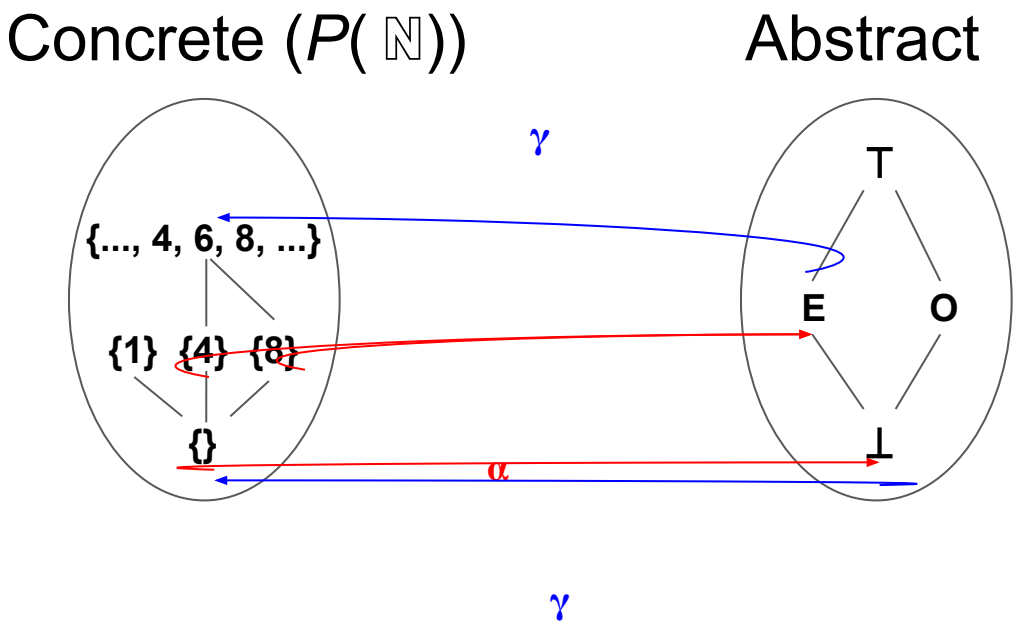
CSE 503

Software Engineering

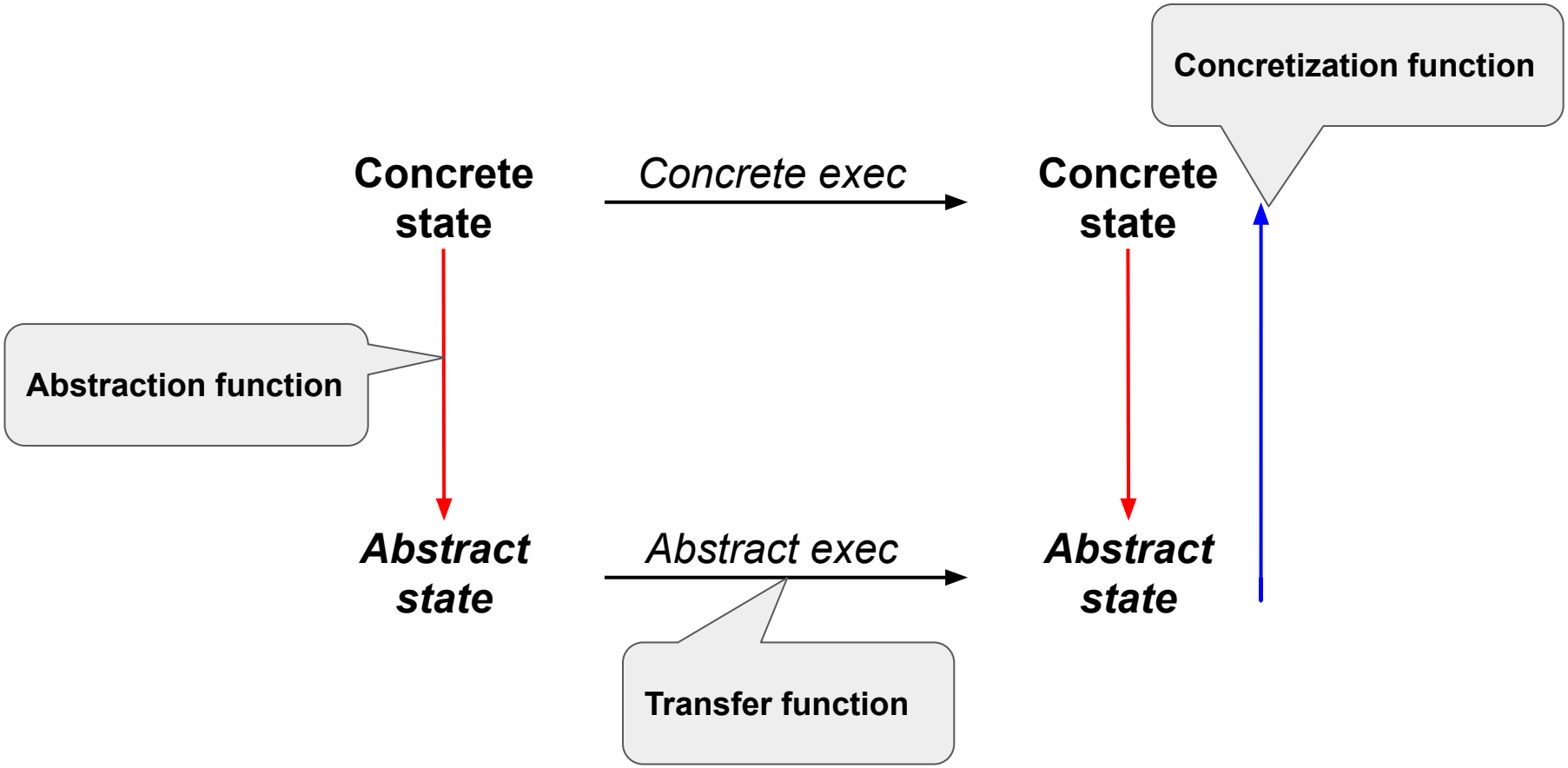
Abstract Interpretation

Recap: abstraction and concretization functions

