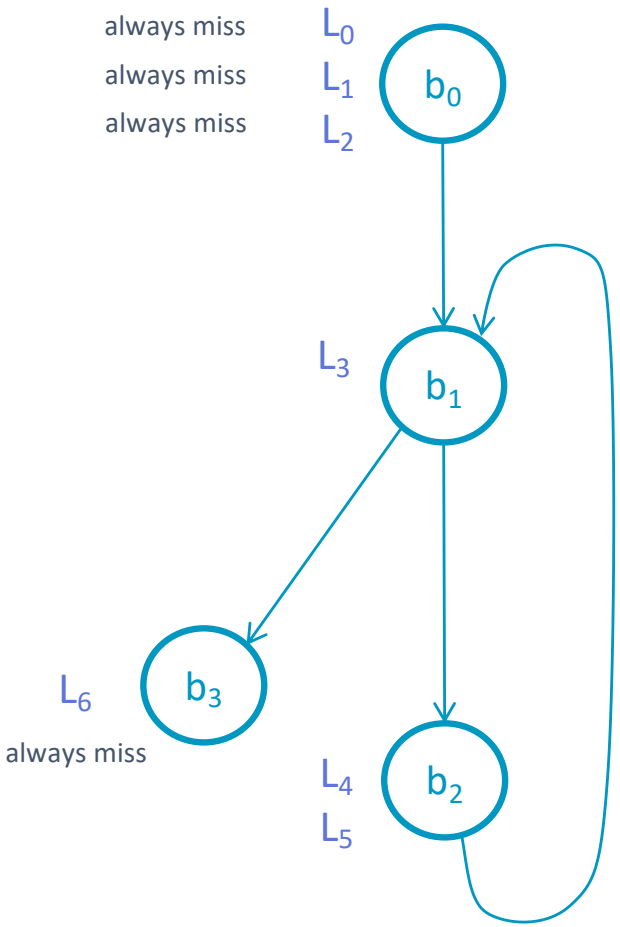
PERSISTENCE analysis

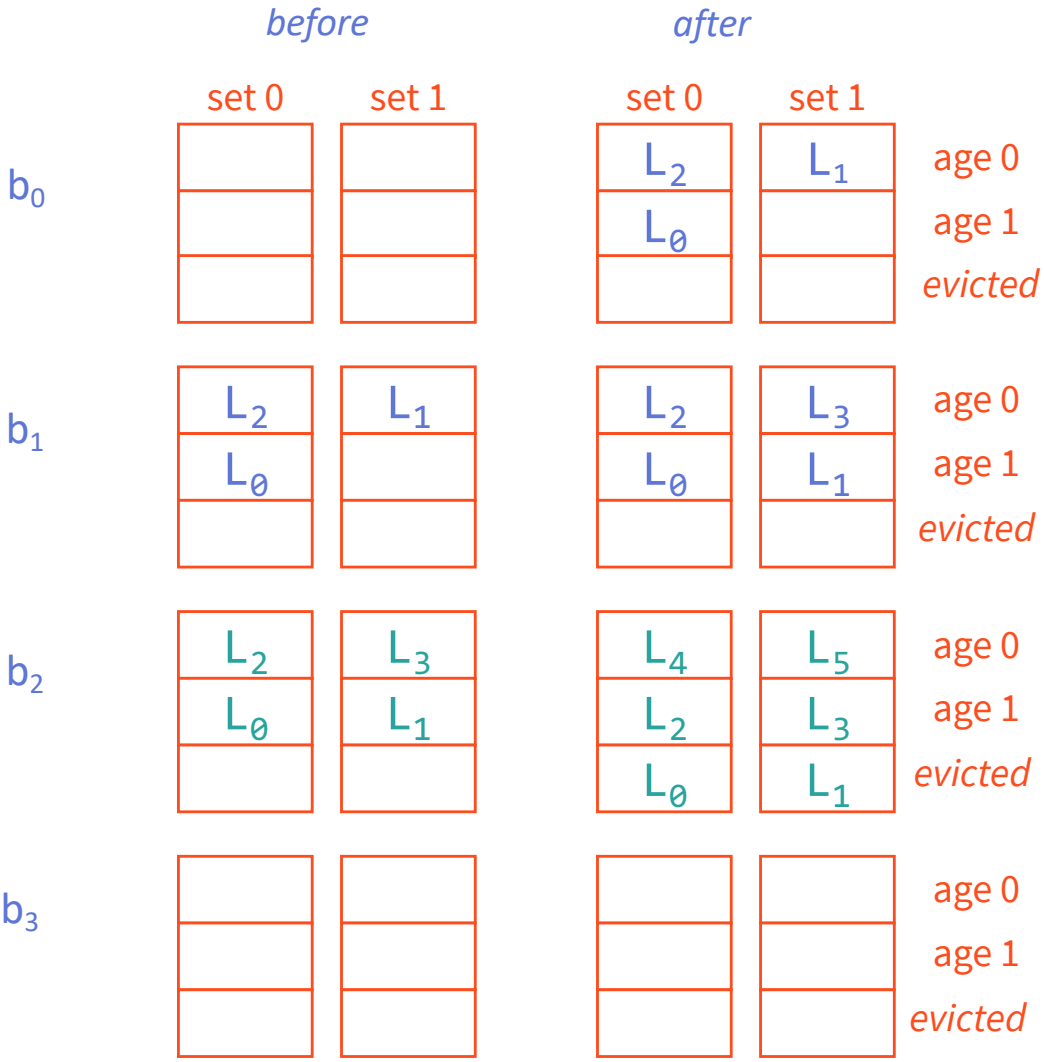




Cache analysis by Abstract Interpretation

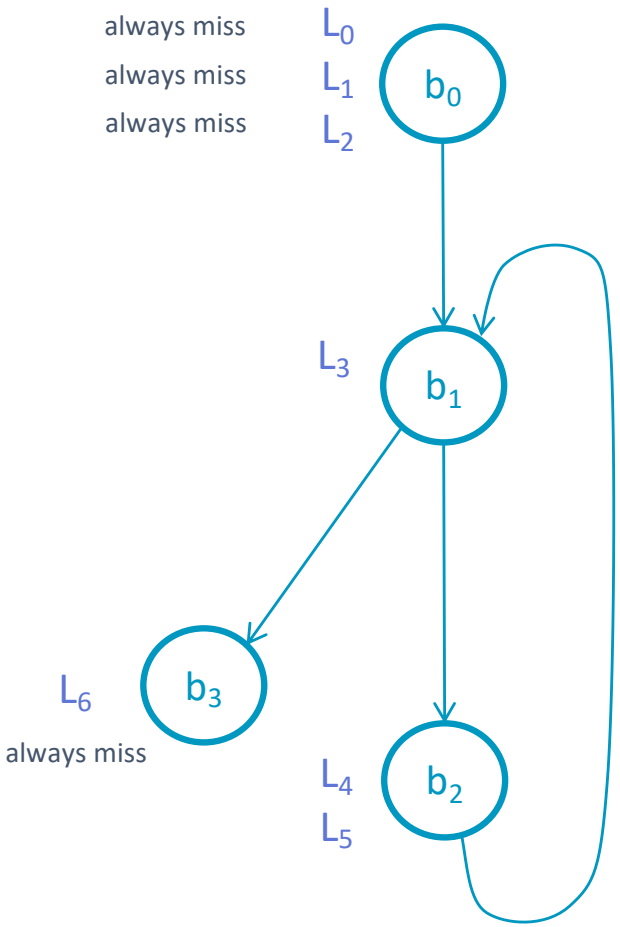
