6.0 ECTS/4.5h VU Programm- und Systemverifikation (184.741)

June 18, 2019

Kennzahl (study id)	Matrikelnummer (student id)	Familienname (family name)	Vorname (first name)	Gruppe (version)
				A

1.) Coverage

Consider the following program fragment and test suite:

```
unsigned gcd (unsigned m, unsigned n) {
  unsigned i;
  if (m > n) {
    i = n;
  } else {
    i = m;
  }
  bool done = false;
  while ((i > 0) && !done) {
    if ((m % i == 0) && (n % i == 0) {
        done = true;
    } else {
        i = i - 1;
    }
  }
  return i;
}
```

Inputs		Outputs
`m`	`n`	return value
0	1	0
1	0	0
1	1	1
2	3	1

(a) **Control-Flow-Based Coverage Criteria**

Indicate (✓) which of the following coverage criteria are satisfied by the test-suite above.

	satisfied	
Criterion	yes	no
statement coverage		
decision coverage		
condition coverage		
modified condition/decision coverage		

For each coverage criterion that is *not* satisfied, explain why this is the case:

(6 points)

(b) **Data-Flow-Based Coverage Criteria**

Indicate (✓) which of the following coverage criteria are satisfied by the test-suite above (here, the parameters of the function do not constitute definitions, the `return` statement is a c-use):

	satisfied	
Criterion	yes	no
all-defs		
all-c-uses		
all-p-uses		
all-c-uses/some-p-uses		
all-p-uses/some-c-uses		

For each coverage criterion that is not satisfied, explain why this is the case:

(9 points)

(c) Consider the two coverage criteria below.

- If the test-suite from above does not satisfy the coverage criterion, augment it with the *minimal* number of test-cases such that this criterion is satisfied. If full coverage cannot be achieved, explain why.
- If the coverage criterion is already achieved, explain why.

MC/DC

Inputs		Outputs
`m`	`n`	result

all-p-uses

Inputs		Outputs
`m`	`n`	result

(2 points)

(d) Consider the expression $((a \vee b) \wedge c)$, where a, b, and c are Boolean variables. Provide a *minimal* number of test cases such that modified condition/decision coverage is achieved for the expression. Clarify for each test case *which* condition(s)/value(s) independently affect(s) the outcome.

MC/DC

Inputs			Outcome
`a`	`b`	`c`	`(a || b) && c`

(3 points)

2.) Hoare Logic

Prove the Hoare Triple below (assume that the domain of all variables except `done` in the program are the unsigned integers including zero, i.e., $\mathtt{i}, \mathtt{m}, \mathtt{n} \in \mathbb{N} \cup \{0\}$, and that `done` is a Boolean variable). You need to find a sufficiently strong loop invariant.

Annotate the following code directly with the required assertions. Justify each assertion by stating which Hoare rule you used to derive it, and the premise(s) of that rule. If you strengthen or weaken conditions, explain your reasoning.

$\{\mathsf{true}\}$

```
if (m > n)

  i = n;

else

  i = m;

done = false;

while ((i > 1) && !done) {

  if ((m % i == 0) && (n % i == 0))

    done = true;

  else

    i = i - 1;

}
```

$\{(\mathsf{i} = 0) \vee (\mathsf{m} \ \% \ \mathsf{i} = 0)\}$

(10 points)

3.) Invariants Consider the following program, where `x` and `y` are non-negative natural numbers (possibly 0):

```
x = y + 1;
while (x != y) {
  x = x + (y % 2);
  y = y + (x % 2);
}
```

Consider the formulas below; tick the correct box (☑) to indicate whether they are loop invariants for the program above.

- If the formula is an inductive invariant for the loop, provide an informal argument that the invariant is inductive.
- If the formula P is an invariant that is *not* inductive, give values of `x` and `y` before and after the loop body demonstrating that the Hoare triple

$$\{P \wedge B\} \quad \texttt{x} = \texttt{x} + (\texttt{y}\%2);\ \texttt{y} = \texttt{y} + (\texttt{x}\%2); \quad \{P\}$$

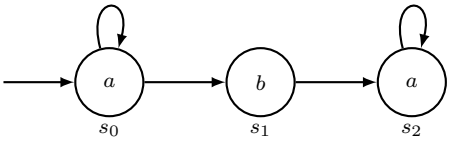
  (where B is `(x != y)`) does not hold.
- Otherwise, provide values of `x` and `y` that correspond to a reachable state showing that the formula is *not* an invariant.