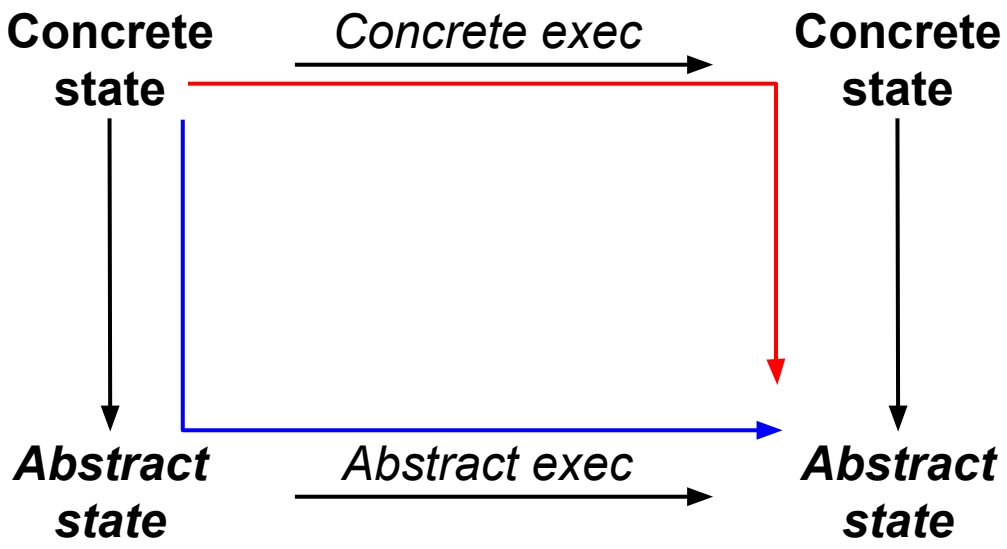


- Abstraction function α: *C* -> *A*
- Concretization function γ: A -> C

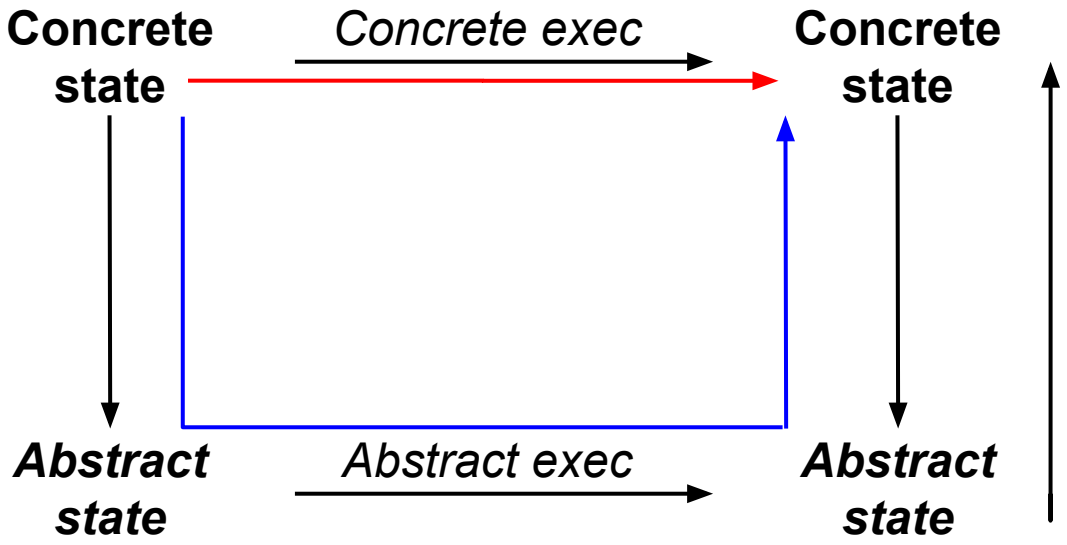
Recap: relationship between concrete & abstract



