

Example: Sign Analysis - example program

```
def select(c : Boolean): Int = {  
  val a = 42  
  val b = 333  
  var x = 0;  
  if (c)  
    x = a + b;  
  else  
    x = a - b;  
  x  
}
```

Handwritten annotations for sign analysis:

- Initial state: $\langle a \rightarrow \perp, b \rightarrow \perp, x \rightarrow \perp \rangle$
- After `val a = 42`: $\langle +, \perp, \perp \rangle$
- After `val b = 333`: $\langle +, +, \perp \rangle$
- After `var x = 0;`: $\langle +, +, 0 \rangle$
- Inside `if (c)` block:
 - Before `x = a + b;`: $\langle +, +, + \rangle$
 - Before `x = a - b;`: $\langle +, +, \perp \rangle$
- Handwritten note: $\perp \Rightarrow T$ (indicating a sign update from \perp to T)

Example: Constant Propagation - example program

```
val z = 3
var x = 1
while(x > 0) {
  if(x == 1) {
    y = 7
  } else {
    y = z + 4
  }
  x = 3
}
```

$x \rightarrow \perp, y \rightarrow \perp, z \rightarrow \perp$

$\langle \perp, \perp, \perp \rangle$

$\langle 1, \perp, 3 \rangle \sqcup$

$\langle T, 7, 3 \rangle$

$\langle 1, 7, 3 \rangle$

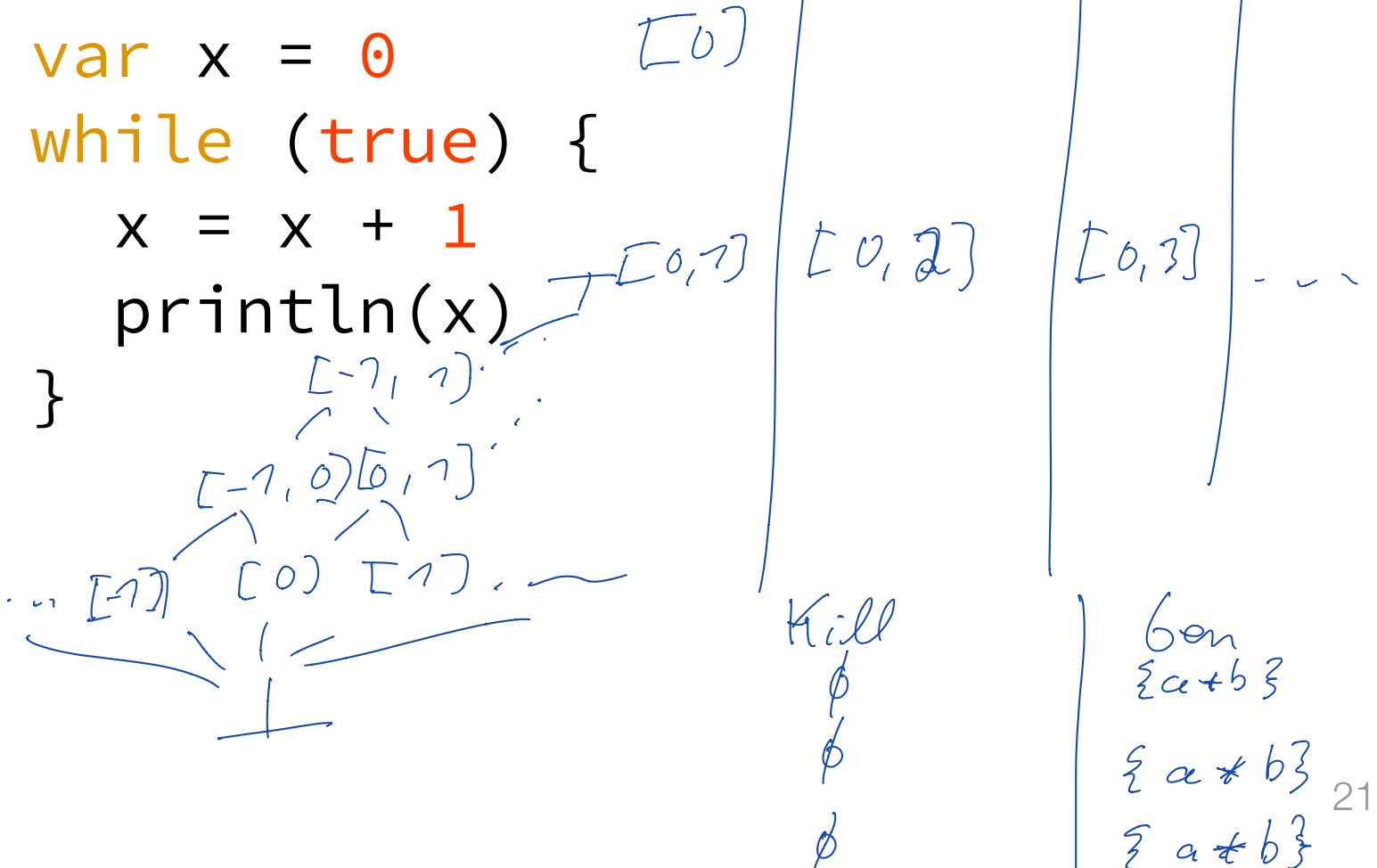
$\langle 1, 7, 3 \rangle$

$\langle 3, 7, 3 \rangle$

$\langle T, 7, 3 \rangle \vee$

Interval analysis - example

```
var x = 0
while (true) {
  x = x + 1
  println(x)
}
```



Available Expressions - Example

```
def m(initialA: Int, b: Int): Int = {
  /*pc 0*/  var a = initialA
  /*pc 1*/  var x = a + b;
  /*pc 2*/  val y = a * b;
  /*pc 3*/  while (y > a + b) {
    /*pc 4*/    a = a + 1
    /*pc 5*/    x = a + b
  }
  /*pc 6*/  a + x
}
```

$\{a+b, a*b, a+1\}$

$\{ \}$

\emptyset

$\{a+b\}$

$AE_{Entry}(pc1) = \emptyset$

$AE_{Entry}(pc3) = AE_{Exit}(pc2)$

\cap
 $AE_{Exit}(pc5)$

$AE_{Exit}(pc3) = AE_{Entry}(pc3) \cup \{a+b\}$