

IC637 Program Analysis

Lecture 2 Review

Minseok Jeon

2025 Fall

Question : How static analysis can be used in practice?

- If we use expressive abstract domain such as interval domain, we can find buffer overflow. For example we can find potential buffer overflow in the following code.

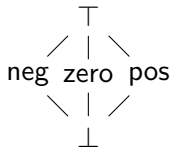
```
#include <stdio.h>
#include <string.h>

void vulnerable() {
    char buffer[8]; // small buffer
    printf("Enter your name: ");
    gets(buffer);    // dangerous!
    printf("Hello, %s\n", buffer);
}

int main() {
    vulnerable();
    return 0;
}
```

Sign Domain

- Abstract values :



- Abstract operations:

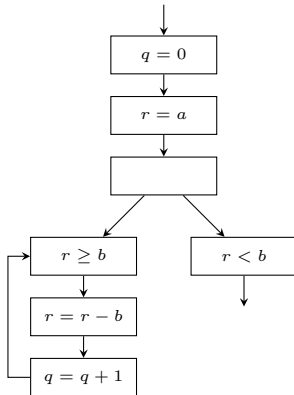
$\hat{+}$	$\top$	pos	zero	neg	$\perp$
$\top$					
pos					
zero					
neg					
$\perp$					

$\hat{-}$	$\top$	pos	zero	neg	$\perp$
$\top$					
pos					
zero					
neg					
$\perp$					

Example Program

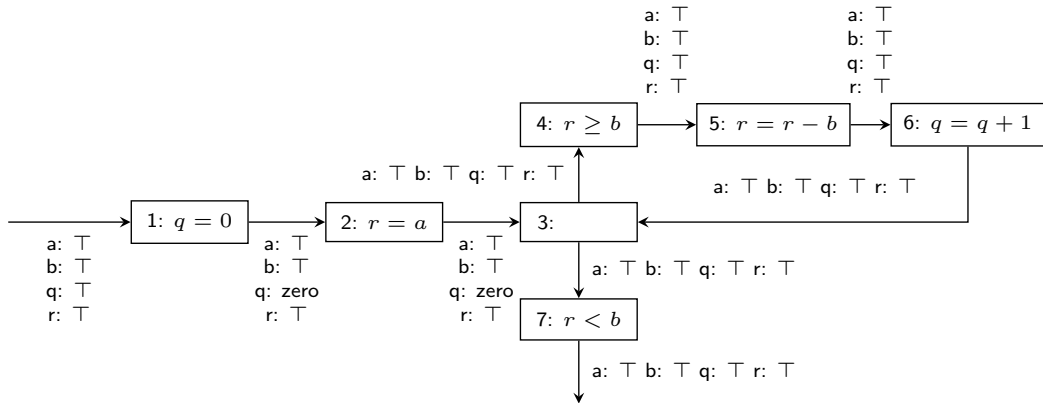
- Example Program

```
//a >= 0, b >= 0
q = 0;
r = a;
while(r >= b){
    r = r - b;
    q = q + 1;
}
assert(q >= 0);
assert(r >= 0);
```

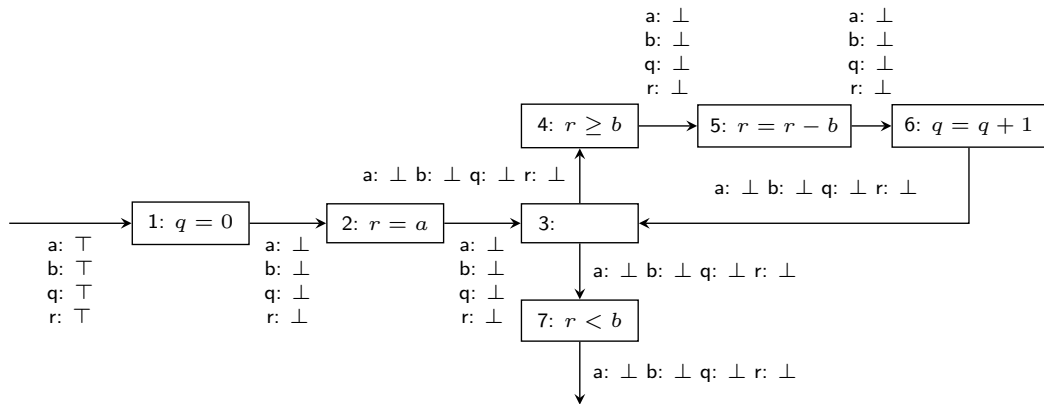


Fixed Point

- Executing any of the program statements does not change the abstract values

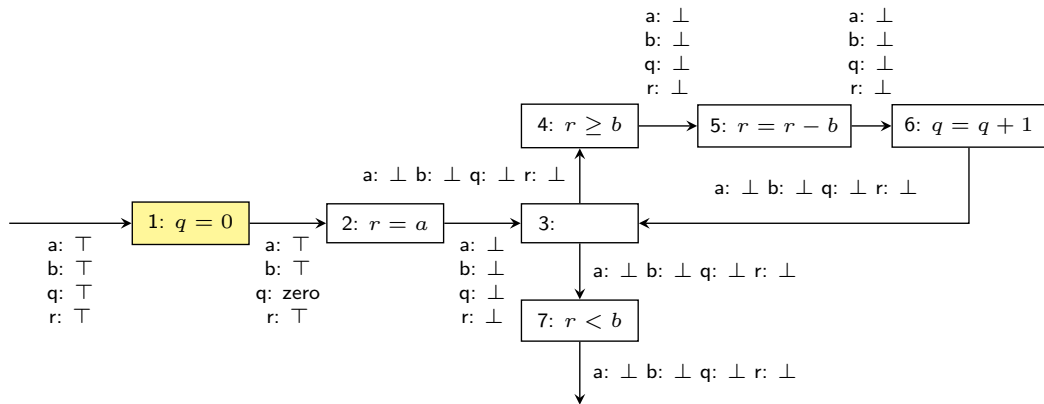


Fixed Point Computation (Worklist)



