Formal Methods in Computer Science

UE 185.A93 WS 2019

Block 3: Deductive Verification of Programs

Laura Kovács

AB Formal Methods in Systems Engineering
TU Wien

for(syte) TU WIEN Informatics

Setting of the Exercise Sheet of Block 3

- We use the Dafny tool for automating deductive verification

Setting of the Exercise Sheet of Block 3

- We use the Dafny tool for automating deductive verification
- **Schedule** of the exercises of Block 3 in WS 2018/19:
  - November 26: presentation of the exercise sheet (in class);
  - December 16: submission deadline (upload on TUWEL);
  - January 14: presentation of sample solutions (in class).

Setting of the Exercise Sheet of Block 3

Submission Instructions:

Upload a single zip archive in TUWEL containing:

1. One pdf file containing your written solutions using the provided LaTeX-template (on TUWEL).

   This file should contain your solution to the exercises, together with explanations to the Dafny files solving your exercises.

2. The Dafny files you had to write and test for solving exercises.

Setting of the Exercise Sheet of Block 3

Submission Guidelines:

Name your zipped solutions as: `surname - studentID.zip`,

where `surname` is your surname and `studentID` is your student ID.

Files in the zip should be named accordingly:

- `surname-studentID.pdf` – for the pdf containing your solutions;
- `surname-studentID-exnr.dfy` – for the Dafny files containing your solutions, where `exnr` is the number of the exercise sub-problem your Dafny solution refers to.

Setting of the Exercise Sheet of Block 3

Notes:

- Keep in mind the general information and guidelines presented during the kick-off meeting (slides on TUWEL);
- Discuss solutions, but do not copy!

Questions?

Introduction to Dafny

- Programming language designed for reasoning about programs
  – developed by Rustan Leino, now at Amazon

Introduction to Dafny

- Programming language designed for reasoning about programs
  – developed by Rustan Leino, now at Amazon
- Programming language features similar to:
  - imperative programming languages
    method, if, while, :=, . . .
  - functional programming languages
    function, datatype, codatatype . . .
  - proof authoring
    lemma, inductive predicate, . . . ,

Introduction to Dafny

- Programming language designed for reasoning about programs
  – developed by Rustan Leino, now at Amazon
- Programming language features similar to:
  - imperative programming languages **– we focus on** IMP
    method, if, while, :=, . . .
  - functional programming languages
    function, datatype, codatatype . . .
  - proof authoring
    lemma, inductive predicate, . . . ,

Introduction to Dafny

- Programming language designed for reasoning about programs
  – developed by Rustan Leino, now at Amazon
- Programming language features similar to:
  - imperative programming languages **– we focus on** IMP
    method, if, while, :=, . . .
  - functional programming languages
    function, datatype, codatatype . . .
  - proof authoring
    lemma, inductive predicate, . . . ,
- Program verifier: proves program is correct wrt its specification

Introduction to Dafny

- Programming language designed for reasoning about programs
  – developed by Rustan Leino, now at Amazon
- Programming language features similar to:
  - imperative programming languages **– we focus on** IMP
    method, if, while, :=, . . .
  - functional programming languages
    function, datatype, codatatype . . .
  - proof authoring
    lemma, inductive predicate, . . . ,
- Program verifier: proves program is correct wrt its specification
- Integrated development environment (IDE)

Using Dafny

- easiest: in web browser at `https://rise4fun.com/Dafny/`
- with more details: `http://github.com/Microsoft/Dafny`
  - Dafny IDE in Visual Studio
  - Dafny mode in Emacs

Getting Familiar with Dafny

Recommended reading material:

- Dafny tutorial:
  https://rise4fun.com/Dafny/tutorial/Guide
- J. Koening and K.R.M. Leino. “Getting Started with Dafny: A Guide”
  https://www.microsoft.com/en-us/research/publication/getting-started-dafny-guide/

Brief Introduction to Dafny

- Dafny has methods, annotated with pre- and post-conditions, invariants, variants, ...
  - Annotations written in first-order logic
- Program correctness proved using weakest (liberal) preconditions;
- Verification conditions (VCs) are proved using the Z3 SMT solver.

Dafny's VC can be "inspected" once Dafny is installed, by running:

```
dafny example.dfy -proverlog example.bpl
```

Brief Introduction to Dafny

Program Variables in Dafny

- Declared with keyword `var`. Types declared by `:`

  Integer type is `int`

- Assignment statement written with `:=`. Equality check written `==`.

- Several variables can be declared at once.

  Example: `var i: int, j: int;`

- Parallel assignments possible.

  Example: `var i, j := 1, 0;`

Brief Introduction to Dafny

Methods in Dafny

- Explicit names for return values.
- Can refer to input arguments and return values in specifications.

Example:

```
method Division(x:int, y:int) returns (quo:int, rem:int)
```

Brief Introduction to Dafny

Dafny Keywords

- Pre-conditions specified by `requires`
- Post-conditions specified by `ensures`
- Loop invariants specified by `invariant`
- Loop variants specified by `decreases`

Note:

Any program accepted by Dafny is guaranteed to be totally correct!

Dafny requires that methods terminate!

Dafny Examples

Demo using:

- `Division` from lecture slides;

and using examples from `https://rise4fun.com/Dafny`:

- `Fibonacci`
- `CountToN` (and its version with `i!=n` instead of `i<n`)
- `CountToAndReturnN`