PERSISTENCE analysis

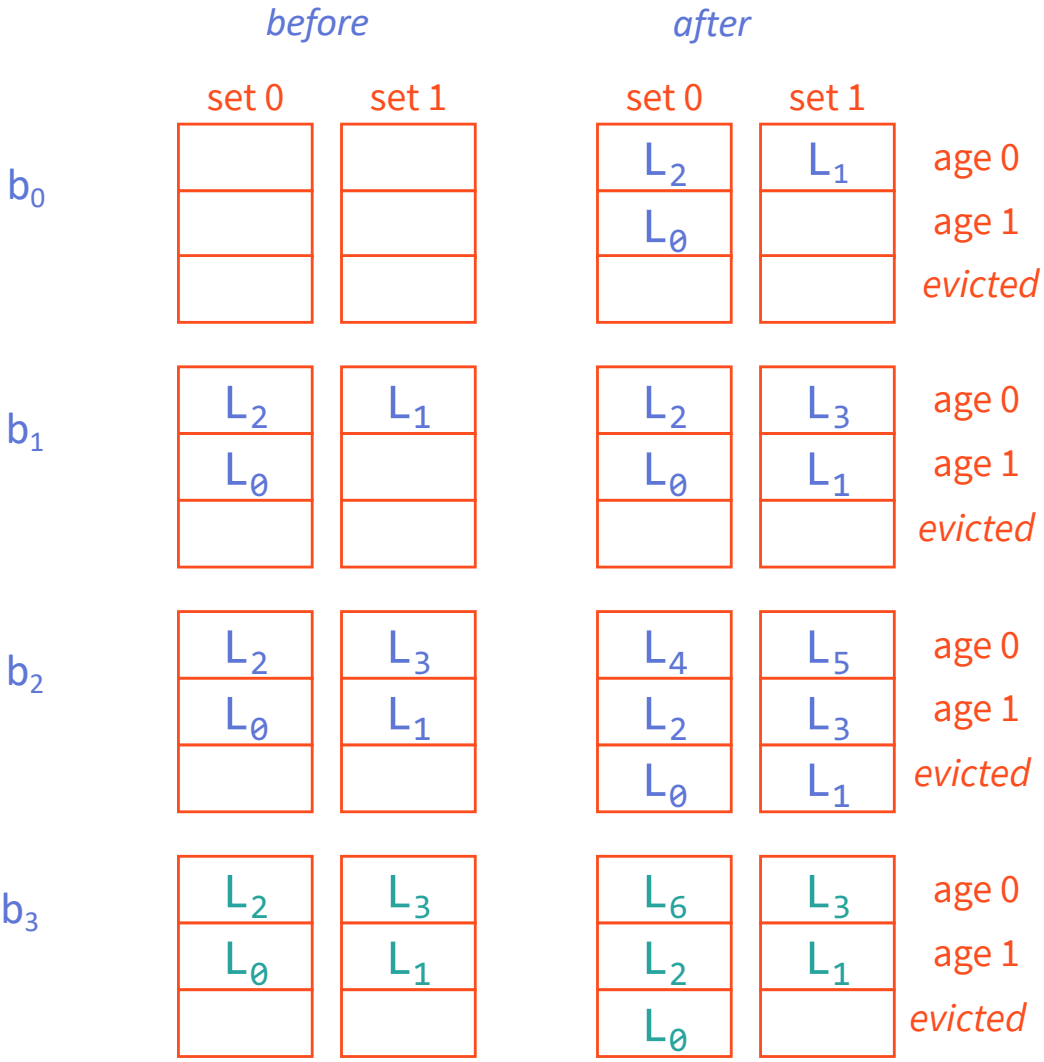




Cache analysis by Abstract Interpretation

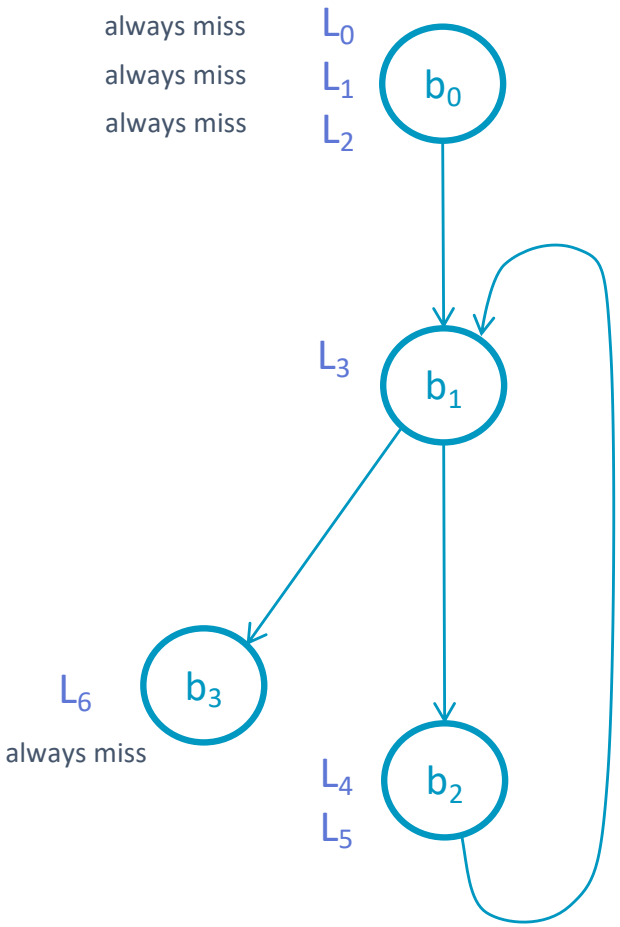