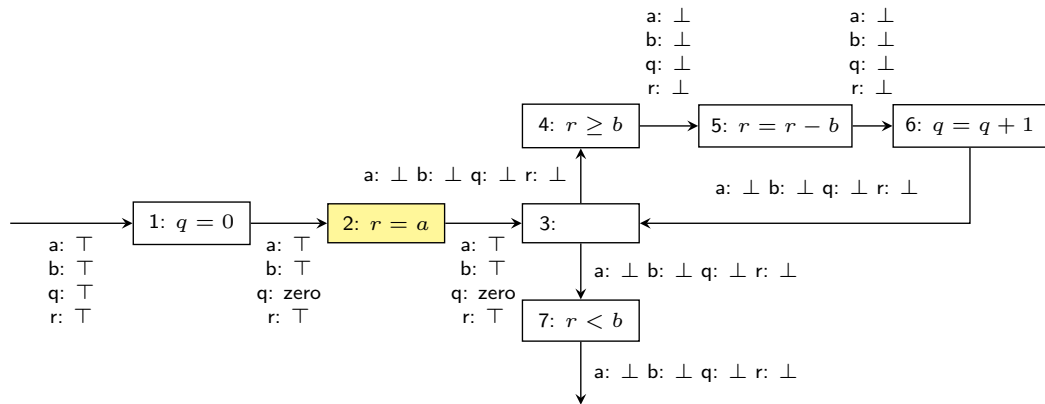
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Fixed Point Computation (Worklist)



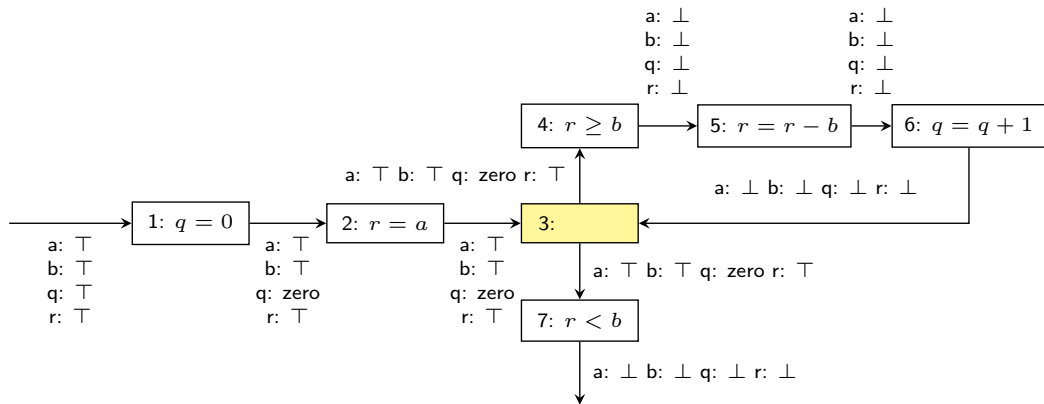
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Fixed Point Computation (Worklist)



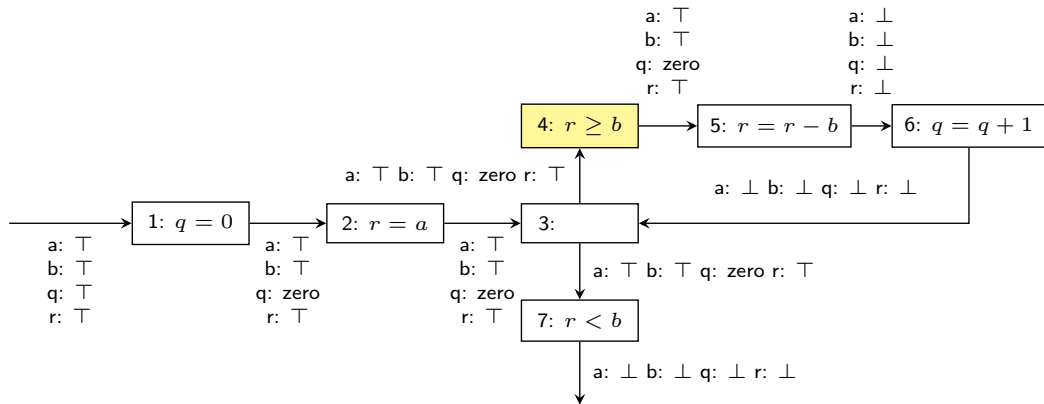
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Fixed Point Computation (Worklist)



