实验 3: Dafny

1 安装软件

下面的粉红色字是超链接。

1. 本次 lab 使用 dafny 形式化验证真实程序。
2. Microsoft research 提供了教程，你需要从中学习如何使用 Dafny。你需要重点阅读其中的 Introduction，Methods，Pre- and Postconditions、Quantifiers、Array 和 Predicates 节。其它部分和本次 lab 关系不大。
3. 你可以输入`make method*`来检查你的代码(如果检查 method1，就输入`make method1`)，这条指令会通过调用 dafny 的可执行文件 dafny/dafny 并传入文件名参数来检查你的程序。即调用 `./dafny/dafny xxx.dfy`（其中 xxx.dfy 是你写的 dafny 代码文件名)。

   如果代码写对了，你会看到

   ```
   Dafny program verifier finished with 1 verified, 0 errors
   Compiled assembly into method1.dll
   ```

   如果代码写错了，你会看到

```
Dafny program verifier finished with 0 verified, 1 errors
```

你要确保的是，errors 的数量为 0，否则你的代码是有问题的。

2 Problem

本次 lab 有 3 个小问题，你需要编写前置条件等。

2.1 method1

```
method method1(x: int, y: int) returns (z: int)
// Add a precondition here.
   ensures z > 0
{
   if x < 0
     { return y; }
   else
     { return x; }
}
```

编写合适的前置条件，保证返回值是正数。

2.2 method2

```
method method2(a: array<int>, v: int) returns (b: int)
// Add a precondition here.
{
   return a[v] / v;
}
```

编写合适的前置条件，使程序能验证通过，保证运行时不产生数组越界、除数是 0 等错误。

2.3 method3

```
predicate notzero(a: array<int>)
  reads a
{
// Add a predicate here.
}

method method3(a : array<int>, n : int) returns (b : int)
  requires n == a.Length && notzero(a)
  ensures b == 0;
{
  var i := 0;
  while i < n
      invariant 0 <= i <= a.Length
      invariant n == a.Length
      invariant forall k :: 0 <= k < i ==> a[k] != 0
  {
    if a[i] == 0
```

```
    { return 1; }

    i := i + 1;
  }
  return 0;
}
```

写一个合适的 predicate 让程序一定返回 0。

3 提交

3 段代码分别放在名字是 method1.dfy、method2.dfy、method3.dfy 的文件中，写完后在 lab3 目录下运行 `make handin` 生成 `lab3.zip`，然后上传到 canvas。