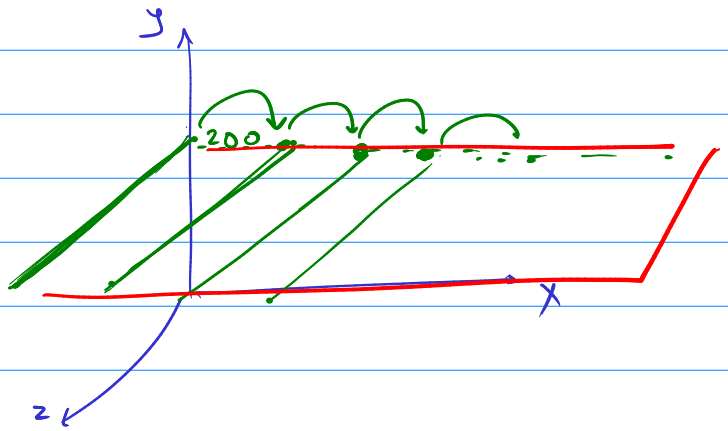
Lecture 23 Predicate Abstraction

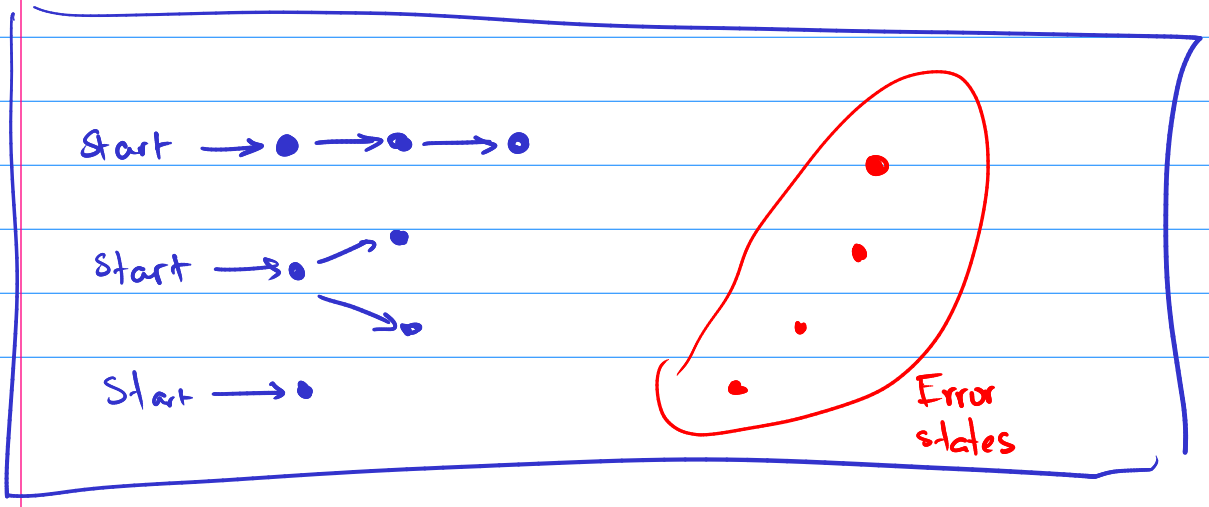
```
x := 0    y := 200    z := input()

while (x < z) {                invariant x>0   y>0
→•
    if (x+y < 0) crash!

    x := x+1;
}
```