Recap: approximation



Recap: approximation



Today

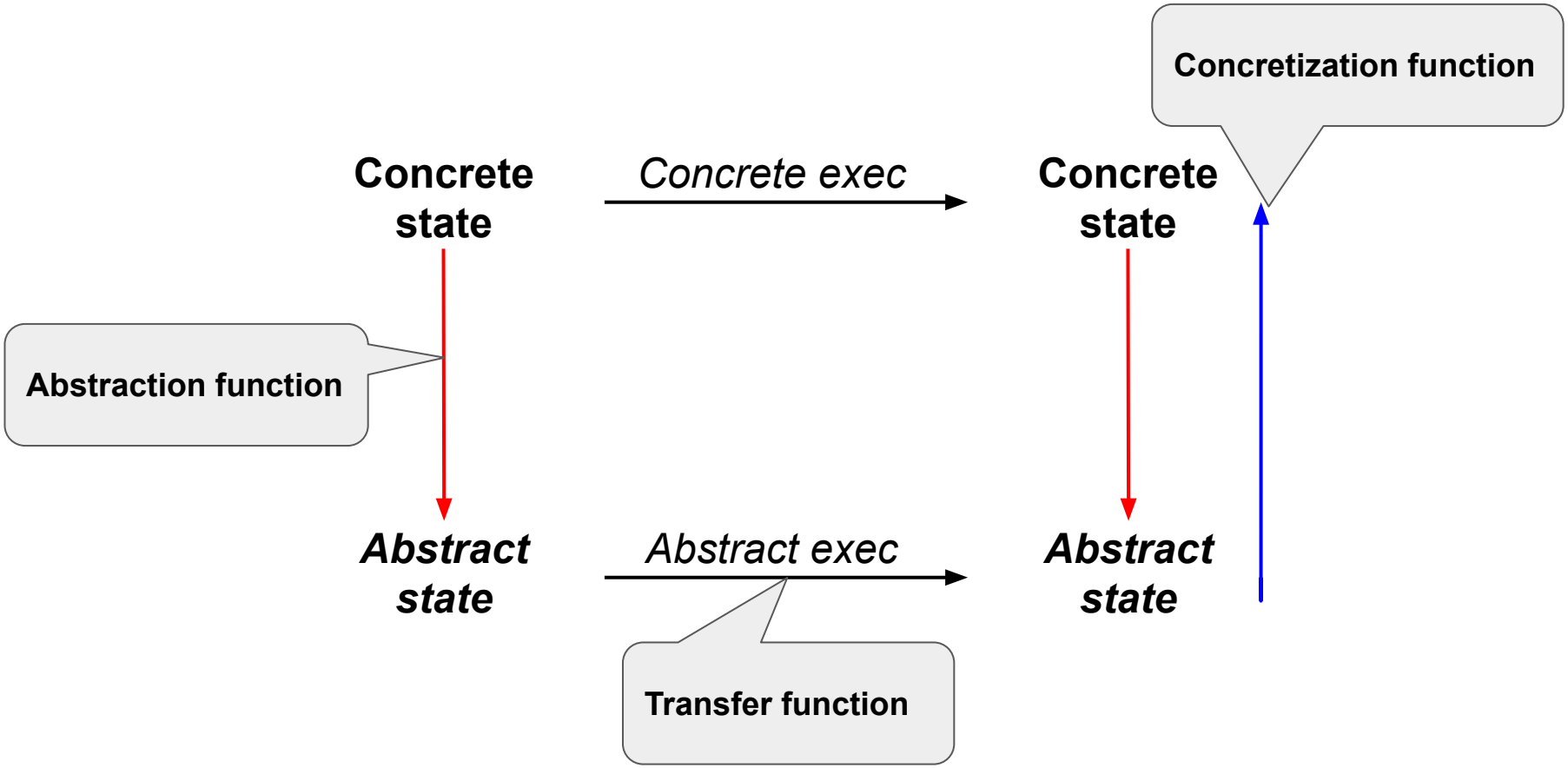
More on Abstract Interpretation

- Galois connection
- Transfer function vs. lub (vs. glb)
- Exercise: concrete examples

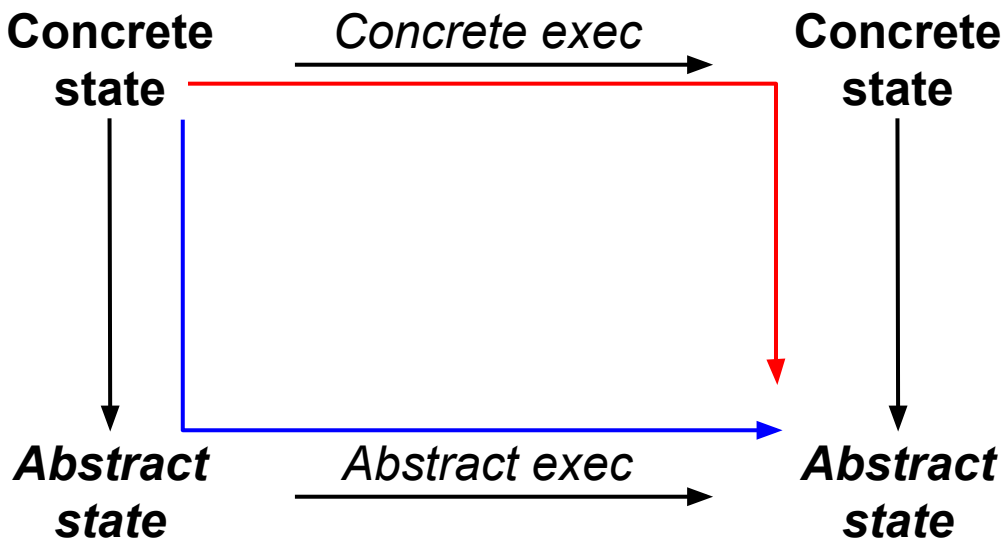
Next week

- Wrap up Abstract Interpretation
- Checker Framework tutorial
- Hands-on applications
- Move on to dynamic and hybrid analyses

Abstract interpretation: big picture



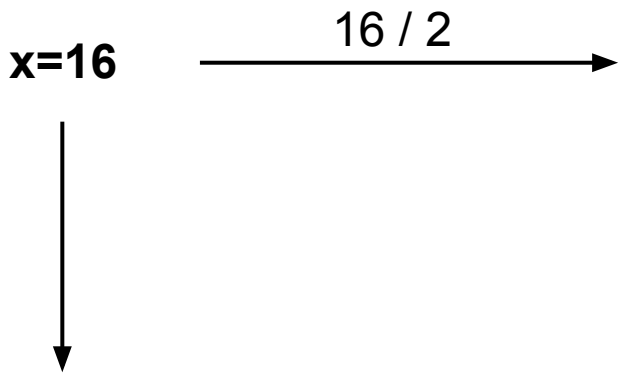
Abstract interpretation: soundness



Sound approximation and safe approximation are synonyms.

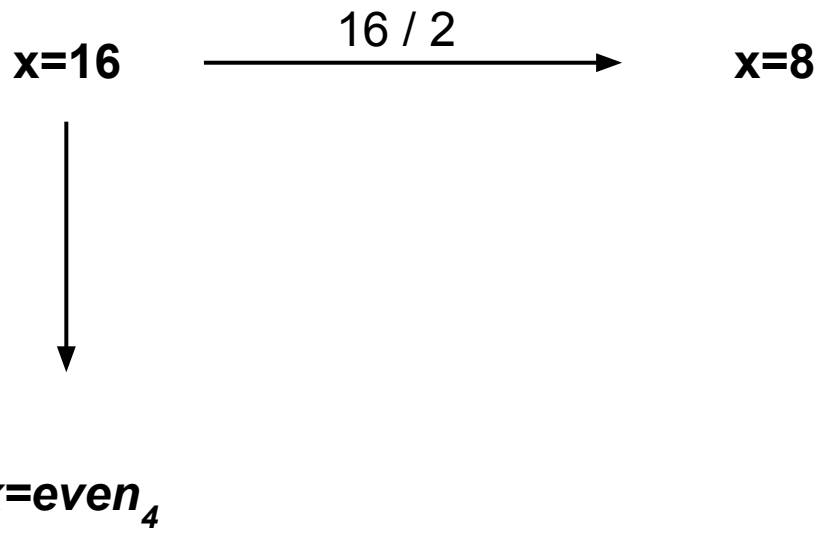
Abstract interpretation: soundness example

