

Unterweisung für Studierende COVID-19-Schutzmaßnahmen an der TU Wien

Diese Unterweisung bezieht sich auf die COVID-19 Schutzmaßnahmen, die an der TU Wien einzuhalten sind.

Inhalt der Unterweisung (siehe beiliegendes Infoblatt)

Allgemeine Vorgaben

- Die Hände sind regelmäßig und gründlich zu reinigen.
- Schutzabstand einhalten.
- Richtige Husten- und Niesetikette beachten (bitte Ellenbogenbeuge vorhalten)
- Händeschütteln und Körperkontakt vermeiden
- Es dürfen keine Gruppen innerhalb der Gebäude bzw. auf dem Gelände der TU Wien gebildet werden.
- Bei Erkrankung UNBEDINGT zu Hause bleiben.

Sicherheitsabstand einhalten

Die wichtigste Schutzmaßnahme gegen COVID-19-Infektionen ist der Sicherheitsabstand von mindestens 1,5 m. Dieser muss innerhalb aller Areale der TU Wien eingehalten werden.

Schutzmasken

Das Tragen von Mund-Nasen-Schutzmasken (MNS-Masken) wird in allen allgemeinen Räumlichkeiten dann empfohlen, wenn ein Kontakt mit anderen Personen sehr wahrscheinlich ist und nicht sichergestellt werden kann, dass der notwendige Sicherheitsabstand eingehalten wird.

In den institutsspezifischen Arbeitsräumen ist das Tragen von MNS-Masken (bei Bedarf auch Visiere) dann verpflichtend, wenn sich mehr als eine Person im Raum aufhält und der Abstand von 1,5 m in Büros, Labors, Werkstätten und Hörsälen nicht durchgehend eingehalten werden kann.

Präventions- und Hygienemaßnahmen

Grundsätzlich gilt: Waschen Sie Ihre Hände regelmäßig und gründlich mit Seife. Das ist für die Haut weniger belastend als Desinfektion und erhält die als Ansteckungsschutz nötige Hautbarriere.

Direkt bei den Eingängen sind Desinfektionsstellen eingerichtet, an denen man sich beim Betreten der Gebäude der TU Wien die Hände desinfizieren kann.

Allgemein: Auf die Hygiene ist unbedingt zu achten (regelmäßiges Händewaschen, keine Berührungen der Mitmenschen, Atemhygiene)!

Risikominimierung

Jede_r, der Krankheitssymptome wie Husten, Fieber, Geschmacksverlust, etc. aufweist oder befürchtet, muss jedenfalls zu Hause bleiben und sofort die telefonische Gesundheitsberatung unter 1450 kontaktieren.

Zusatzinformationen

Weitere Informationen können Sie auch unter folgendem Link nachlesen:
<https://colab.tuwien.ac.at/pages/viewpage.action?pageId=11638263>

Ich erkläre hiermit, dass ich über die Schutzmaßnahmen unterrichtet wurde, diese verstanden und zur Kenntnis genommen habe und verpflichte mich, diese einzuhalten.

.....
Datum

.....
Name und Unterschrift

6.0 ECTS/4.5h VU Programm- und Systemverifikation (184.741) July 31, 2020				
Kennzahl (study id)	Matrikelnummer (student id)	Familienname (family name)	Vorname (first name)	Platz (seat)

1.) Coverage

Consider the following program fragment and test suite:

```

unsigned gcd (unsigned x, unsigned y) {
    unsigned m, k;
    if (x > y) {
        k = x;
        m = y;
    } else {
        k = y;
        m = x;
    }
    while (m != 0) {
        unsigned r = m % k;
        k = m;
        m = r;
    }
    return k;
}

```

Inputs		Outputs
x	y	return value
0	0	0
0	1	0
3	2	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
path coverage		
statement coverage		
branch coverage		
decision coverage		

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

(5 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-du-paths		

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

(8 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.

all-c-uses

Inputs		Outputs
x	y	result

all-uses

Inputs		Outputs
x	y	result

(2 points)

(d) Indicate whether the following statements are true or false!

