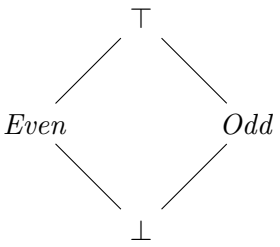
Assignment 8

Exercise 1

In the lecture you have seen two abstract domains: *Sign* and *Interval*. For this exercise we use another domain called *Parity*:



where *Even* represents all even numbers, including zero, and *Odd* represents all other numbers.

An (abstract) parity state is a mapping $PC \mapsto Vars \mapsto \{\bot, \top, odd, even\}$.

You are given the following program P:

```
   function (int x, int y) {
0:    x := y * 2
1:    while (x >= 0)
2:       if x = 1
3:          x := x + 1
          else
4:          y := x - 1
5:          x := y - 1
6:    end
7: }
```

The initial parity state of P is:

pc	x	y
0	$\bot$	$\bot$
1	$\bot$	$\bot$
2	$\bot$	$\bot$
3	$\bot$	$\bot$
4	$\bot$	$\bot$
5	$\bot$	$\bot$
6	$\bot$	$\bot$
7	$\bot$	$\bot$

Iterate over P's states, starting from the initial parity state, until you reach a fixed-point. An intuitive overview of the iterations is given in slide set 6 of the lecture. What is the fixed-point state?

Exercise 2

Are the domains Sign, Parity, and Interval pair-wise comparable?

1. For those that are comparable, which one is more precise?
2. For those that are incomparable, give an example program that can be verified only with each domain.

Exercise 3

Can you find a program and a property where with the Sign domain it takes strictly more iterations (same as the examples from the lecture) to terminate than with the Interval domain (and where we succeed in proving the property) ?

Exercise 4

Can you find a program and a property where with the Sign domain it takes strictly less iterations to terminate than with the Interval domain (and where we succeed in proving the property) ?

- For the Interval, assume the widening in the lecture and that it is applied immediately.
- For the iterations, assume that you follow the program structure, just as we did in the lecture.

Exercise 5

The Interval domain in lectures uses widening. Can you modify the structure of the Interval domain (not the elements themselves, they are still a pair [a,b]), making your new domain finite and bounded, so that you do not need widening any more and can prove the property for the following program?

```
foo (int i) {
  int x = 5;
  int y = 7;
  while (i >= 0) {
    y = y + 1;
    i = i - 1;
  }
}
assert(x + y >= 12);
```