Assignment 11

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`.

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

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

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

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

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

4. If you did not reach the `ERROR` statement yet, 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 does.

5. Compare your concrete inputs to the test cases that are generated by the concolic test-generation tool **pathcrawler-online.com** on the code in Listing 1.

   Go to the web page and click on "Test your code". As illustrated in Figure 1, provide the code (`main.zip`), test function (`main`), file under test (`main.c`) and click on "Test with default parameters" (the analysis will can take up to 5 minutes). Note that the result is not deterministic.

   To view the generated test cases, click on "Test Cases" and then on "Input values". Switch between test cases by selecting the number of the test case on the left hand side of the screen.

   For further instructions and documentation please refer to the About page on **pathcrawler-online.com** (click on "About").

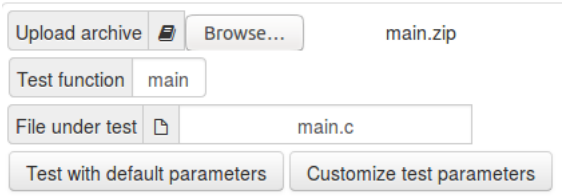

Figure 1: Screenshot from **pathcrawler-online.com**.

```c
#include <stdio.h>
#include <assert.h>

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

int main(int b, int e){
  int r=my_pow(b,e);
  if (e%2==0){
    if (r<0){
      assert(0); // ERROR
    }
  }
  return 0;
}
```

Listing 1: C++ code we want to test.

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`)?

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

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

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)?