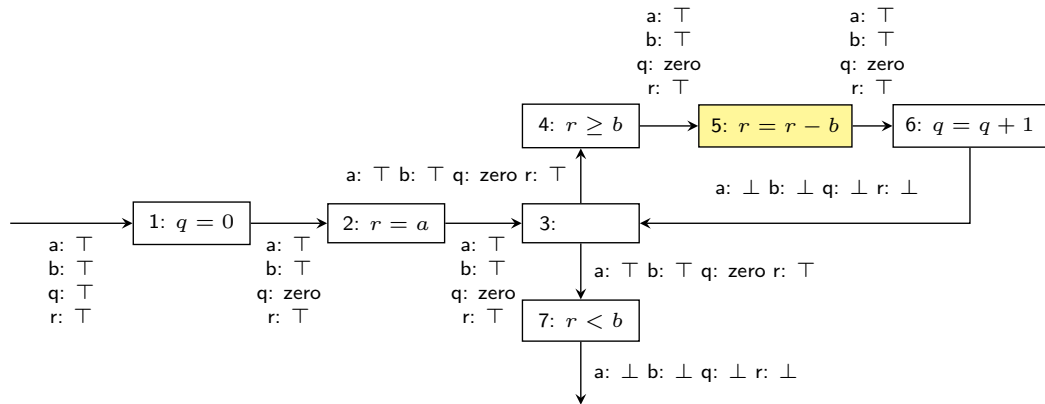
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Fixed Point Computation (Worklist)



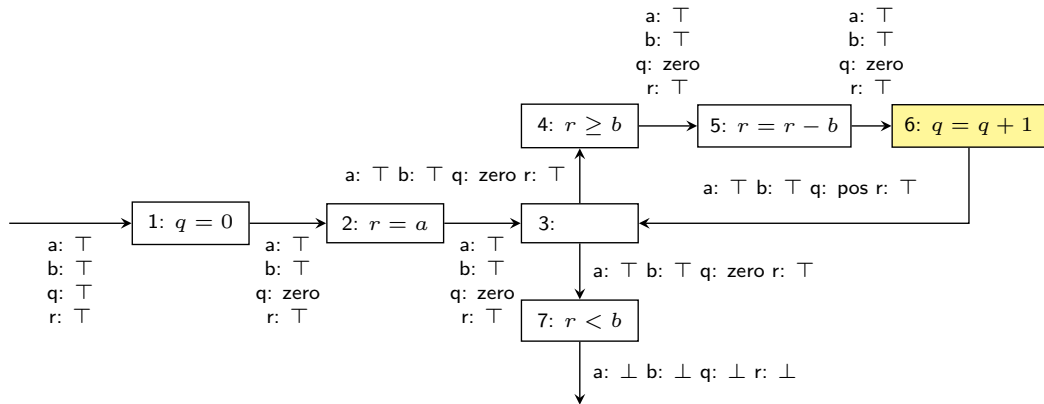
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Fixed Point Computation (Worklist)



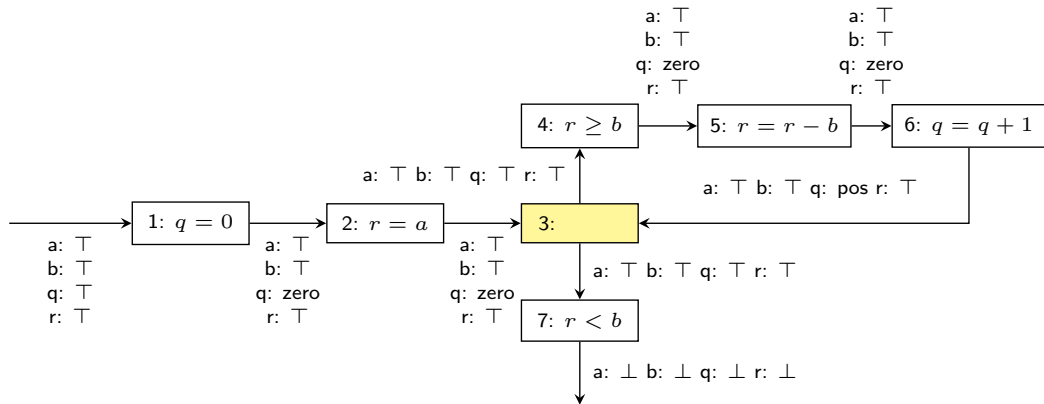
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Fixed Point Computation (Worklist)



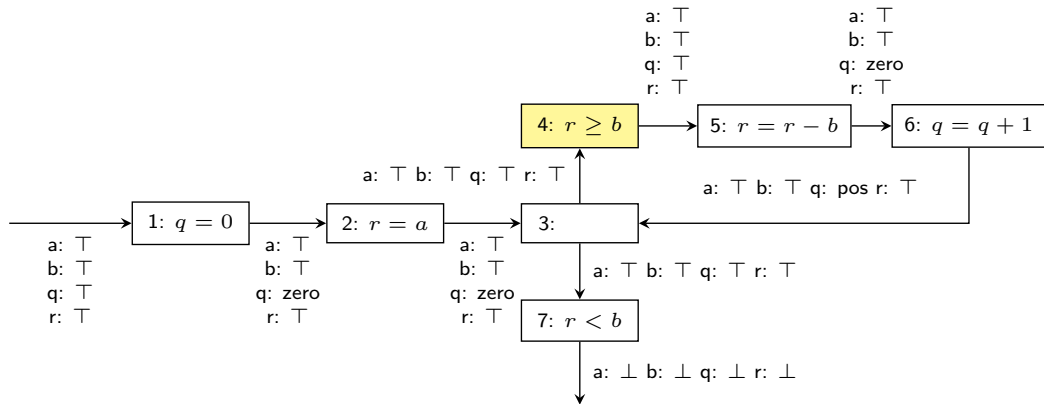
Worklist = $\{\cancel{1}, \cancel{2}, 3, \cancel{4}, \cancel{5}, \cancel{6}, 7\}$

Fixed Point Computation (Worklist)