Abstract domain: {*odd, even*$_2$, *even*$_4$, *?*}



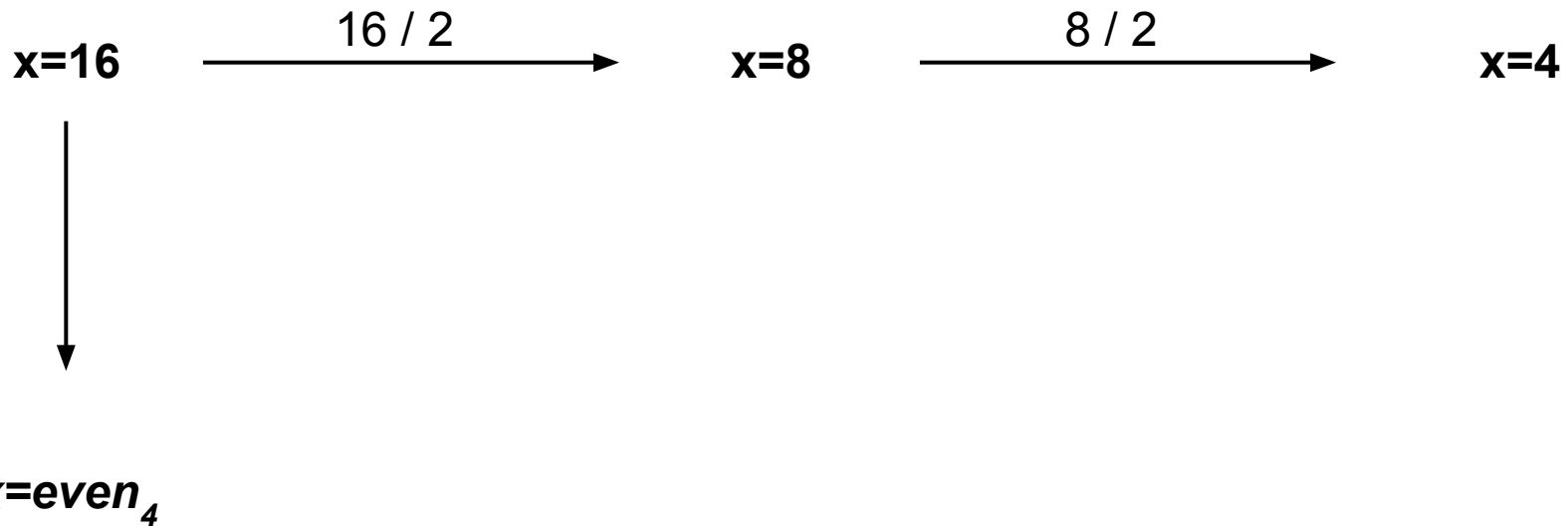
Abstract interpretation: soundness example

Abstract domain: {*odd, even*$_2$*, even*$_4$*, ?*}



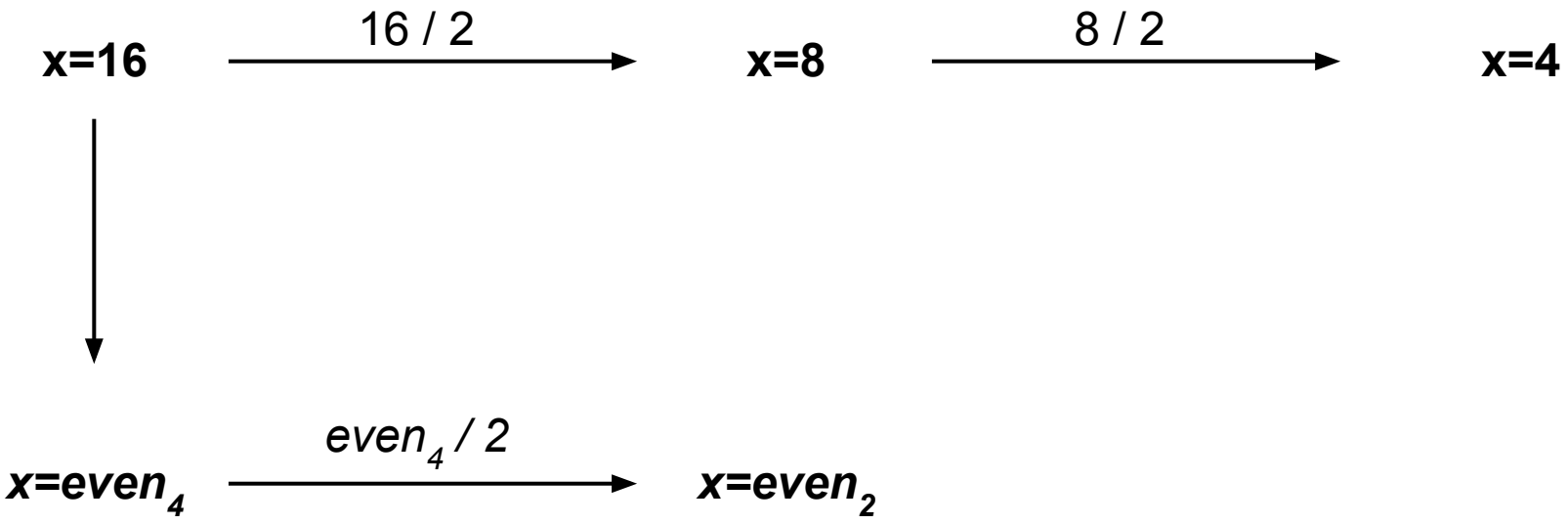
Abstract interpretation: soundness example

