INTRODUCTION TO DAFNY

Lin Tzu-Chi

September 21, 2020

DAFNY

Dafny is a imperative programming language with built-in annotations to prove correctness of code.

BASIC SYNTAX

METHODS

`method`s are functions in typical imperative languages.

```
method Abs(x: int) returns (y: int)
{
  if x < 0
    { return -x; }
  else
    { return x; }
}
```

The input parameters are read only, and an implicit `return` is added automatically at the end of a method, where the current values of return parameters are returned as-is.

METHODS

There can be multiple return values.

```
method MultipleReturns(x: int, y: int)
returns (more: int, less: int)
{
  more := x + y;
  less := x - y;
  // comments.
}
```

PRE- AND POSTCONDITIONS

`ensures` annotates postconditions of a method for Dafny to check its correctness.

```
method MultipleReturns(x: int, y: int)
returns (more: int, less: int)
  ensures less < x
  ensures x < more
{
    more := x + y;
    less := x - y;
}
```

Dafny rejects this program.

PRE- AND POSTCONDITIONS

`requires` annotates preconditions. It is the programmer's job to establish them.

```
method MultipleReturns(x: int, y: int)
returns (more: int, less: int)
   requires 0 < y
   ensures less < x < more
{
   more := x + y;
   less := x - y;
}
```

Dafny verifies this program successfully.

ASSERTIONS

`assert` is a keyword to place assertions in the moddle of a method.

```
// use definition of Abs() from before.
method Testing()
{
   var v := Abs(3);
   assert 0 <= v;
}
```