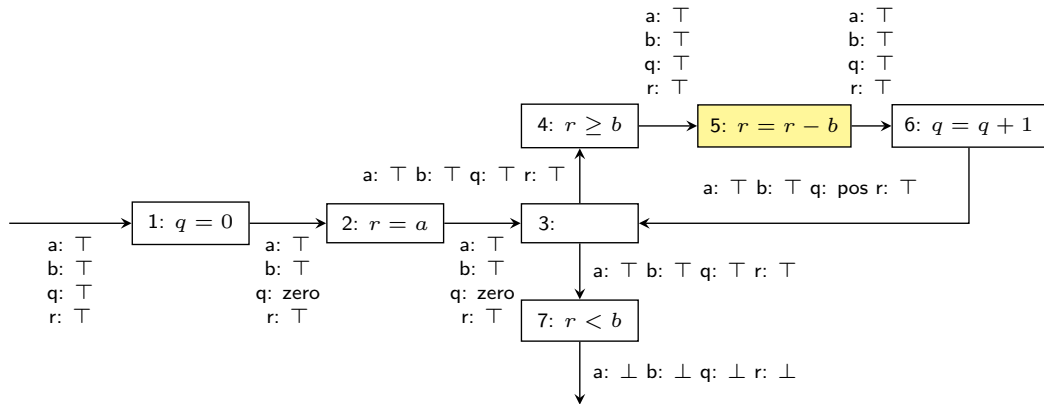
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Fixed Point Computation (Worklist)



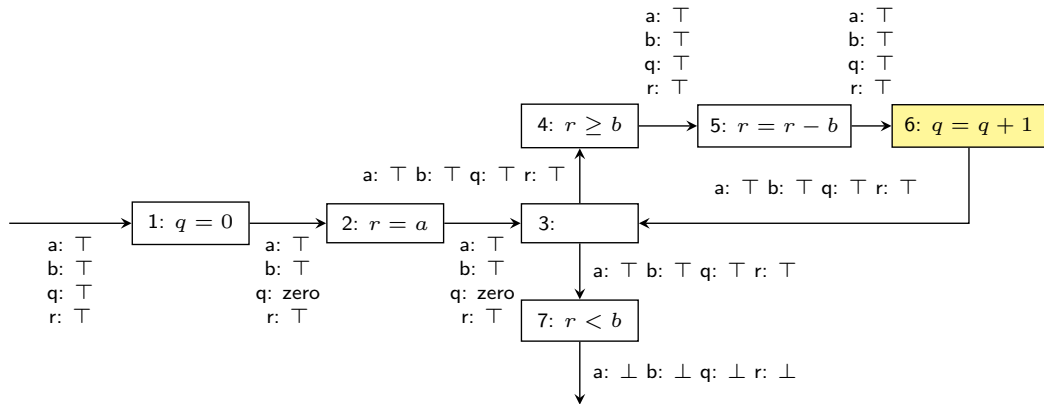
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Fixed Point Computation (Worklist)



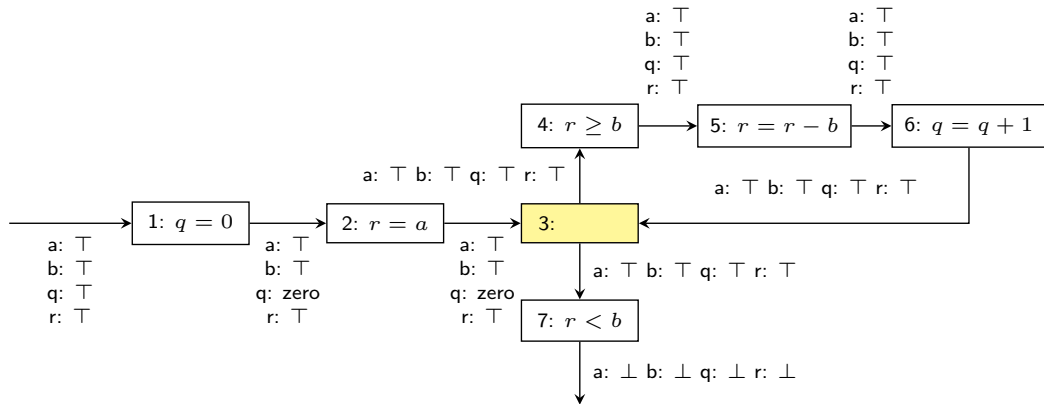
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Fixed Point Computation (Worklist)



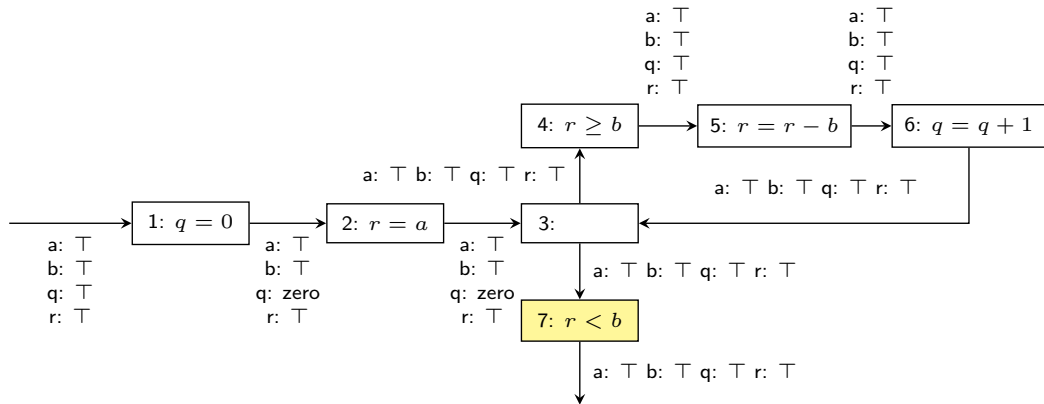
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Fixed Point Computation (Worklist)



Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

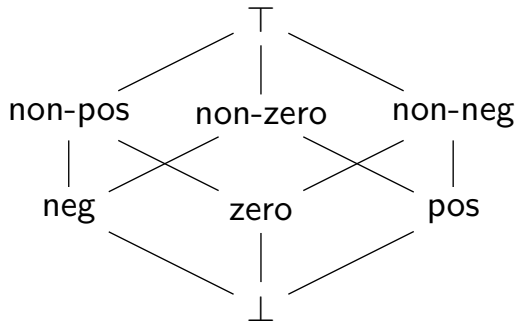
Fixed Point Computation (Worklist)



Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Extended Sign Domain

- Abstract values :



Extended Sign Domain

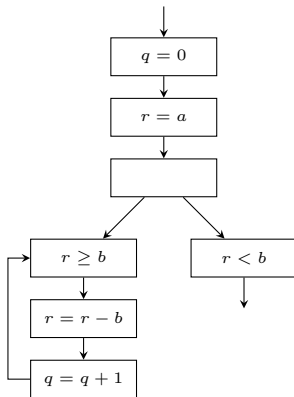
$\hat{+}$	$\top$	pos	zero	neg	non-pos	non-zero	non-neg	$\perp$
$\top$								
pos								
zero								
neg								
non-pos								
non-zero								
non-neg								
$\perp$								

$\hat{*}$	$\top$	pos	zero	neg	non-pos	non-zero	non-neg	$\perp$
$\top$								
pos								
zero								
neg								
non-pos								
non-zero								
non-neg								
$\perp$								

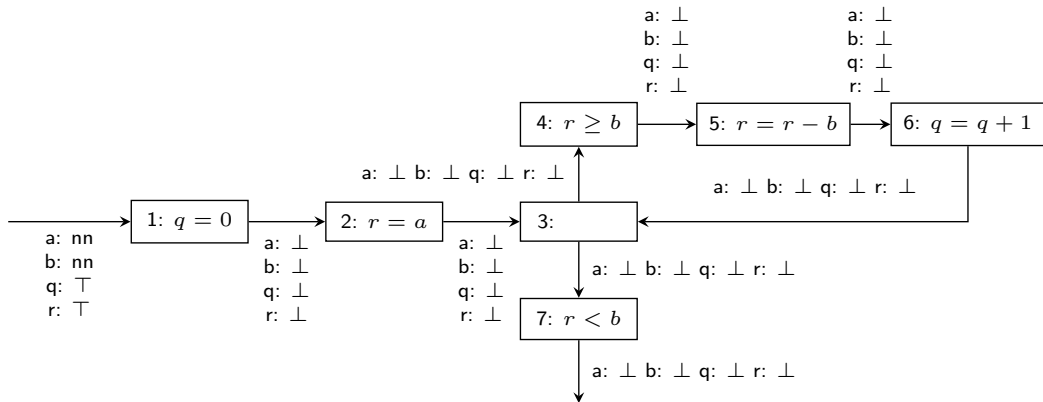
Extended Sign Domain

- Describe the result of the analysis with the extended sign domain

```
//a >= 0, b >= 0
q = 0;
r = a;
while(r >= b){
    r = r - b;
    q = q + 1;
}
assert(q >= 0);
assert(r >= 0);
```