PERSISTENCE analysis

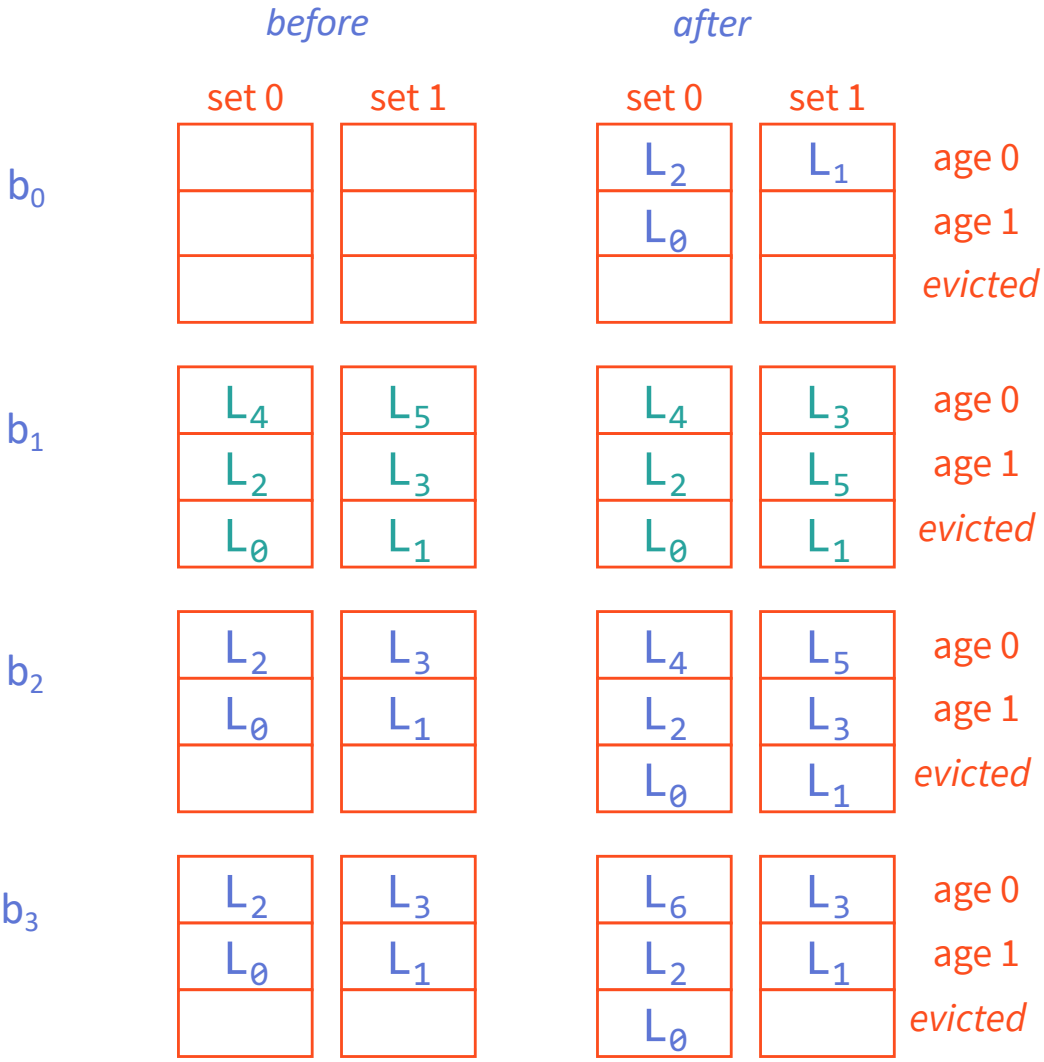




Cache analysis by Abstract Interpretation

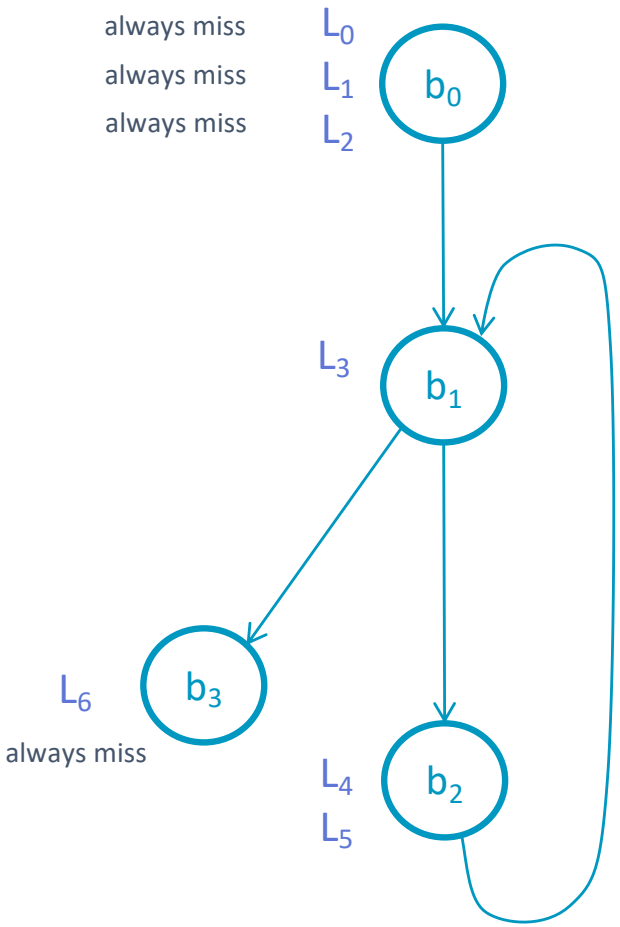
PERSISTENCE analysis