Statement	True	False
If a test-suite achieves all-c-uses and all-defs coverage, it also achieves all-c-uses/some-p-uses coverage.	☐	☐
If a test-suite achieves path coverage, it also achieves statement coverage.	☐	☐
If a test-suite does not achieve all-uses coverage, it cannot achieve all-c-uses either.	☐	☐
For any given program, it is always possible to achieve MC/DC coverage.	☐	☐
If full statement coverage cannot be achieved, then the program contains unreachable code.	☐	☐

(5 points)

2.) Hoare Logic

Prove the Hoare Triple below (assume that the domain of all variables in the program are the integers, i.e., $t, m, n \in \mathbb{Z}$. 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.

Note: No points for assertions that were not clearly derived by using one of the rules from the lecture!

```
{true}
```

```
if (m > n) {
```

```
    int t = n;
```

```
    n = m;
```

```
    m = t;
```

```
} else {
```

```
    skip;
```

```
}
```

```
while (m < n) {
```

```
    m = m + 1;
```

```
}
```

```
{(m = n)}
```

(10 points)

3.) Invariants Consider the following program, where t , x and y are integer values in $\mathbb{Z}$ (that means no over- or underflow can happen):

```

if (x > y) {
    int t = x; x = y; y = t;
}
while (y > x) {
    x = x + 1;
    y = y - 1;
}

```

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

$$\{P \wedge B\} \quad x = x + 1; y = y - 1; \quad \{P\}$$

(where B is $(y > x)$) does not hold.

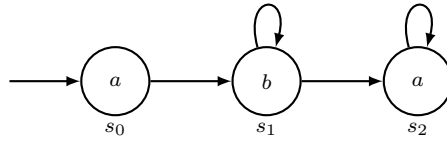
- Otherwise, provide values of x and y that correspond to a reachable state showing that the formula is *not* an invariant.

$(x \leq y)$ Justification:	☐ Inductive Invariant	☐ Non-inductive Inv.	☐ Neither
$(x - y) \leq 1$ Justification:	☐ Inductive Invariant	☐ Non-inductive Inv.	☐ Neither
$(y - x) + 2 \neq 0$ Justification:	☐ Inductive Invariant	☐ Non-inductive Inv.	☐ Neither

(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. **AFG** a

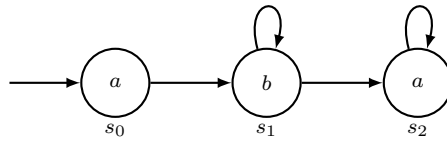
ii. **AFEG** b

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

iv. **A** $(b \mathbf{U} a)$

(4 points)

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



i. Does the LTL formula **AFG** b hold in the initial state s_0 ?

☐ yes ☐ no

Justify your answer!

ii. Does the CTL formula **AFEG** 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 formula in propositional logic; is it satisfiable? If yes, provide a satisfying assignment, if not, give the reasoning that leads to this conclusion.

$$\begin{aligned} & (x_{11} \vee x_{12}) \wedge (x_{21} \vee x_{22}) \wedge (x_{31} \vee x_{32}) \wedge \\ & (\neg x_{11} \vee \neg x_{21}) \wedge (\neg x_{11} \vee \neg x_{31}) \wedge (\neg x_{12} \vee \neg x_{22}) \wedge (\neg x_{12} \vee \neg x_{32}) \wedge \\ & (\neg x_{21} \vee \neg x_{31}) \wedge (\neg x_{22} \vee \neg x_{32}) \quad (1) \end{aligned}$$

(4 points)

- (b) Consider the following formulas in Equality Logic and Equality Logic with Uninterpreted Functions (EUF); 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.

$$i = j \wedge j = k \wedge k = l \wedge l \neq m \wedge l \neq n \wedge m = n \wedge o \neq p \wedge o = q \quad (2)$$

$$i = j \wedge j = k \wedge k = l \wedge l \neq n \wedge m = n \wedge g(i) \neq g(m) \wedge f(i) \neq f(l) \quad (3)$$

- (c) Construct an Ordered Binary Decision Diagram that encodes the following Boolean formula:

$$(x_1 \oplus x_2) \wedge (x_1) \tag{4}$$

(where $\oplus$ is the exclusive or operator).

Illustrate the construction steps and not just the final result!

(4 points)