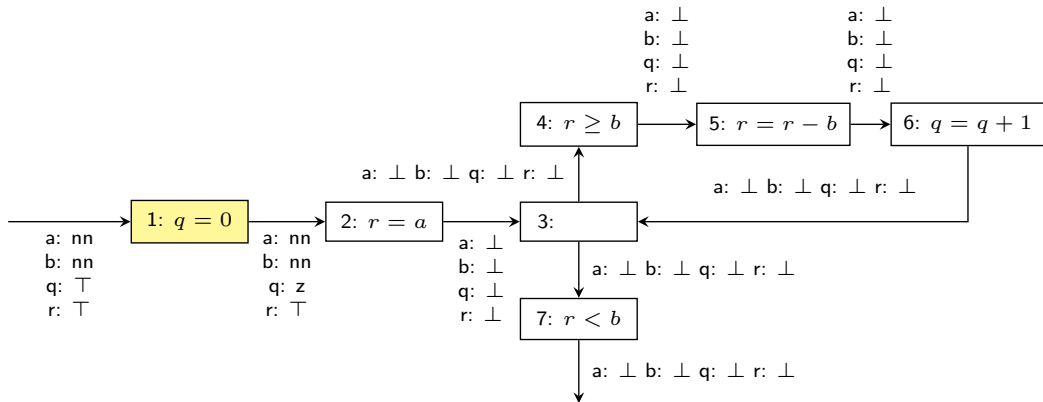


Extended Sign Domain



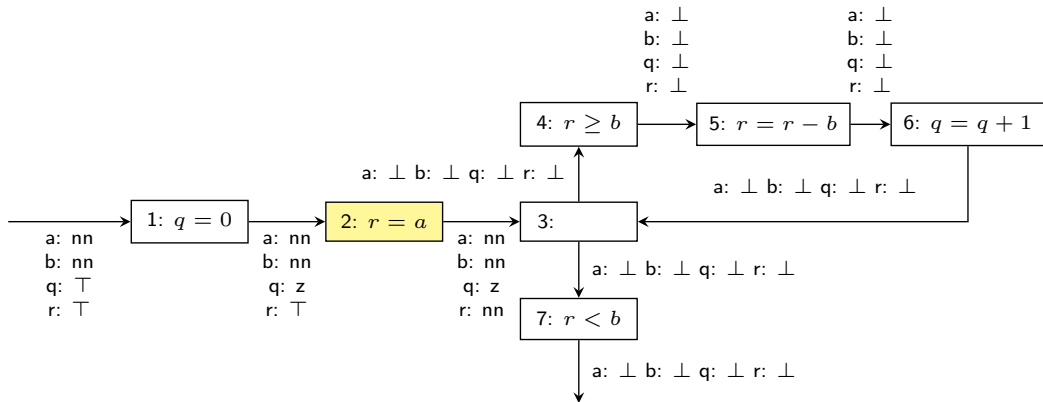
Worklist = {1, 2, 3, 4, 5, 6, 7}

Extended Sign Domain



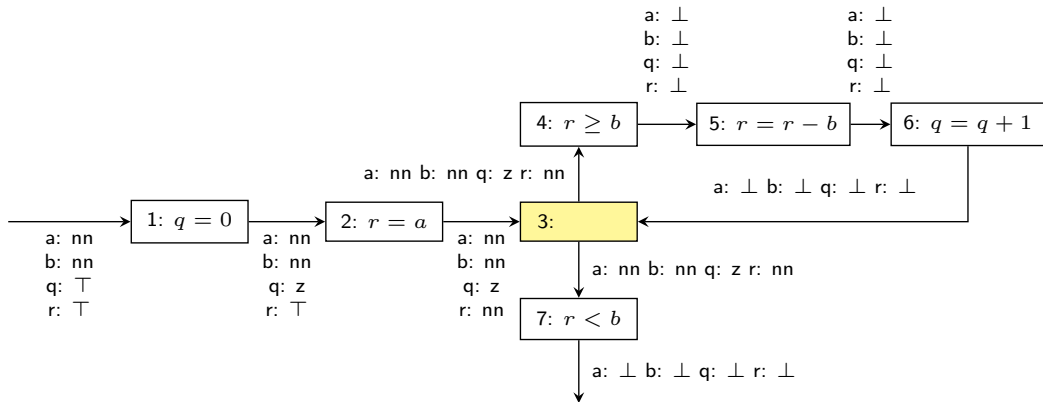
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Extended Sign Domain



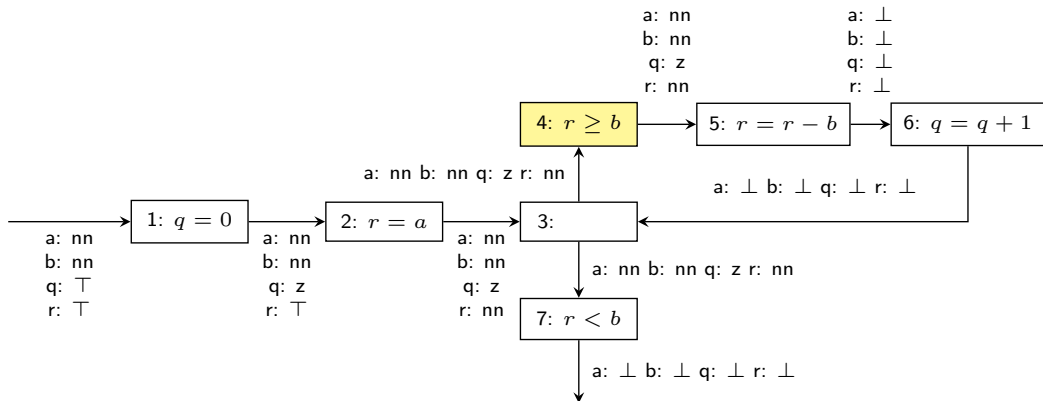
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Extended Sign Domain



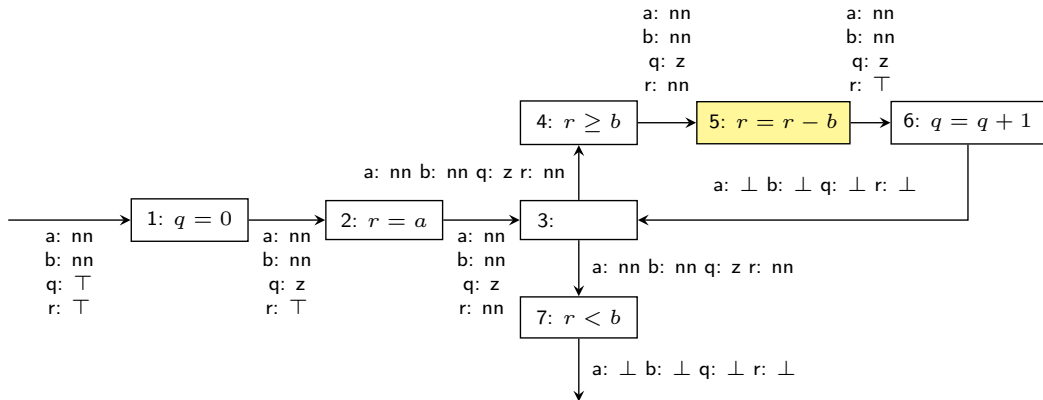
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Extended Sign Domain



