

This document contains the solution for Test 3. I have added additional comments, usually preceded by the keyword **comment**.

Test 3 - Solution

1. The annotated SPL code given in question 1 gives us enough information to deduce the typechecking rules of the modulus operator.

- We are told that `x := (y+2) % 5` is correctly typed. A quick inspection to the right-hand side of the assignment (an expression that uses the modulus operator) tells us that both operands, `(y+2)` and `5`, are of type integer.
- Also, the modulus operator must return an integer since an assignment requires that the types of the assigned variable and the expression (right-hand) are the same.
- The next two lines suggest that the modulus operator does not accept float operands.
- The final line can be used to verify that the modulus operator returns an int. Type-checking fails because the assigned variable is a float.

Answer:

```
typecheck(Modulus(e1,e2),f,stable) =  
  t1 = typecheck(e1,f,stable)  
  ReportError if (t1 != int)  
  t2 = typecheck(e2,f,stable)  
  ReportError if (t2 != int)  
  return int
```

that is, `e1` and `e2` must be correctly typed, and their type must be int. The modulus operator (as defined in this example) returns an integer. Could we have a version that accepts floats?

2. Typechecking.

The errors are:

- Line 7: The type of the right-hand side of the assignment must match the type of the assigned variable. In this case, the type of the right-hand side is float (the addition of two floats returns a float) but the assigned variable is an int.
- Line 9: The function declaration of `g(int,float)` indicates that it requires two parameters: an int and a float. However, the function call `g(x,y)` passes two integers - the second argument has the wrong type.
comment: Note the return type of `g`, boolean, matches the type of the left-hand side of the assignment (the type of variable `b`).
- Line 11: The `+` operation requires both operands to be of the same type, int or float.
- Line 15: Undeclared variable `z`.

comment: Line 10 is correct. First of all, the test/condition in an IF statement must be an expression. Variables are expressions so the syntax of line 10 is correct. Furthermore, the type of the test must be boolean: the type of expression `b` is boolean as well.

3. Stack frames

Given the following SPL program extract:

```
int fone(int x, float y) {  
  int a; int b;  
  float c;  
  // body is not important  
  d := ftwo((float) c, y, 15.5);
```

```

    // more body comes here
    return (float) c;
}

boolean ftwo(float p1, int p2, float p2) {
    // body is not important
}

```

comment: Note that I have added two (float) type casts to fix the type errors.

- a. The stack frame layout of function fone is:

offset	content	type(size)
0	local a	int(1)
1	local b	int(1)
2	local c	float(2)
4	local d	boolean(1)
5	return address	address(1)
6	returned value	boolean(1)
7	ftwo param p3	float(2)
9	ftwo param p2	int(1)
10	ftwo param p1	float(2)

Frame size = 12

comment: A function's stack frame (assuming that it does not have to store any temporaries or values stored in registers) stores:

- its local variables (fone's a,b,c,and d)
 - If it makes any function calls (in our example, fone calls ftwo):
 - * The return address. This is the address of the instruction where execution resumes after the call to ftwo terminates i.e. the address of the instruction immediately after `d := ftwo((float) c, y, 15.5)`. The virtual machine needs to know where to return! The size of this is always one (the size of an address).
 - * The value returned by the called function (ftwo's returned value, a boolean). The size of this depends on the size of the return type of the called function.
 - * The parameters passed to the called function (ftwo's parameters p1,p2 and p3)
- b. How does fone access its parameter y? Write TPL code that stores 10.5 into y.

Parameters x and y are located above fone's frame pointer (FP). They have offsets -1 and -3, respectively. The global location of y is FP(-3). If we want to store 10.5 into y then we have to write:

STORE 10.5, FP(-3)

comment: The space to store fone's x and y is part of the stack frame of fone's caller (not shown here since this is just a program extract).