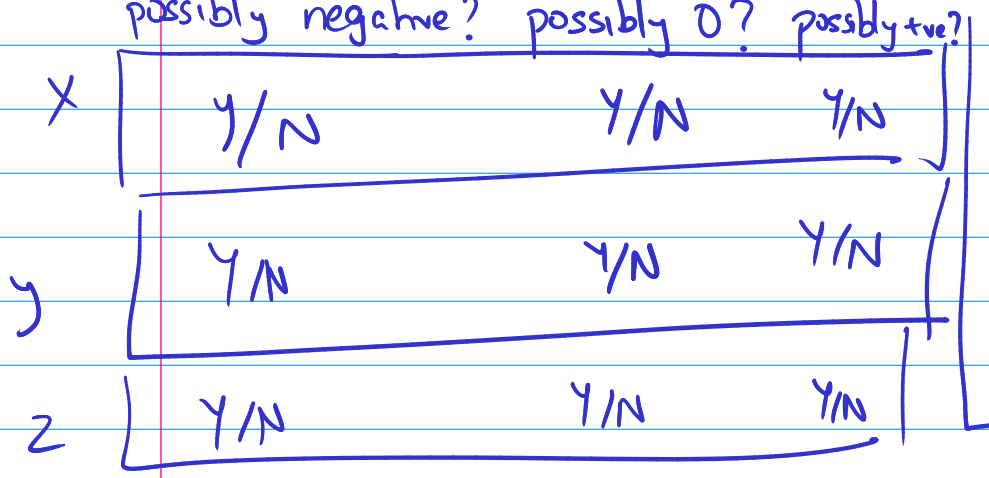


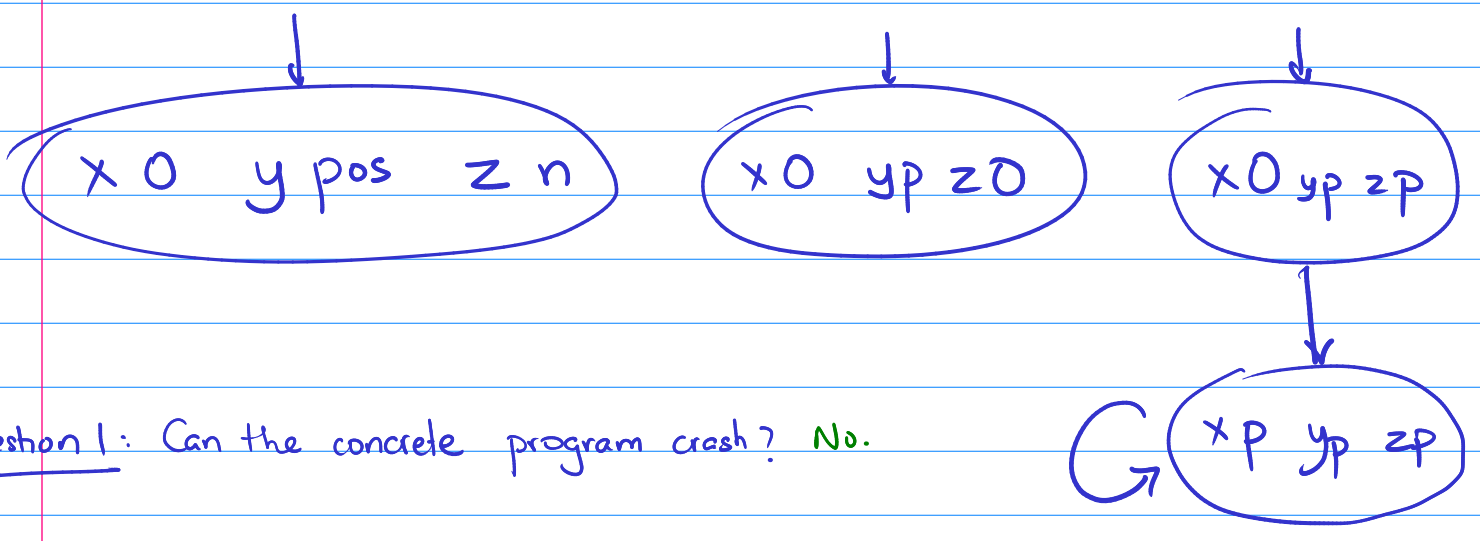
Program State Space



```
x:=0   y:=200   z:=input()
while (x < z)   invariant x>0  y>0
{
   if (x+y<0) crash!
   x := x+1;
}
```

```
x := 0
y := pos
z := neg|0|pos
while (x<z) {
   if (x+y<0) crash!
   x := if (x=neg) then
           neg|0
        else pos.
```





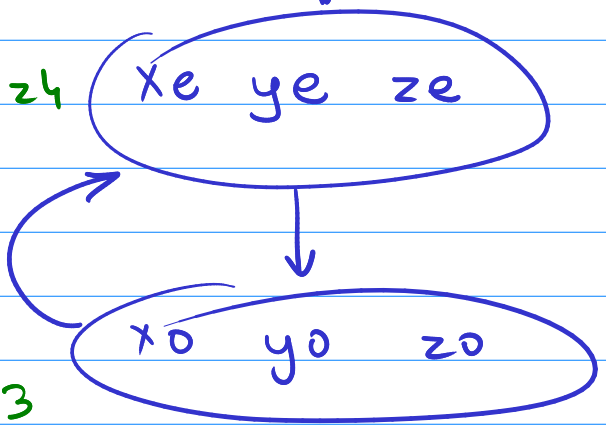
Question 1: Can the concrete program crash? No.

