Technische Universität München
Fakultät für Informatik
Prof. Tobias Nipkow, Ph.D.
Simon Wimmer

WS 2019/20
13.01.2020

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:

```
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 "$\vdash_t$ {λ*s*. (*bval b s* ⟶ *P1 s*) ∧ (¬ *bval b s* ⟶ *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*:
"$\vdash_t$ {λ*s*. *0* ≤ *s* ''*y*'' ∧ *s*=*sorig*}
    MUL
  {λ*s*. *s* ''*z*'' = *s* ''*x*'' ∗ *s* ''*y*'' ∧ (∀ *v*. *v*∉{''*z*'',''*c*''} ⟶ *s v* = *sorig v*)}"

theorem *SQRT_totally_correct*:
"$\vdash_t$ {λ*s*. *s*=*sorig* ∧ *s* ''*x*'' ≥ *0*}
    SQRT
  {λ*s*. (*s* ''*r*'')ˆ*2* ≤ *s* ''*x*'' ∧ *s* ''*x*'' < (*s* ''*r*''+*1*)ˆ*2* ∧ (∀ *v*. *v*∉{''*s*'',''*r*''} ⟶ *s v* = *sorig v*)}"

Homework 10.1 Using the VCG

Submission until Monday, January 20, 10:00am.

Consider the following IMP program that given a value *n* ≥ *0* in variable ''*n*'' computes *2*ˆ*n* and stores the result in variable ''*x*''.

definition
  "*POWER2* ≡
  ''*x*'' ::= *N 1*;;
  WHILE Less (*N 0*) (*V* ''*n*'') *DO* (
    ''*x*'' ::= *Plus* (*V* ''*x*'') (*V* ''*x*'');;
    ''*n*'' ::= *Plus* (*V* ''*n*'') (*N* (−*1*))
)"

Using the VCG, prove the following Hoare triple, stating the program is correct.

theorem *POWER2_correct*:
"⊢ {λ*s*. *s* ''*n*'' = *n* ∧ *n* ≥ *0*}
   POWER2
  {λ*s*. *s* ''*x*'' = *2* ˆ *nat n*}"

Hint: The theorem collection *algebra_simps* and sledgehammer can be helpful to discharge proof obligations about arithmetic.

Homework 10.2 Collecting Semantics

Submission until Monday, January 20, 10:00am.

This question concerns the iterative computation of the collecting semantics of the following annotated command:

```
IF x < 0 THEN {A1}
  {A2}
  WHILE 0 < y DO
    {A3}
    (y := y + x {A4})
  {A5}
ELSE {A6} SKIP {A7}
{A8}
```

Show how the annotations change with each application of the step function. Fill in this table to show how the process evolves until a least fixpoint is reached:

	0	1	2	3	4	5	6	7	8	9	10
$A1$	$\emptyset$										
$A2$	$\emptyset$										
$A3$	$\emptyset$										
$A4$	$\emptyset$										
$A5$	$\emptyset$										
$A6$	$\emptyset$										
$A7$	$\emptyset$										
$A8$	$\emptyset$										

Let S be $\{<x := -2,\ y := 3\rangle, \langle x := 1,\ y := 2\rangle\}$ when you execute the step function. For brevity, write such a set of states as $-2{,}3 \mid 1{,}2$ when you fill in the table. Entries that do not change can be left blank.