Hands-on session UT: Dafny

Vincent Bloemen, Freark van der Berg, Arnd Hartmanns,
Marieke Huisman, Jaco van de Pol

Formal Methods and Tools, University of Twente

BSR Winter school
October 27, 2016

General idea

Dafny: verifying functional correctness

General idea

Dafny: verifying functional correctness

- Goal: Check whether the implementation satisfies the specification

General idea

Dafny: verifying functional correctness

- Goal: Check whether the implementation satisfies the specification
- Annotate code with the desired behaviour

General idea

Dafny: verifying functional correctness

- Goal: Check whether the implementation satisfies the specification
- Annotate code with the desired behaviour
  (both written in the Dafny language)

General idea

Dafny: verifying functional correctness

- Goal: Check whether the implementation satisfies the specification
- Annotate code with the desired behaviour
  (both written in the Dafny language)
- Dafny then checks if the specification holds

General idea

Dafny: verifying functional correctness

- Goal: Check whether the implementation satisfies the specification
- Annotate code with the desired behaviour
  (both written in the Dafny language)
- Dafny then checks if the specification holds
  - ⋆ If not, it shows why it can't verify the code

Specification

Pre-condition:

Post-condition:

Loop invariant:

Specification

Pre-condition: Statements that hold **before** executing

```
requires ...
```

Post-condition:

Loop invariant:

Specification

Pre-condition: Statements that hold **before** executing

```
requires ...
```

Post-condition: Statements that hold **after** executing

```
ensures ...
```

Loop invariant:

Specification

Pre-condition: Statements that hold **before** executing

```
requires ...
```

Post-condition: Statements that hold **after** executing

```
ensures ...
```

(Hoare logic: `{P} S {Q}`)

Loop invariant:

Specification

Pre-condition: Statements that hold **before** executing

```
requires ...
```

Post-condition: Statements that hold **after** executing

```
ensures ...
```

(Hoare logic: `{P} S {Q}`)

Loop invariant: Property that is true **before and after** each iteration of a loop

```
invariant ...
```

Using Dafny

Can be used online at http://rise4fun.com/Dafny

Some last things

Verification is modular: you only know what has been specified

Some last things

Verification is modular: you only know what has been specified

- We can assert the post-condition after the method returns

Some last things

Verification is modular: you only know what has been specified

- We can assert the post-condition after the method returns

Useful steps for verification:

Some last things

Verification is modular: you only know what has been specified

- We can assert the post-condition after the method returns

Useful steps for verification:

1. Prove termination

Some last things

Verification is modular: you only know what has been specified

- We can assert the post-condition after the method returns

Useful steps for verification:

1. Prove termination
2. Specify a post-condition

Some last things

Verification is modular: you only know what has been specified

- We can assert the post-condition after the method returns

Useful steps for verification:

1. Prove termination
2. Specify a post-condition
3. Verify that the post-condition holds

Some last things

Verification is modular: you only know what has been specified

- We can assert the post-condition after the method returns

Useful steps for verification:

1. Prove termination
2. Specify a post-condition
3. Verify that the post-condition holds

Good luck!