Assignment 11 (solution)

Suppose we execute the function `main` (see below) concolically with the two symbolic variables b_0 and e_0 for `b` and `e`. For the first concrete execution we assume that `b` and `e` are both `0`.

```
int pow(int b, int e)
{
  int r = b;
  for (int i = 0; i < e; i++)
  {
    r = r * b;
  }
  return r;
}

void main(int b, int e)
{
  var r = pow(b, e);
  if (e % 2 == 0)
  {
    if (r < 0)
    {
      ERROR;
    }
  }
}
```

1. What is the path constraint that will be gathered during this first execution?

   Solution:

   ```
   PC_0 == !(0 < e_0) && (e_0 % 2 == 0) && !(b_0 < 0)
   ```

2. Negate the last conjunct in the path constraint and solve the resulting formula to generate a new input.

   Solution (other solutions are possible): `e_0 == 0, b_0 == -1`

3. What is the path constraint that will be gathered when executing function `main` with the new input?

   Solution:

   ```
   PC_1 == !(0 < e_0) && (e_0 % 2 == 0) && (b_0 < 0)
   ```

4. Repeat this process (1. run and record path constraint, 2. negate conjunct in path constraint and generate new input by solving the constraint) until you find an execution that reaches the `ERROR` statement.

   Solution: We always reach the `ERROR`.

5. Compare your concrete inputs to the test cases that are generated by the concolic test-generation tool.

   Solution:

   In our run, **pathcrawler-online.com** generated the following test case inputs (your results may differ):

   - `b == -167681, e == -175491`
   - `b == -167681, e == 0` (failing)
   - `b == 0, e == 0`

6. Now, suppose that function `pow` was uninstrumented and can only be executed in the concrete (e.g., because it was part of a native library). What is the path constraint that will be gathered during the first execution of function `main` (again with `b == 0` and `e == 0`)?

   Solution:

   ```
   PC_0 == (e_0 % 2 == 0)
   ```

7. Negate the last conjunct in the path constraint and solve the resulting formula to generate a new input.

   Solution (other solutions are possible): `e_0 == 1, b_0 == 0`

8. What is the path constraint that will be gathered when executing function `main` with the new input?

   Solution:

   ```
   PC_1 == !(e_0 % 2 == 0)
   ```

9. Is it possible to reach the `ERROR` statement by repeating this process (1. run and record path constraint, 2. negate conjunct in path constraint and generate new input by solving the constraint)?

   Solution: Only if the same constraints are solved multiple times and the solver returns a new solution/input that so happens to reach the `ERROR` statement.