Question 2: Can the abstract program crash? No

Claim (Soundness): If AP can't crash then CP can't crash either.

```
x := 0 , y := 0 , z := 0
while (input() = 1)
    if ((x+y+z) % 3 != 0)
        crash!
    x := x+1
    y := y+1
    z := z+1
```

	odd?	even?
x	Y/N	Y/N
y	Y/N	Y/N
z	Y/N	Y/N



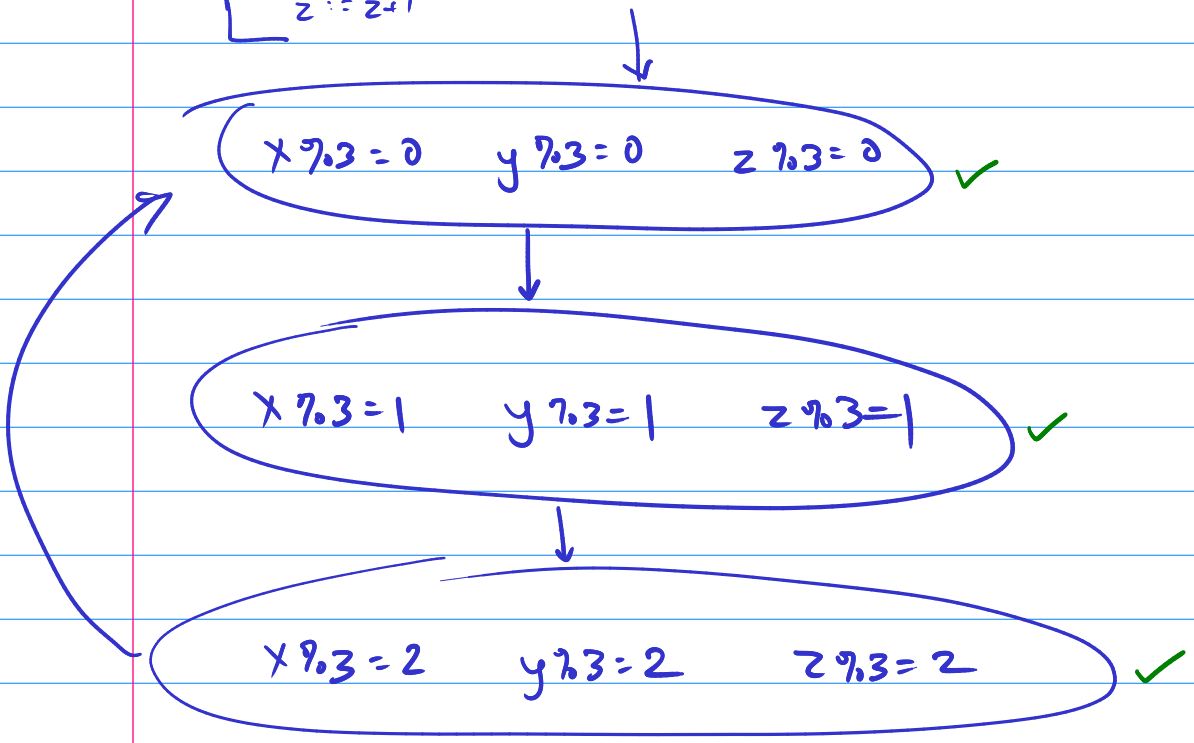
Question 1: Can the concrete program crash? No.

Question 2: Can the abstract program crash? Yes

Claim (soundness): If AP can't crash then CP can't crash either.

```
x := 0 , y := 0 , z := 0
while (input() = 1)
    if ((x+y+z) % 3 != 0)
        crash!
    x := x+1
    y := y+1
    z := z+1
```

	%3=0	%3=1	%3=2
x	Y/N	Y/N	Y/N
y	Y/N	Y/N	Y/N
z	Y/N	Y/N	Y/N



Question 2 : Can the AP crash? No.

10000 ft view of predicate abstraction

- Guess a set of predicates ← How to guess?
- Explore state space of abstract program ← Lazy abstraction
- Check if AP can crash. ← What if AP crashed? CEGAR
- Use the soundness theorem to show that CP can't crash either.

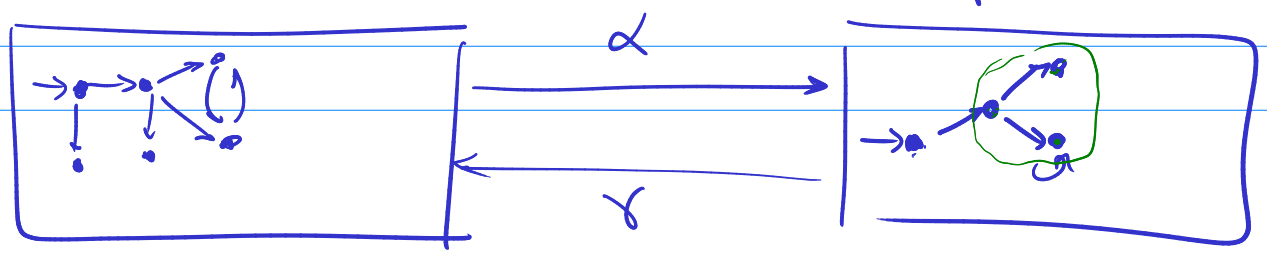
Question: When does the soundness theorem hold?

