

# Automated Testing and Verification

## Project 1 – A Small Program Verifier

**Deadline:** 22.11.2012

You will find all resources needed for this project in <https://www.st.cs.uni-saarland.de/edu/automatedtestingverification12/projects/project1.tar.gz>. This archive file contains:

- The project description (this document)
- The source-code for the Pest parser/AST
- A set of 10 Pest programs (named TEST1.PEST ... TEST10.PEST)

### (15 points) Exercise 1

Install the CVC3 automatic theorem prover. It can be downloaded from <http://www.cs.nyu.edu/acsys/cvc3>.

Please prove the following properties over integer numbers using CVC3.

- a) (5 points) The binary relation  $\leq$  is transitive (i.e. if  $a \leq b$  and  $b \leq c$  then  $a \leq c$ ).
- b) (5 points) For each pair of integers  $a, b$  only one of the following formulas hold:
  - $a < b$ ,
  - $a = b$ ,
  - $a > b$ .
- c) (5 points) Each integer  $b$  such that  $a \leq b$  and  $b \leq a+1$  satisfies only one of the following properties:
  - $b = a$
  - $b = a + 1$ .

### (65 points) Exercise 2

The goal of this exercise is to provide a translator from PEST into CVC3. The translator takes a PEST program as input and creates a file with CVC3 commands as output. Successfully executing all queries should imply the correctness of the program w.r.t. its specification. Your translator should be able to translate the 10 provided programs into valid CVC3 commands.

In order to alleviate the task, a component for parsing PEST programs is available. We also recommend a *backward* generation of the *verification conditions*. Please remember the correctness of loop invariants should also be verified.

**Important:** supporting procedure invocation is optional and will not be considered for grading the project.

**Grading:** 6 points for passing each Pest program TEST1 to TEST9, and 11 points for passing TEST10.

### (20 points) Exercise 3

Execute the implemented PEST to CVC3 translator on the provided PEST programs. Fix the specifications until CVC3 reports no potential problems. The intended behavior of the program should be respected.

**Grading:** 2 points for fixing each Pest program.

## Handout format

This project should be delivered before or during the handout date written at the very beginning of this document.

An email should be sent to the staff email (`atv12@st.cs.uni-saarland.de`) with the following material:

1. A file `src.zip` with the project source code. Code must be fully commented.
2. A file `readme.txt` with instructions on how to execute the delivered project.
3. A file `report.pdf` with a description of the resolution of all exercises, including a brief discussion on the most important design decision taken during the project.
4. A file `id.txt` containing the full names and matriculation numbers of all group members.

The e-mail subject should be:

[ATV-project1] name1 (matriculation1) / name2 (matriculation2)

where name1 and name2 should be lexically ordered. No printed material will be accepted.