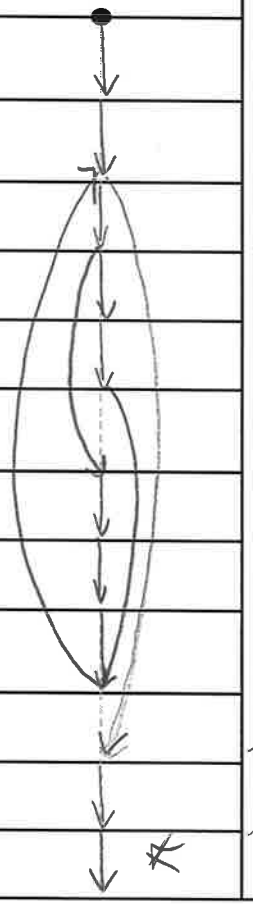
Program	**Abstraction** No predicate	**Boolean MC**
1. a = 8		
2. b = -2		
3. while (a > 6){		
4. if (b == 0) {		
5. a = 6		
6. } else {		
7. a = a + 1		
8. b = b + 2		
9. }		
10. }		
11. b = b * (-1)		
12. assert(b == 0)		

Program	Abstraction No predicate	Boolean MC
1. a = 8	skip	
2. b = -2	skip	
3. while (a > 6){	while(*){	
4. if (b == 0) {	if (*) {	
5. a = 6	skip	
6. } else {	} else {	
7. a = a + 1	skip	
8. b = b + 2	skip	
9. }	}	
10. }	}	
11. b = b * (-1)	skip	
12. assert(b == 0)	assert(*)	



Counterexample:
1, 2, 3, 11, 12,

{8 ≤ 6} → False

1: a=8
{2 ≠ 0 ∧ a ≤ 6}
2: b=-2
{-b ≠ 0 ∧ a ≤ 6}
3: assume(a ≤ 6)
{-b ≠ 0}
11: b = b*(-1)
{b ≠ 0}
12: assume(b ≠ 0)

Learned Predicate:
a ≤ 6

Program	Abstraction p: b<-2	Boolean MC	Abstraction p: b<-2 q:	Boolean MC
1. a = 8				
2. b = -2				
3. while (a > 6){				
4. if (b == 0) {				
5. a = 6				
6. } else {				
7. a = a + 1				
8. b = b + 2				
9. }				
10. }				
11. b = b * (-1)				
12. assert(b == 0)				

Program	Abstraction p: b<-2	Boolean MC p ¬p	Abstraction p: b<-2 q: a > 6	Boolean MC p∧q p∧¬q ¬p∧q ¬p∧¬q
1. a = 8	skip		q = True	
2. b = -2	p = False		p = False	
3. while (a > 6){	while(*){		while (q) {	
4. if (b == 0) {	if(p? F:*){		if(p ? F : *) {	
5. a = 6	skip		q = False	
6. } else {	} else {		} else {	
7. a = a + 1	skip		q = q ? T : *	
8. b = b + 2	p = p ? * : F		p = p ? * : F	
9. }	}		}	
10. }	}		}	
11. b = b * (-1)	p = p ? F : *		p = p ? F : *	
12. assert(b == 0)	assert(p ? F : *)	× *	assert(p ? F : *);	× *

Counterexample 1:
1, 2, 3, 11, 12

$\{8 \leq 6\}$ → False!

1. $a = 8$
   $\{2 \neq 0 \wedge \underline{a \leq 6}\}$
2. $b = -2$
   $\{b \neq 0 \wedge a \leq 6\}$
3. assume$(\underline{a \leq 6})$
   $\{b \neq 0\}$
11. $b = b * (-1)$
    $\{b \neq 0\}$
12. assume$(b \neq 0)$

Learned Predicate:
$a \leq 6$
(you can also use $a > 6$)