Claim (Soundness): If AP can't crash then CP can't crash either.

If CP can crash, then AP can also crash

Concrete state space

Abstract state space



Property 1: Error state in concrete must not normal state in AP

$\forall$ concrete error states e, $\alpha(e)$ is abnormal

Property 2: Normal state in AP, after concretizing, must not contain any error states.

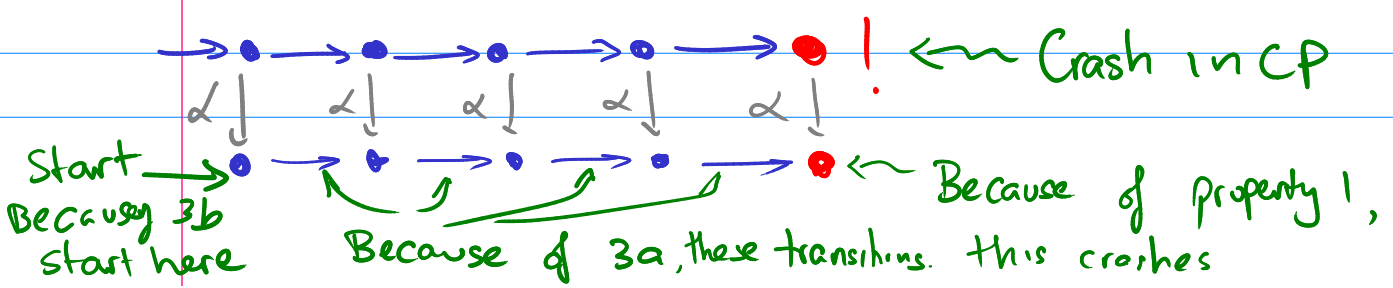
$\forall$ normal abstact states $\hat{n}$, everything in $\gamma(\hat{n})$ must be error-free.

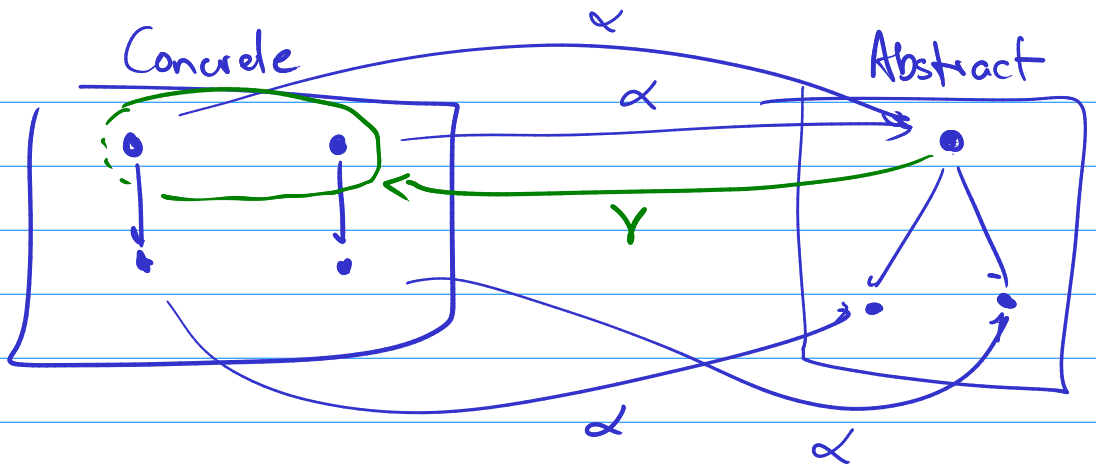
Property 3: Whenever $q \xrightarrow[\text{concrete}]{} q'$ (3a)

$$\alpha(q) \xrightarrow[\text{abstract}]{} \alpha(q')$$

Whenever $start \xrightarrow[\text{concrete}]{} q$ (3b)

$$start \xrightarrow[\text{abstract}]{} \alpha(q)$$





$\alpha : S \to \hat{S}$ $\qquad$ $\gamma : \hat{S} \to 2^S$

Problem : Existential abstractions (property 1+3a+3b)

require AP to have a minimum set of transitions

Does not immediately specify an upper bound

"Keep at least these arrows" vs.

"Keep only these arrows".