Abstract domain: {*odd, even*$_2$, *even*$_4$, *?*}



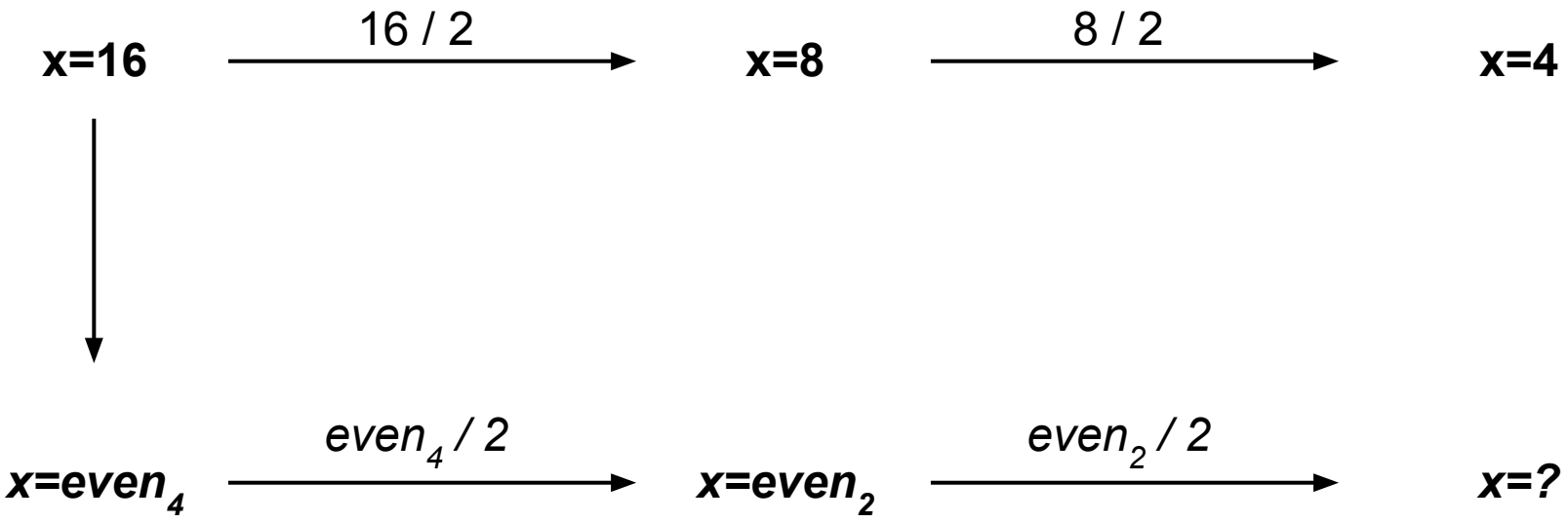
Abstract interpretation: soundness example

Abstract domain: {*odd, even$_2$, even$_4$, ?*}



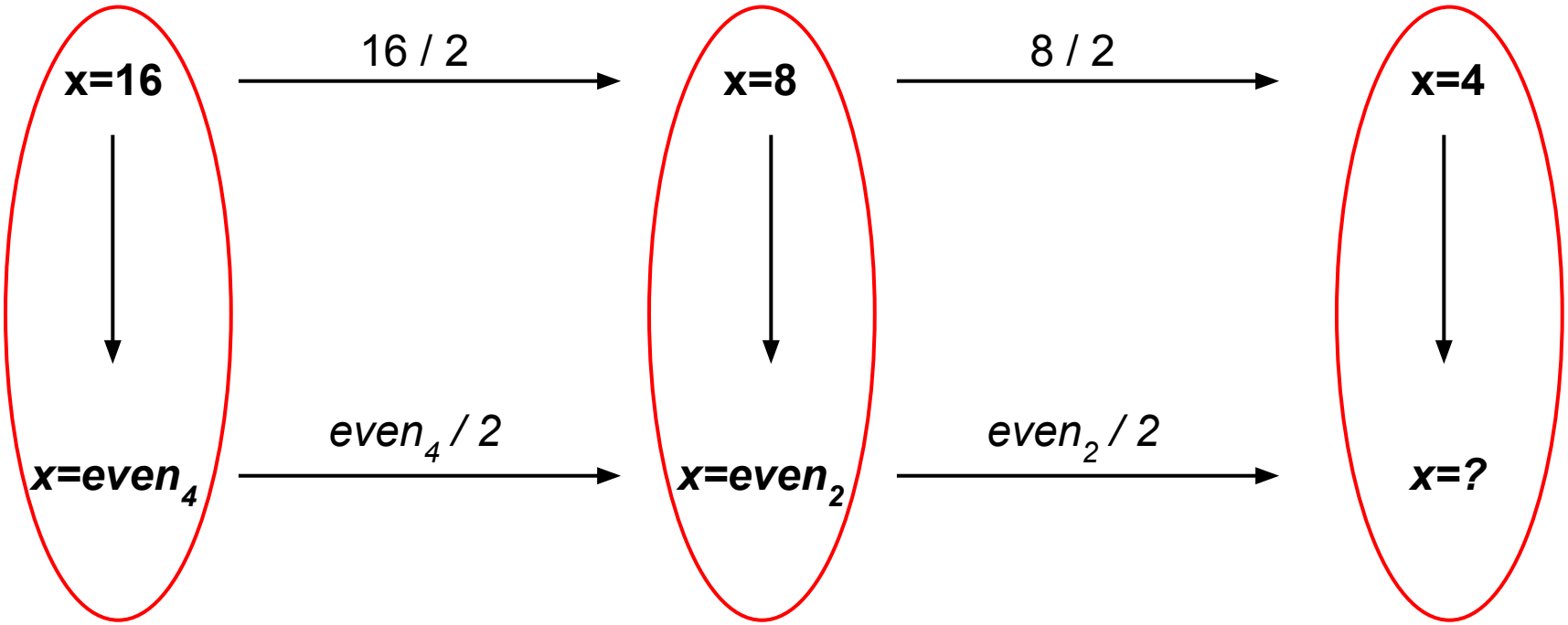
Abstract interpretation: soundness example