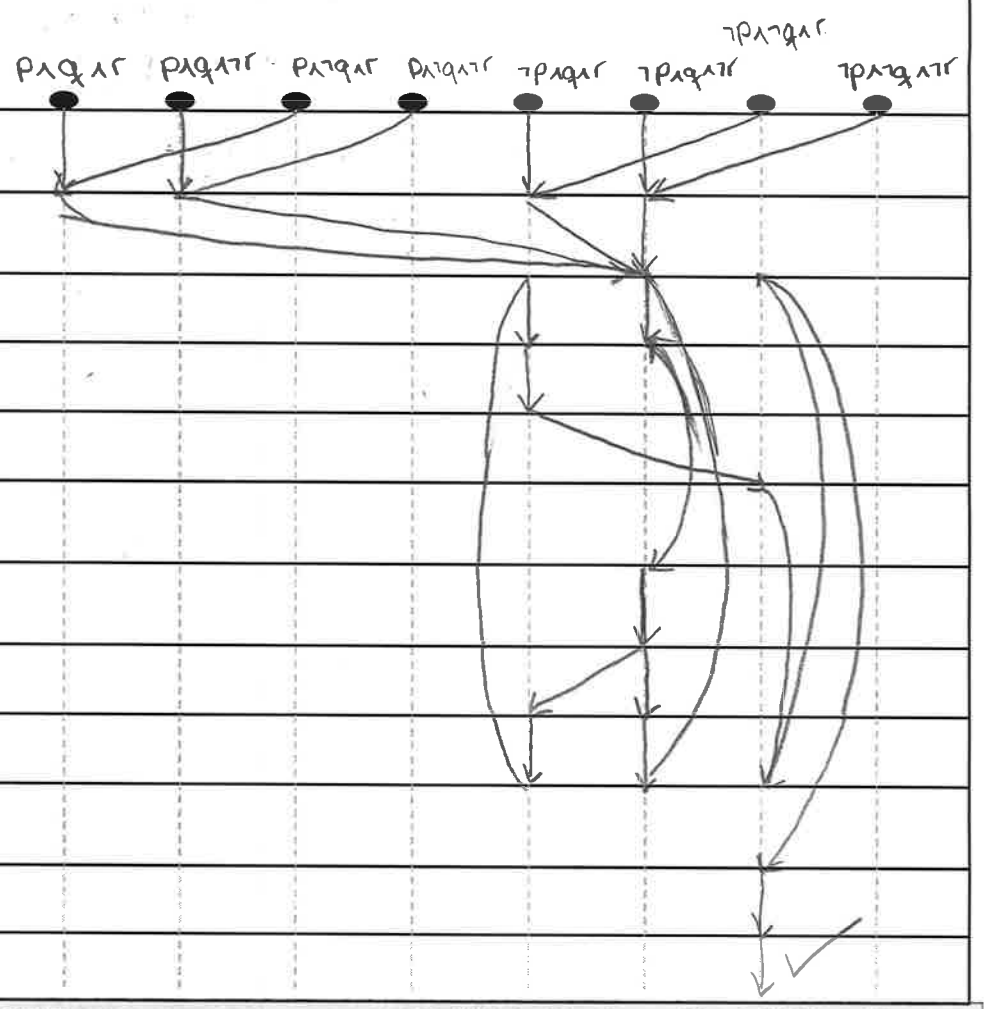
Counterexample 2:
1, 2, 3, 4, 5, 3, 11, 12

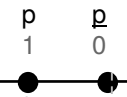
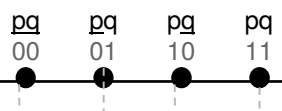
$\{\underline{-b \neq 0} \wedge \underline{b = 0}\}$ → False!

4: assume$(\underline{b = 0})$
   $\{\underline{-b \neq 0} \wedge a \leq 6\}$
5: $a = 6$
   $\{\underline{-b \neq 0} \wedge a \leq 6\}$
3: assume$(a \leq 6)$
   $\{\underline{-b \neq 0}\}$
11: $b = b * (-1)$
    $\{b \neq 0\}$
12: assume$(b \neq 0)$

learned predicate:
$b = 0$
(you can also use $b \neq 0$ or $-b \neq 0$)

Program	Abstraction p: b<-2 q: r:	Boolean MC
1. a = 8		
2. b = -2		
3. while (a > 6){		
4. if (b == 0) {		
5. a = 6		
6. } else {		
7. a = a + 1		
8. b = b + 2		
9. }		
10. }		
11. b = b * (-1)		
12. assert(b == 0)		

Program	Abstraction p: b<-2 q: a > 6 r: b = 0	Boolean MC
1. a = 8	q = True	
2. b = -2	p = False; r = False;	
3. while (a > 6){	while (q) {	
4. if (b == 0) {	if (r) {	
5. a = 6	q = False	
6. } else {	} else {	
7. a = a + 1	q = q ? T : *	
8. b = b + 2	p = p ? * : F r = p ? F : (r ? F : *)	
9. }	}	
10. }	}	
11. b = b * (-1)	r = r ? T : F p = p ? F : (r ? F : *)	
12. assert(b == 0)	assert (r)	

Program	Abstraction p: y < 44	Boolean MC 	Abstraction p: y < 44 q:	Boolean MC 
y = 22				
x = 12				
z = x*x+1				
if (x <= 0) {				
if (y>42) {				
y = y - x				
} else {				
y = 42				
}				
}				
assert (y<44)				

	Program	Abstraction p: y < 44	Boolean MC	Abstraction p: y < 44 q: x ≤ 0	Boolean MC
			p 1 · p 0		pq 00 · pq 01 · pq 10 · pq 11
1	y = 22	p = True		p = True	
2	x = 12	skip		q = False	
3	z = x*x+1	skip		skip	
4	if (x <= 0) {	if (*) {		if (q) {	
5	if (y>42) {	if (p ? * : T) {		if (p ? * : T) {	
6	y = y - x	p = *		p = (p ∧ ¬q) ? T : ((¬p ∧ q) ? F : *)	
7	} else {	} else {		} else {	
8	y = 42	p = True		p = True	
9	}	}		}	
10	}	}		}	
11	assert (y<44)	assert (p)	✓ ✗	assert (p)	✓

Counterexample 1:

1, 2, 3, 4, 5, 6, 11

$\{y - 12 \geq 44 \wedge y > 42 \wedge \underline{12 \leq 0}\}$

2 $x = 12$

$\{y - x \geq 44 \wedge y > 42 \wedge \underline{x \leq 0}\}$

3 $z = x * x + 1$

$\{y - x \geq 44 \wedge y > 42 \wedge \underline{x \leq 0}\}$

4 assume$(\underline{x \leq 0})$

$\{y - x \geq 44 \wedge y > 42\}$

5 assume$(y > 42)$

$\{y - x \geq 44\}$

6 $y = y - x$

$\{y \geq 44\}$

11 assume$(y \geq 44)$

learned predicate:

$x \leq 0$

(could also learn $x > 0$)