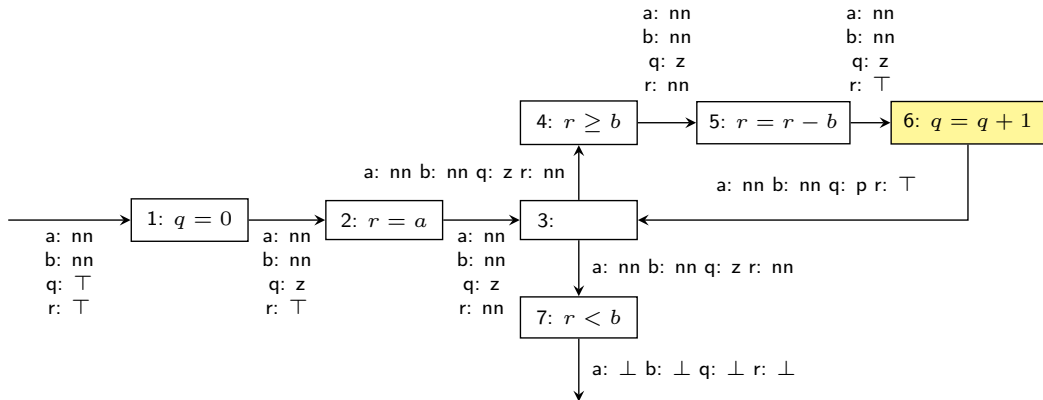
Worklist = $\{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, 5, 6, 7\}$

Extended Sign Domain



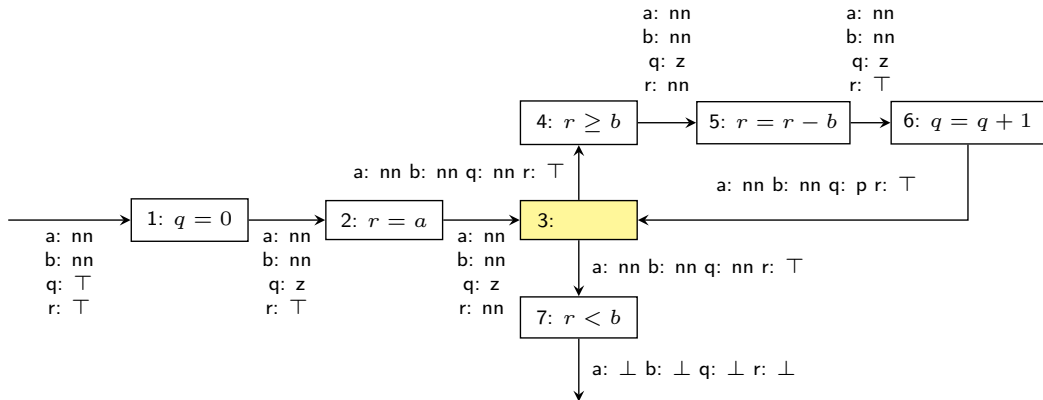
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Extended Sign Domain



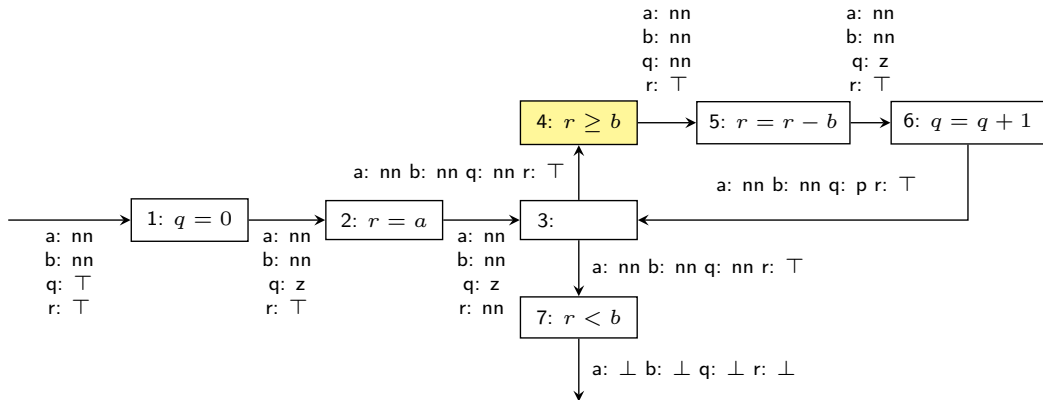
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Extended Sign Domain



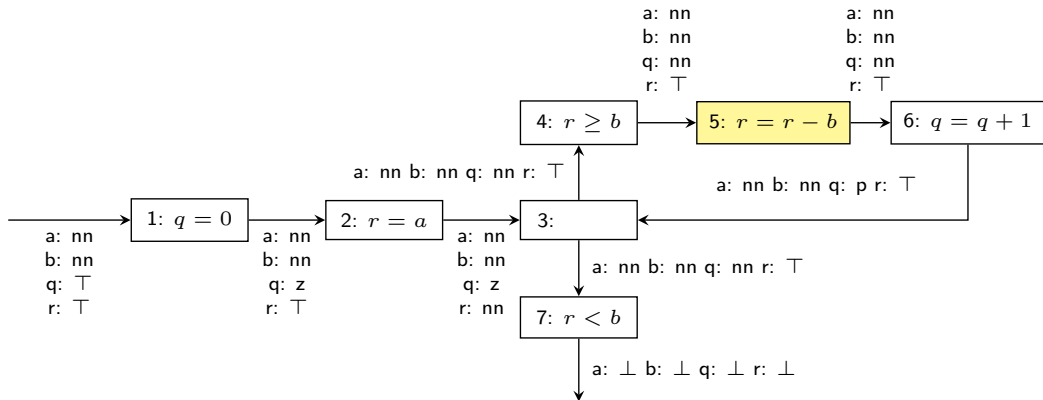
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Extended Sign Domain