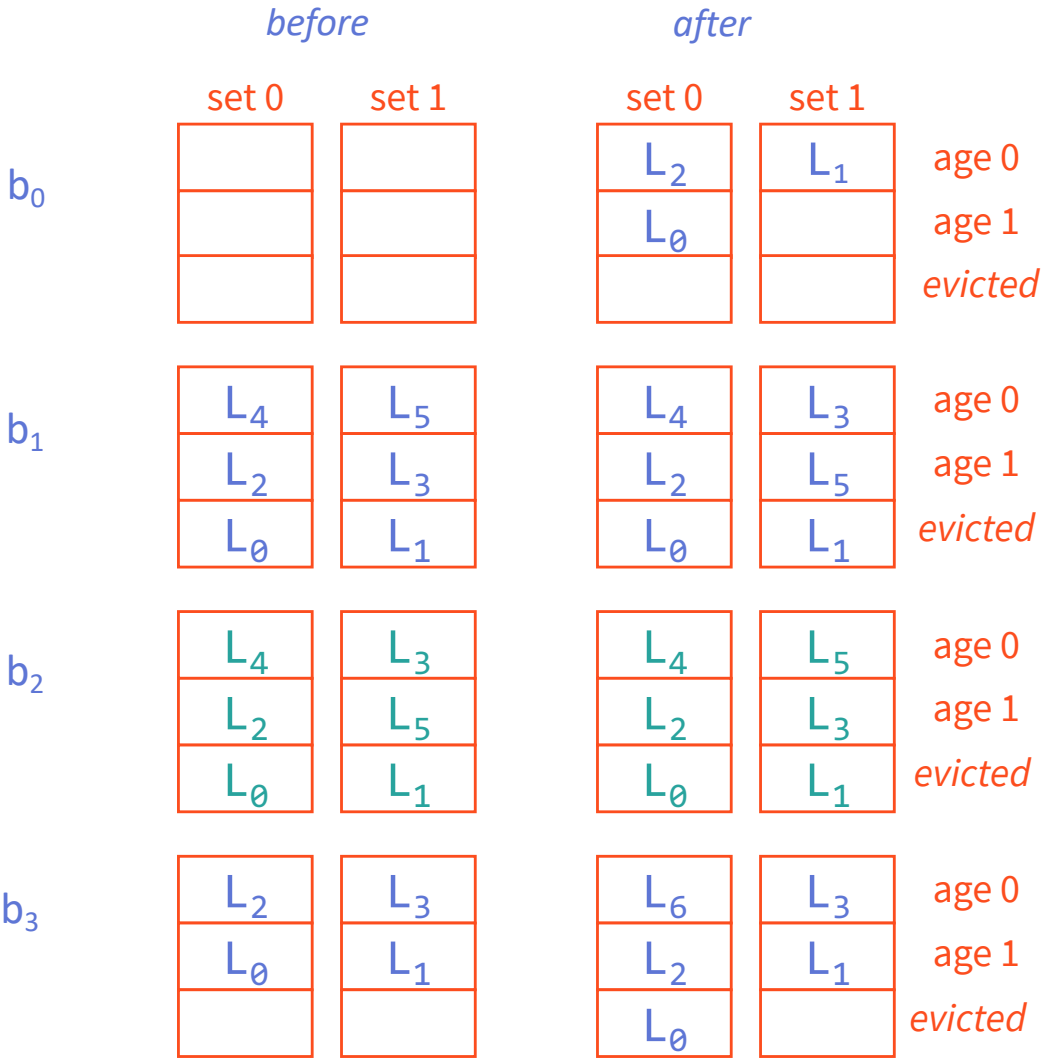




Cache analysis by Abstract Interpretation

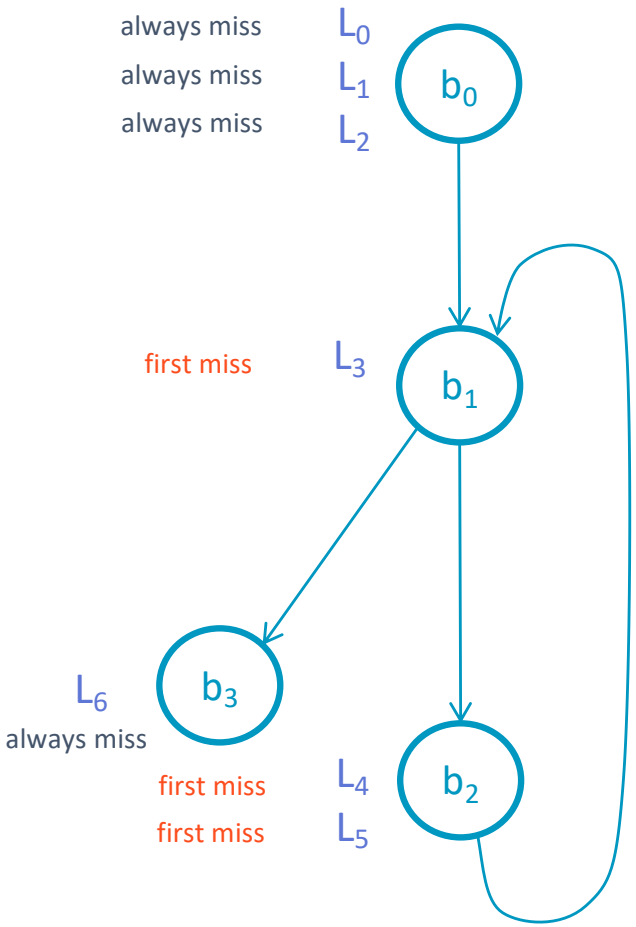
PERSISTENCE analysis