Abstract domain: {*odd, even$_2$, even$_4$, ?*}



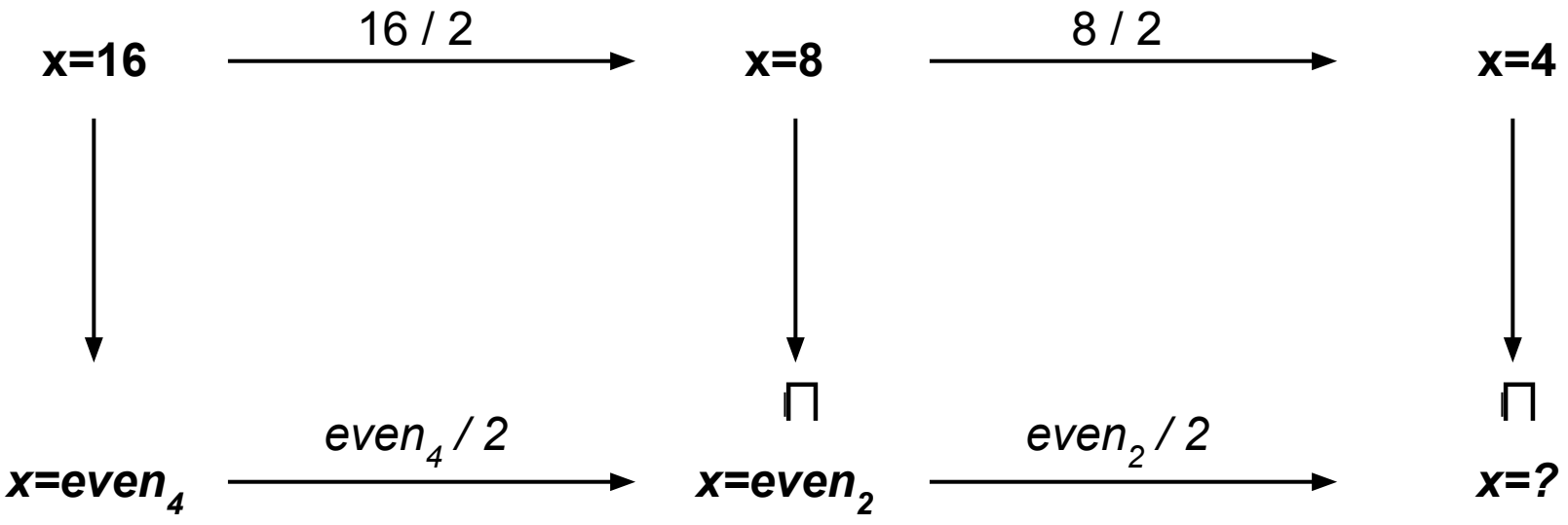
Abstract interpretation: soundness example

