Dr. Matthias Heizmann
Dominik Klumpp

Hand in until 10:00 on June 07, 2021
Discussion: June 07, 2021

Tutorial for Program Verification
Exercise Sheet 11

In this exercise sheet we work with the Hoare proof system, its soundness proof, and the Ultimate Referee tool. We then continue with some exercises on the theory of arrays.

Submit your solution by uploading it as PDF in ILIAS. Upload SMT scripts as text files.

Exercise 1: Hoare Logic Proof — 3 Bonus Points

Consider the following Boostan program $P = (V, \mu, \mathcal{T})$ with $V = \{e, x, y, z\}$, $\mu(e) = \mu(x) = \mu(y) = \mu(z) = \mathbb{Z}$, and $\mathcal{T}$ a derivation tree for the program code shown below.

```
e := 1;
z := 0;

while (z < y) {
     e := e * x;
     z := z + 1;
}
```

Give a Hoare logic proof showing that $\{y \geq 0\}\, \mathcal{T}\, \{e = x^y\}$ is a valid Hoare triple.

Exercise 2: Soundness of the Weakening Postcondition Rule — 1 Point

Prove that the weakening postcondition rule of the Hoare proof system displayed below is sound.

$$(weakpos)\frac{\{\varphi\}st\{\psi\}}{\{\varphi\}st\{\psi'\}} \text{ if } \psi \models \psi'$$

More precisely, prove the following lemma from the lecture:

> If the Hoare triple $\{\varphi\}st\{\psi\}$ is valid and the side condition $\psi \models \psi'$ holds, then the Hoare triple $\{\varphi\}st\{\psi'\}$ is valid.

Exercise 3: Soundness of the Conditional Rule — 2 Points

Prove that the conditional rule of the Hoare proof system displayed below is sound.

$$(condi)\frac{\{\varphi \land expr\}\ st_1\ \{\psi\} \quad \{\varphi \land \neg expr\}\ st_2\ \{\psi\}}{\{\varphi\}\ \texttt{if}(expr)\{st1\}\texttt{else}\{st2\}\ \{\psi\}}$$

More precisely, prove the following lemma from the lecture:

> If the Hoare triple $\{\varphi \land expr\}\ st_1\ \{\psi\}$ is valid and the Hoare triple $\{\varphi \land \neg expr\}\ st_2\ \{\psi\}$ is valid, then the Hoare triple $\{\varphi\}\ \texttt{if}(expr)\{st1\}\texttt{else}\{st2\}\ \{\psi\}$ is valid.

Exercise 4: Square 2 Points

Find an inductive loop invariant for the while loop of the following program that is strong enough to prove that the program satisfies the given precondition-postcondition pair, i.e., the formulas after `requires` and `ensures`, respectively. Use Ultimate Referee[1] to check your solution.

Alternatively you may also use your own solution to exercise 4 on exercise sheet 9.

```
procedure square(n: int) returns (res: int)
requires n >= 0;
ensures res == n*n;
{
  var i, odd : int;
  i := 0;
  odd := 1;
  res := 0;
  while (i < n) {
    res := res + odd;
    odd := odd + 2;
    i := i + 1;
  }
}
```

Exercise 5: Satisfiability in the Theory of Arrays 2 Points

Determine which of the following FOL formulas is satisfiable in the theory of arrays. If a formula is satisfiable, give a satisfying assignment. You may assume that the arrays have integer indices and values.

(a) $\mathit{select}(a, i) = i \land \mathit{store}(a, i, k) = a \land i \neq k$

(b) $a = \mathit{store}(b, k, v) \land \mathit{select}(a, i) \neq \mathit{select}(b, i) \land \mathit{select}(a, j) \neq \mathit{select}(b, j)$

(c) $b = \mathit{store}(a, k, v) \land \forall i.\ i \neq j \rightarrow \mathit{select}(a, i) = \mathit{select}(b, i)$

You may use an SMT solver to solve this task. To declare an array constant `a`, you can use the SMT-LIB command `(declare-fun a () (Array Int Int))`. The function applications for the *select* function and the *store* function are written as usual, e.g. `(select a i)` and `(store a i v)`.

Exercise 6: Theory of Arrays 2 Points

Formalize the following statements as first order logic formulas.

(a) The array a has the value 0 at every index except at index 5, where the value is 23.

(b) The array a contains no duplicate values between the indices 0 and 10 inclusive.

[1] `https://ultimate.informatik.uni-freiburg.de/?ui=int&tool=referee`