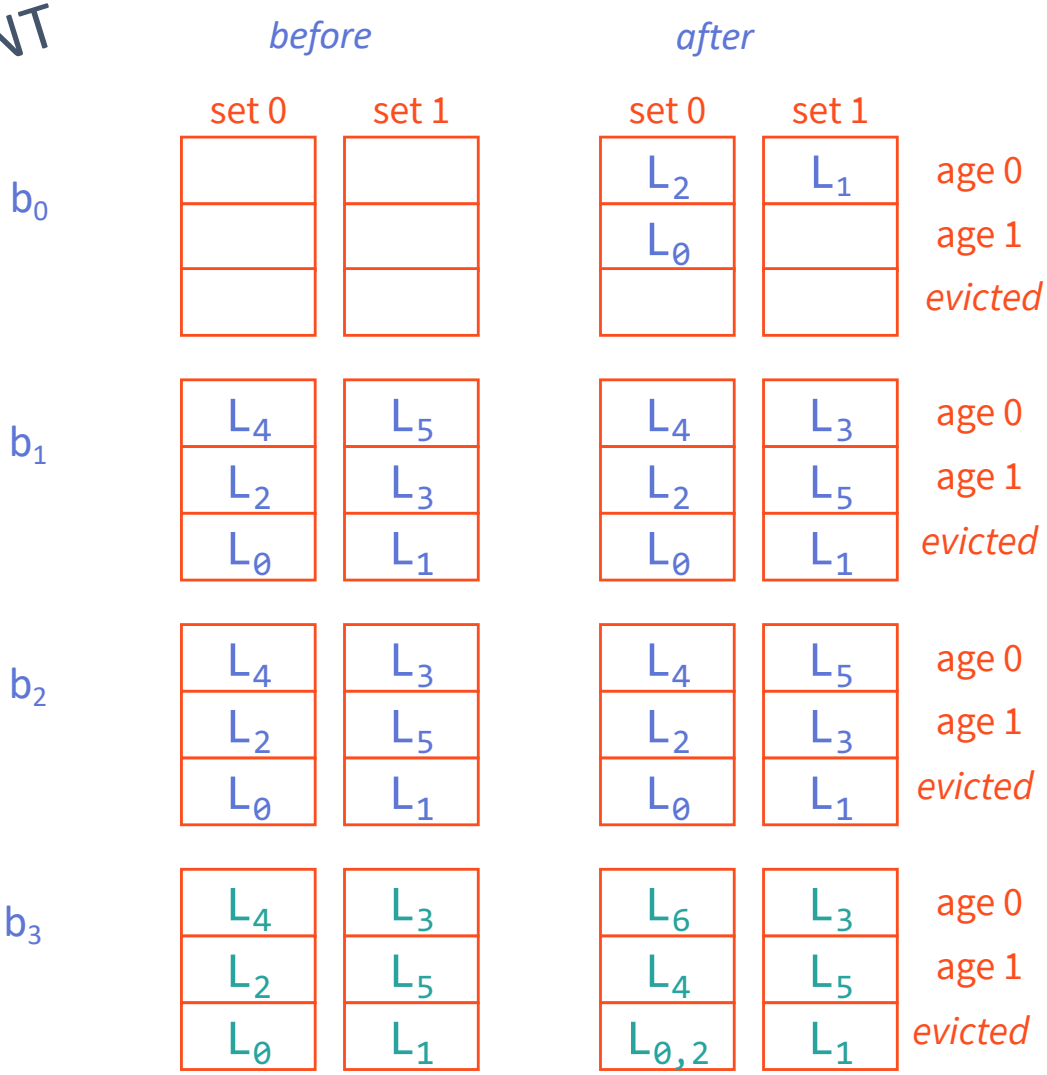


Cache analysis by Abstract Interpretation

PERSISTENCE analysis



FIX POINT



New ILP formulation

$$\max T = 13\,x_0 + 2\,x_{0\text{-}1} + 5\,x_{1\text{-}2} + 3\,x_{2\text{-}1} + 8\,x_{1\text{-}3} + \text{misspenalty} * (3\,x_0 + 1 + x_3 + 2)$$

$$x_0 = 1$$

$$x_0 = x_{0\text{-}1}$$

$$x_1 = x_{0\text{-}1} + x_{2\text{-}1}$$

$$x_1 = x_{1\text{-}2} + x_{1\text{-}3}$$

$$x_2 = x_{1\text{-}2}$$

$$x_2 = x_{2\text{-}1}$$

$$x_3 = x_{1\text{-}3}$$

$$x_3 = 1$$

$$x_{2\text{-}1} \leq 256$$