Abstract domain: {*odd, even*$_2$, *even*$_4$, *?*}

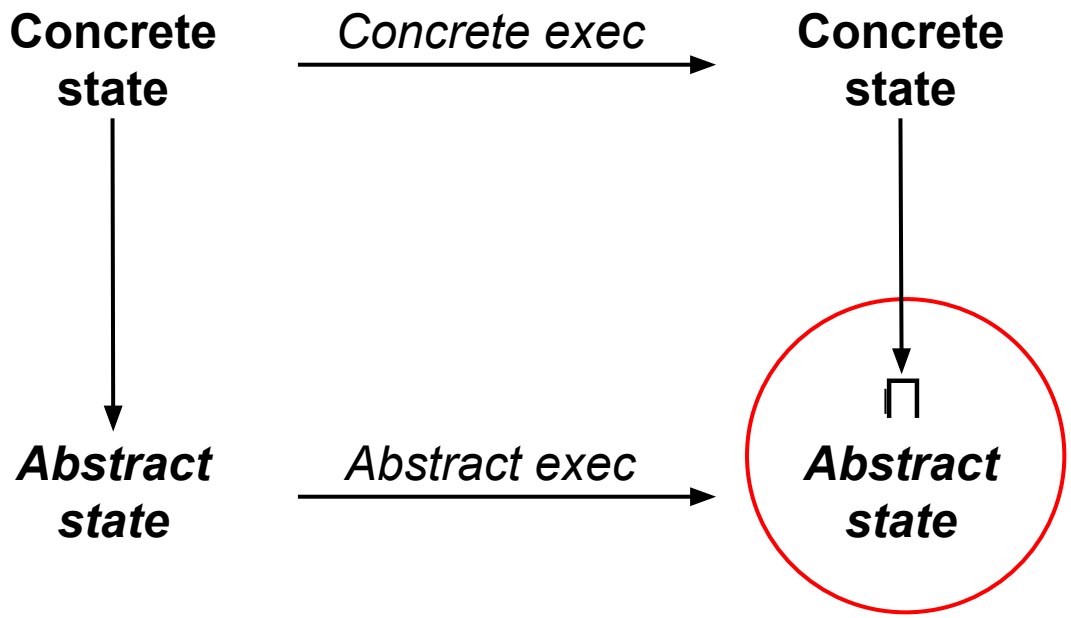


Abstract interpretation: soundness example

Abstract domain: {*odd, even$_2$, even$_4$, ?*}

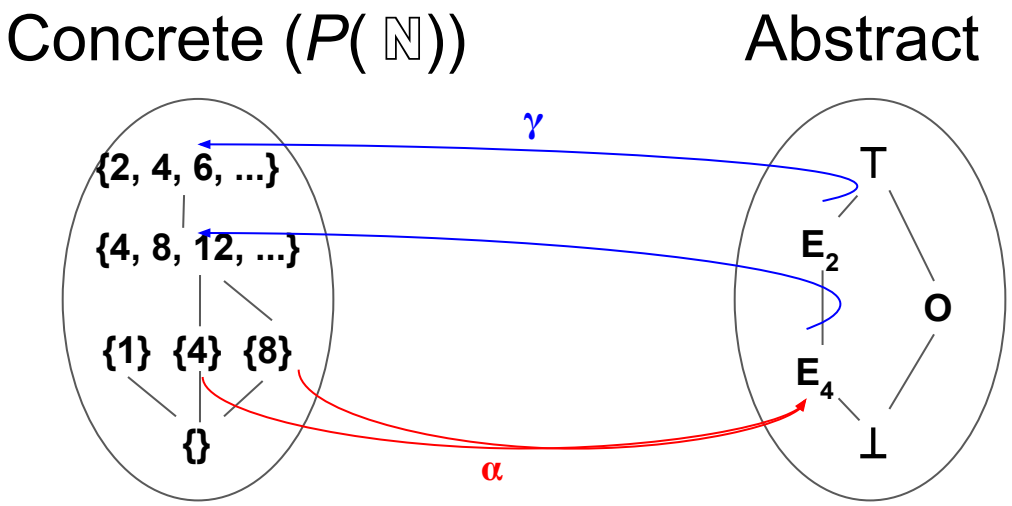


Abstract interpretation: soundness



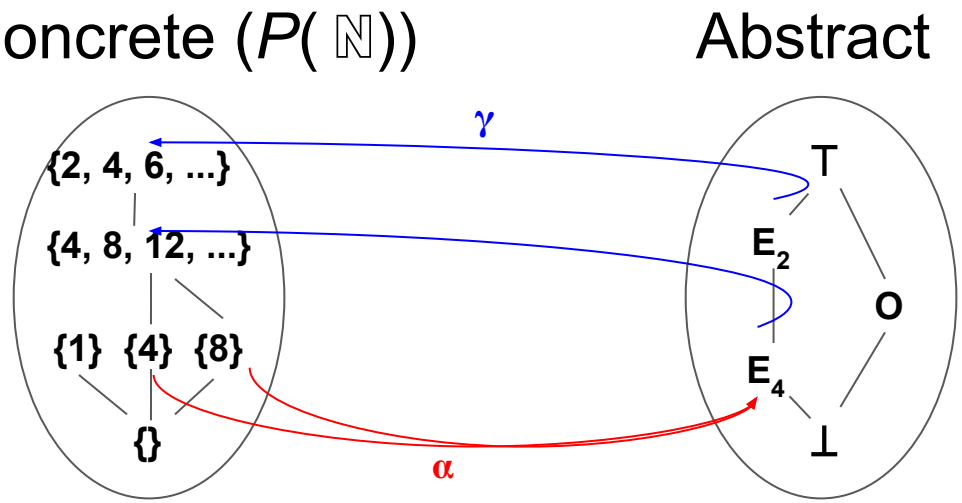
What properties must be satisfied by the abstraction, concretization, and transfer functions?

Sound approximation: properties



What properties must α and γ satisfy?