$(x - y) \leq 1$	☐ Inductive Invariant	☐ Non-inductive Inv.	☐ Neither
Justification:			
$(x - y) \leq 2$	☐ Inductive Invariant	☐ Non-inductive Inv.	☐ Neither
Justification:			
$(x - y)\%2 = 1$	☐ Inductive Invariant	☐ Non-inductive Inv.	☐ Neither
Justification:			

(10 points)

4.) Temporal Logic

(a) Consider the following Kripke Structure:



For each formula, give the states of the Kripke structure for which the formula holds. In other words, for each of the states from the set $\{s_0, s_1, s_2\}$, consider the computation trees starting at that state, and for each tree, check whether the given formula holds on it or not.

i. $\mathbf{A}(\mathbf{F}\,\mathbf{G}\,a)$

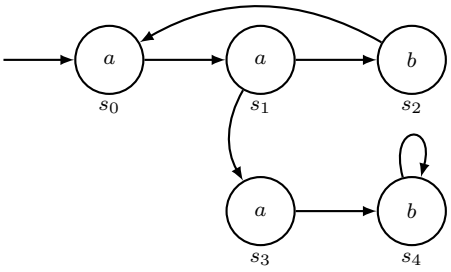
ii. $\mathbf{AF}\,\mathbf{AG}\,a$

iii. $\mathbf{A}(a \wedge \mathbf{X}\,a)$

iv. $\mathbf{E}(b\,\mathbf{U}\,a)$

(4 points)

(b) Consider the following Kripke Structure with initial state s_0:



i. Does the LTL formula $\mathbf{AF}\,\mathbf{X}\,b$ hold in the initial state s_0?

☐ yes ☐ no

Justify your answer!

ii. Does the CTL formula $\mathbf{AF}\,\mathbf{AX}\,b$ hold in the initial state s_0?

☐ yes ☐ no

Justify your answer!

iii. Do the formulas (i) and (ii) above express the same property?

☐ yes ☐ no

If not, explain why.

(6 points)

5.) Decision procedures

(a) Consider the following formulas in propositional logic; are they satisfiable? If yes, provide a satisfying assignment over booleans, if not, give the reasoning that leads to this conclusion.

$$(\neg a \vee \neg b) \ \wedge \ c \ \wedge \ (b \vee d) \ \wedge \ \neg d \ \wedge \ e \ \wedge \ (\neg e \vee \neg c \vee a) \tag{1}$$

$$f \ \wedge \ (\neg g \vee f) \ \wedge \ (h \vee \neg f) \ \wedge \ (g \vee h) \ \wedge \ (\neg g \vee \neg h) \tag{2}$$

(2 points)

(b) Consider the following formulas in Equality Logic; are they satisfiable? If yes, provide a satisfying assignment over integers, if not, give the reasoning based on equivalence classes that leads to this conclusion.

$$i = j \ \wedge \ k = \ell \ \wedge \ k \neq m \ \wedge \ \ell \neq i \ \wedge \ m = i \ \wedge \ f = j \tag{3}$$

$$n \neq o \wedge p = q \wedge r = s \wedge r = t \wedge t = n \wedge s = q \wedge q = g \wedge g = o \wedge r = g \wedge t = q \tag{4}$$

(2 points)

(c) Check the satisfiability of the following SMT formulas. Assume that $u, v, w, x, y, z \in \mathbb{Z}$ are integer constants, and $f : \mathbb{Z} \to \mathbb{Z}$ and $g : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ are unary and binary uninterpreted functions over integers respectively. Whenever a formula is satisfiable, give a satisfying assignment for it, i.e., integer values for all variables and function intepretations over integers that make the formula true under the assignment. Whenever a formula is not satisfiable, give a reason why it is unsatisfiable.

$$g(3, y) = 5 \ \wedge \ g(y, 3) = 5 \ \wedge \ g(y, x) \neq g(x, y) \tag{5}$$

$$g(1, x) = 2 \ \wedge \ g(1, x) = g(x, 1) \ \wedge \ f(f(x)) = g(1, x) \\ \wedge \ f(f(f(1))) = 1 \ \wedge \ f(1) \neq g(x, 1) \ \wedge \ 1 = f(x) \tag{6}$$

$$g(z, z) = z \ \wedge \ g(u, u) = u \\ \wedge \ (g(z, z) = 0 \vee g(z, z) = 1) \ \wedge \ (g(u, u) = 0 \vee g(u, u) = 1) \\ \wedge \ g(z, u) = g(u, z) \ \wedge \ g(3, 2) = w \ \wedge \ g(2, 3) = v \ \wedge \ v \neq w \tag{7}$$

(6 points)