Assignment 11

1. For the flow-sensitive pointer analysis from the lecture, write down the abstract transformer for field updates.

2. You are given the following program:

```
0:  c = newObject T;
1:  t = c;
2:  i = 0;
3:  while (i < count) {
4:    n = newObject T;
5:    c.f = n;
6:    c = n;
7:    i++;
8:  }
9:  c.f = t;
10: assert t != n;
```

   (a) Run the flow-sensitive pointer analysis from the lecture on it.

   (b) Can you prove the assertion on line 10 using the results of the analysis?

3. Write a program for which the flow-sensitive pointer analysis from the lecture infers the following abstract state at the end of the program:

   `{ a->{A0}, b->{A0,A1}, A0.f->{A0}, A1.f->{A0} }`

4. Run both the flow-sensitive and the flow-insensitive pointer analysis on the following program:

```
0: a = newObject T;
1: b = a;
2: if (a == b) {
3:   b = newObject T;
4: } else {
5: }
```

5. Suppose we execute the following function symbolically with the two symbolic variables b_0 and e_0 for `b` and `e`.

```
int pow(int b, int e)
{
  int r = b;
```

```
  for (int i = 0; i < e; i++)
  {
    r = r * b;
  }
  if (e % 2 == 0)
  {
    if (r < 0)
    {
      ERROR;
    }
  }
  return r;
}
```

(a) Enumerate all path constraints that reach the ERROR statement when unrolling the loop at most 2 times.

(b) Use the concolic test-generation tool *Pex* to find inputs that satisfy these path constraints by manually unrolling the original program. Go to `http://www.pexforfun.com/`, click on "New", and start from the following program:

```
using System;
using System.Diagnostics.Contracts;

public class Program
{
  public static int Puzzle(int b, int e)
  {
    int r = b;
    for (int i = 0; i < e; i++)
    {
      r = r * b;
    }
    if (e % 2 == 0)
    {
      if (r < 0)
      {
        Contract.Assert(false);
      }
    }
    return r;
  }
}
```