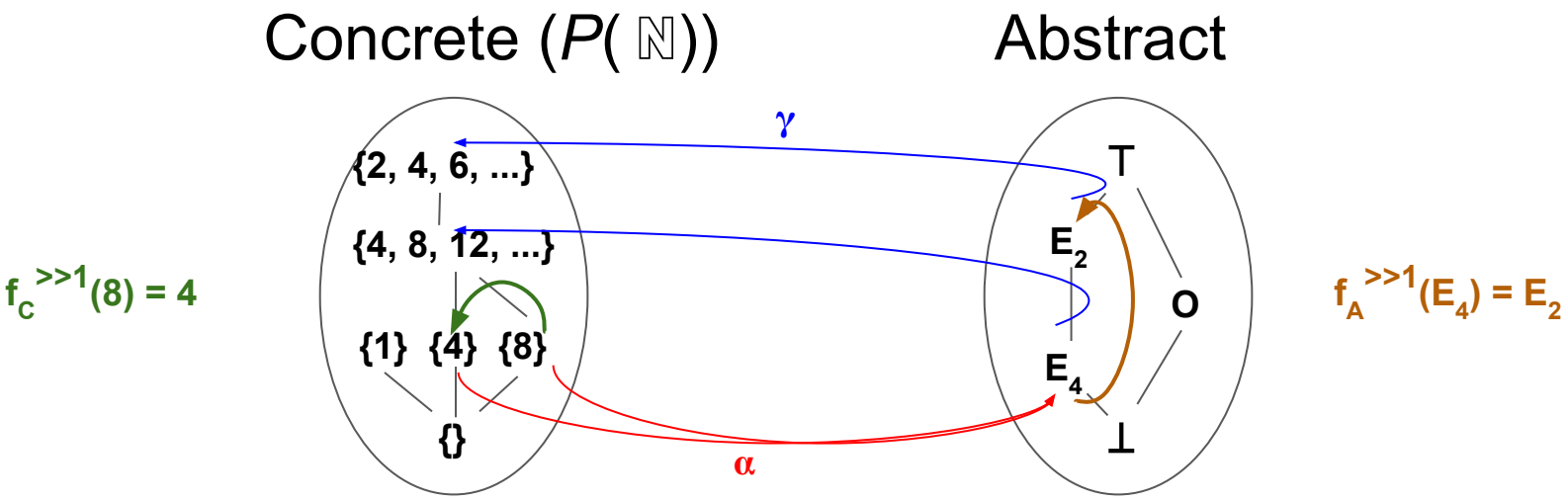
Sound approximation: galois connection



Galois connection

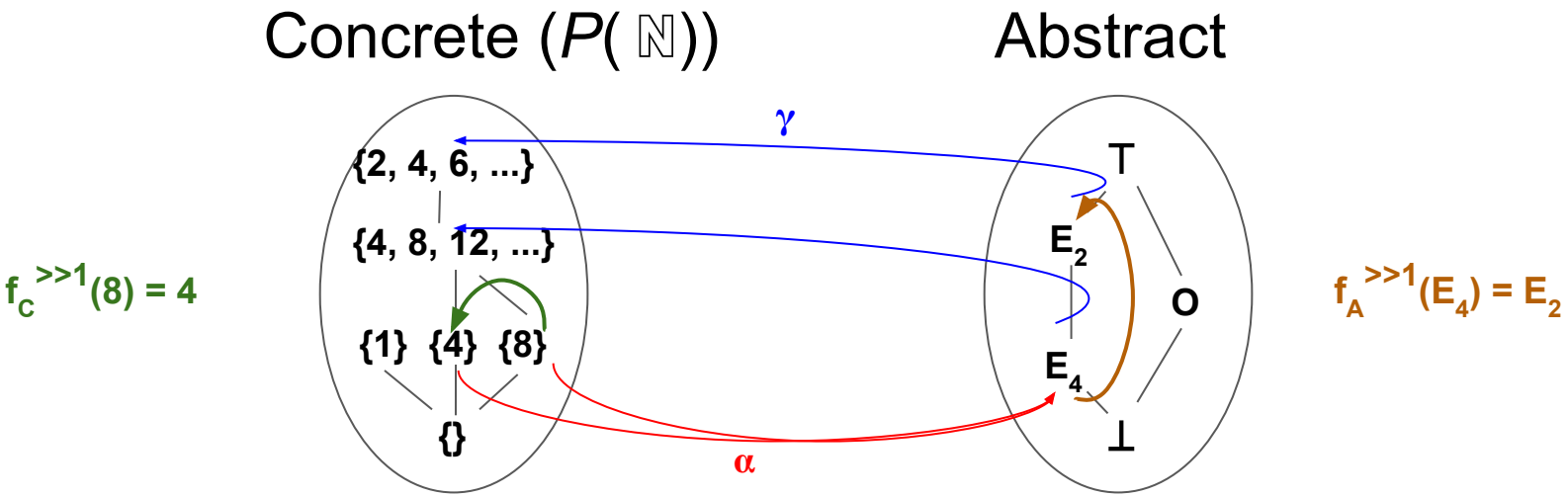
- α: *C* -> *A*
- γ: A -> C
- ∀c∈C: c ≤ γ(α(c))
- γ and α are order preserving

Sound approximation: properties



What properties must the transfer function(s) satisfy?

Sound approximation: consistency



Transfer function

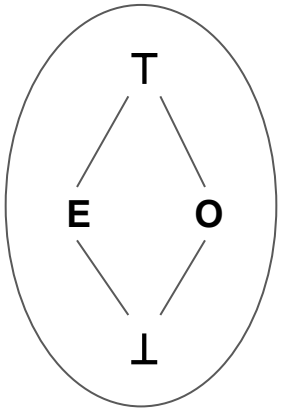
- Consistent with concrete function
  - c: concrete state; c' = $f_C(c)$
  - a: $\alpha(c)$
  - a' = $f_A(a)$
  - c'' = $\gamma(a')$
  - c' $\leq$ c''

Sound approximation: properties

Transfer function

- f_A^+: A x A -> A

+	E	O	T	...
E	E	O	T	
O	O	E	T	
T	T	T	T	
...				



Lub

- lub: A x A -> A

lub(E, O) = T

What properties must the lub function satisfy?

Sound approximation: monotonicity

Transfer function

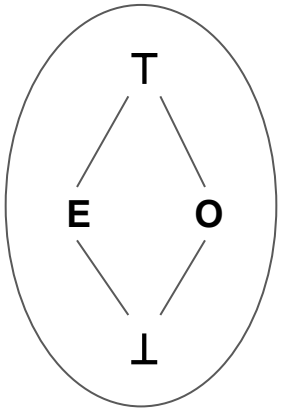
- f_A^+: A x A -> A
- might not be monotone

Lub

- lub: A x A -> A
- must be monotone

+	E	O	T	...
E	E	O	T	
O	O	E	T	
T	T	T	T	
...				

lub(E, O) = T

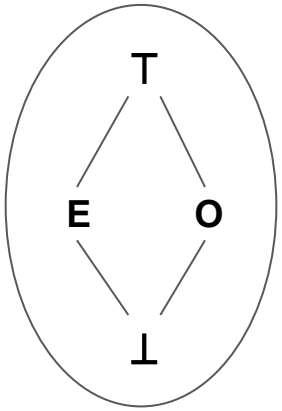


Sound approximation: example

Transfer function

- f_A^+: A x A -> A
- might not be monotone

+	E	O	T	...
E	E	O	T	
O	O	E	T	
T	T	T	T	
...				



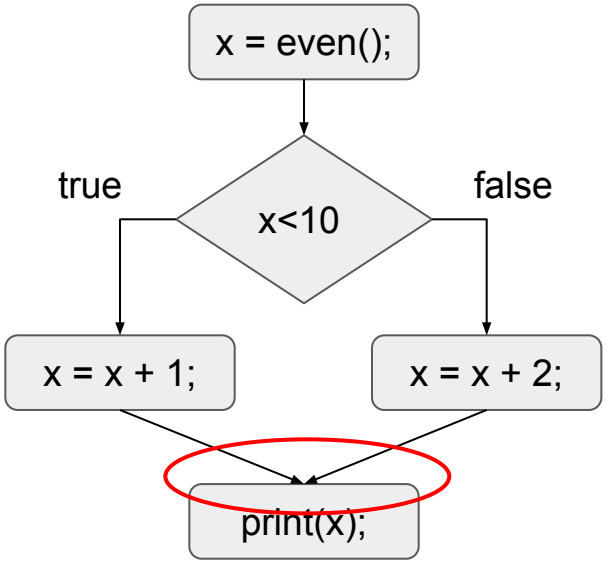
Lub

- lub: A x A -> A
- must be monotone

lub(E, O) = T

```
int x = even();

if (x < 10) {
  x = x + 1;
} else {
  x = x + 2;
}
print(x);
```



Small-group exercise

- Work through two examples:
  - Join vs. meet operation (`f(int a, int b, int c): int`)

```
if (cond) {
  x = a * b;
} else {
  x = a * c;
}
return(x);
```

Which parameters (`a, b, c`)
- will definitely be used?
- may be used?

(`cond` is independent of the parameters)

  - Termination/fix point iteration

```
int x = 2;
while (x < 10) {
  x = x + 2;
}
```

Is the value of x after the loop an even number? Use an abstract domain with {odd, 2, $even_2$, and $even_4$}

- Report to class (random call)