Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Fabian Huch

WS 2022/23
10.01.2022

Semantics of Programming Languages

Exercise Sheet 10

Exercise 10.1 Using the VCG

Use the VCG to prove correct a multiplication and a square root program:
*Note: With the template, use *unbundle COM/unbundle ACOM* before writing down a *com/acom* definition, otherwise parsing will be slow.

```
definition MUL :: com where
"MUL =
  ''z''::=N 0;;
  ''c''::=N 0;;
  WHILE (Less (V ''c'') (V ''y'')) DO (
    ''z''::=Plus (V ''z'') (V ''x'');;
    ''c''::=Plus (V ''c'') (N 1))"

theorem MUL_partially_correct:
"⊢ {λs. 0 ≤ s ''y'' ∧ s=sorig}
     MUL
   {λs. s ''z'' = s ''x'' * s ''y'' ∧ (∀ v. v∉{''z'',''c''} ⟶ s v = sorig v)}"

definition SQRT :: com where
"SQRT =
  ''r'' ::= N 0;;
  ''s'' ::= N 1;;
  WHILE (Not (Less (V ''x'') (V ''s''))) DO (
    ''r'' ::= Plus (V ''r'') (N 1);;
    ''s'' ::= Plus (V ''s'') (V ''r'');;
    ''s'' ::= Plus (V ''s'') (V ''r'');;
    ''s'' ::= Plus (V ''s'') (N 1)
  )"

theorem SQRT_partially_correct:
"⊢ {λs. s=sorig ∧ s ''x'' ≥ 0}
     SQRT
   {λs. (s ''r'')^2 ≤ s ''x'' ∧ s ''x'' < (s ''r''+1)^2 ∧ (∀ v. v∉{''s'',''r''} ⟶ s v = sorig v)}"
```

Exercise 10.2 Total Correctness

Prove total correctness of the multiplication and square root program.

Rotated rule for sequential composition:

lemmas *Seq_bwd = Hoare_Total.Seq[rotated]*

Prove the following syntax-directed conditional rule (for total correctness):

lemma *IfT*:
assumes "$\vdash_t$ {*P1*} c_1 {*Q*}"
and "$\vdash_t$ {*P2*} c_2 {*Q*}"
shows "$\vdash_t$ {$\lambda s.$ (*bval b s* $\longrightarrow$ *P1 s*) $\wedge$ ($\neg$ *bval b s* $\longrightarrow$ *P2 s*)} *IF b THEN* c_1 *ELSE* c_2 {*Q*}"

lemmas *hoareT_rule[intro?] = Seq_bwd Hoare_Total.Assign Hoare_Total.Assign' IfT*

theorem *MUL_totally_correct*:
"$\vdash_t$ {$\lambda s.$ $0 \le s\ ''y'' \wedge s{=}sorig$}
MUL
{$\lambda s.$ $s\ ''z'' = s\ ''x'' * s\ ''y'' \wedge (\forall v.\ v \notin \{''z'', ''c''\} \longrightarrow s\ v = sorig\ v)$}"

theorem *SQRT_totally_correct*:
"$\vdash_t$ {$\lambda s.$ $s{=}sorig \wedge s\ ''x'' \ge 0$}
SQRT
{$\lambda s.$ $(s\ ''r'')\hat{}2 \le s\ ''x'' \wedge s\ ''x'' < (s\ ''r''+1)\hat{}2 \wedge (\forall v.\ v \notin \{''s'', ''r''\} \longrightarrow s\ v = sorig\ v)$}"

Homework 10.1 Squaring (5 pts)

Submission until Monday, Jan 16, 23:59pm.

Consider the following program for squaring a non-negative integer:

```
z := 1;
a := 0;
WHILE 0 < n DO (
  a := a + z;
  z := z + 2;
  n := n + (-1)
)
```

Here is a picture illustrating the algorithm:

```
1 2 3 4
2 2 3 4
3 3 3 4
4 4 4 4
```

Using the VCG, prove that this algorithm correctly squares non-negative integers:

theorem *SQUARE_correct*:
"$\vdash \{\lambda s.\ s\ ''n'' \geq 0 \wedge s{=}s_0\}\ SQUARE\ \{\lambda s.\ s\ ''a'' = s_0\ ''n'' * s_0\ ''n''\}$"

Hints:

- Your invariant should state something about all three variables, n, a, and z.
- *algebra_simps* can help to solve any remaining verification conditions.

Homework 10.2 Be Original!

Submission until Monday, Jan 16, 23:59pm.
We're giving you one more week to finalize and polish your developments!
In total, this exercise will be worth 15 points, plus bonus points for nice submissions.