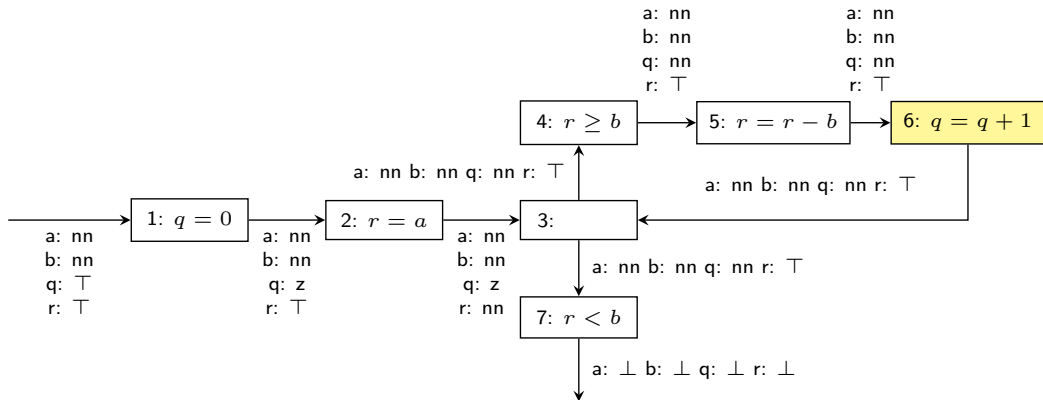
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Extended Sign Domain



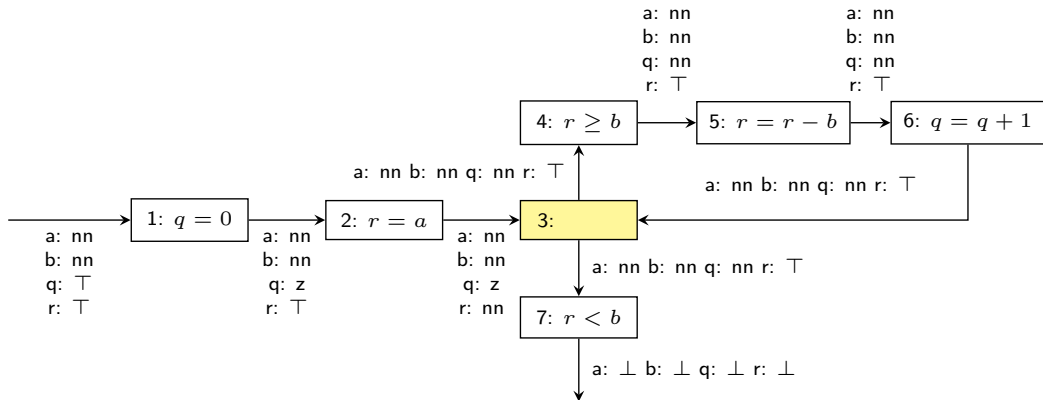
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Extended Sign Domain



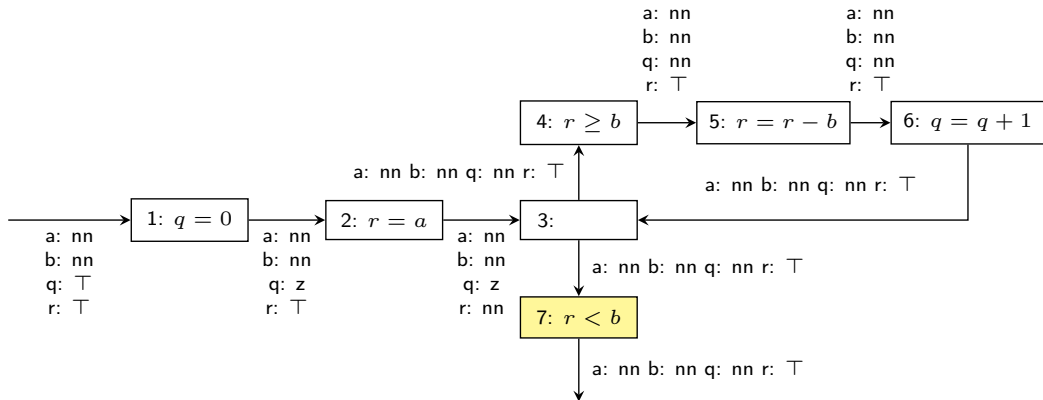
Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Extended Sign Domain



Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Extended Sign Domain

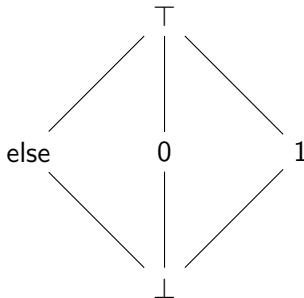


Worklist = $\{1, 2, 3, 4, 5, 6, 7\}$

Question

- Provide an example that shows a limitation of extended sign domain.

```
q = 0;  
q = q + 1;  
assert(q == 1);
```



Question : Is There Only Numeric Domain?

- There are many other variations of domains.
- For example, pointer analysis is a domain that deals with memory addresses.

```
Object f(){  
    x = new A(); //l1  
    return x;  
}
```

```
Object g(){  
    y = new B(); //l2  
    return y;  
}
```

$$x \rightarrow \{l_1\} \quad y \rightarrow \{l_2\} \quad p \rightarrow \{l_1\} \quad q \rightarrow \{l_2\}$$

```
void main() {  
    Object p = f();  
    Object q = g();  
}
```