Exam Software Testing & Risk Assessment

June 30, 2021
9:00 — 12:00

Remarks.

- Please provide all answers in English.
- Explain every solution. Without a proper explanation, you may not receive any points for an otherwise correct answer.
- The exam is divided in sections. When answering questions, start each section on a **new page**.
- You are allowed a 1 A4 cheat sheet, double printed/written.

Good luck!

1 Risk Management

The following article appeared on CBS News:

Highly contagious Delta variant could cause next COVID-19 wave: "This virus will still find you"

Whether it's airports, NBA games, or concerts, crowds are gathering across the nation as Americans start a return to pre-pandemic life.

But with just 45% of Americans fully vaccinated and only 16 states that have fully vaccinated more than half of their populations, health experts are worried about the spread of the highly contagious Delta variant. It is 60% more contagious than the Alpha variant discovered in the U.K., which was the last variant of major concern, according to infectious disease expert Michael Osterholm.

"In the areas where they have large pockets of unvaccinated people, we can surely expect to see surges in cases, in some situations challenging the health care capacity of that local area," Osterholm told "CBS This Morning" lead national correspondent David Begnaud.

A hospital dealing with an overfill of patients is Mercy Hospital in Springfield, Missouri. Mercy Hospital President Craig McCoy said the hospital is "holding patients in the ER, waiting on admissions, waiting on discharges on any given day".

In Springfield, only 32% of the surrounding county is vaccinated, and COVID-19 hospitalizations are up more than 210% since June 1. Perhaps most alarming—90% of all COVID samples being sequenced from that county are testing positive for the Delta variant.

1. Describe three aspects / concepts of risk management that are applicable to this article. *(3 points)*

   Risk: we have uncertainty and high impact. Impact: there are consequences on safety. Vaccination is a risk management strategy. Risk perception is important: people choose not to vaccinate themselves, to avoid the risk of side effects (at the cost of being infected).

2. Mention 2 risk strategies that are deployed and categorize them in the four risk strategy classes (the four T's). *(2 points)*

   Vaccination: Treatment: lower probability of infection and lower impact. Some people apply a risk avoidance strategy

2 Fault trees

Consider the fault tree F in Figure 1 for a container seal design. This fault tree has seven basic events (B1, B2, ..., B7). The probability for the basic events to fail are given by $p_{B1} = 0.8$, $p_{B2} = p_{B4} = p_{B5} = p_{B6} = 0.1$, $p_{B3} = 0.3$, and $p_{B7} = 0.2$. The failure probability of the top event is P.

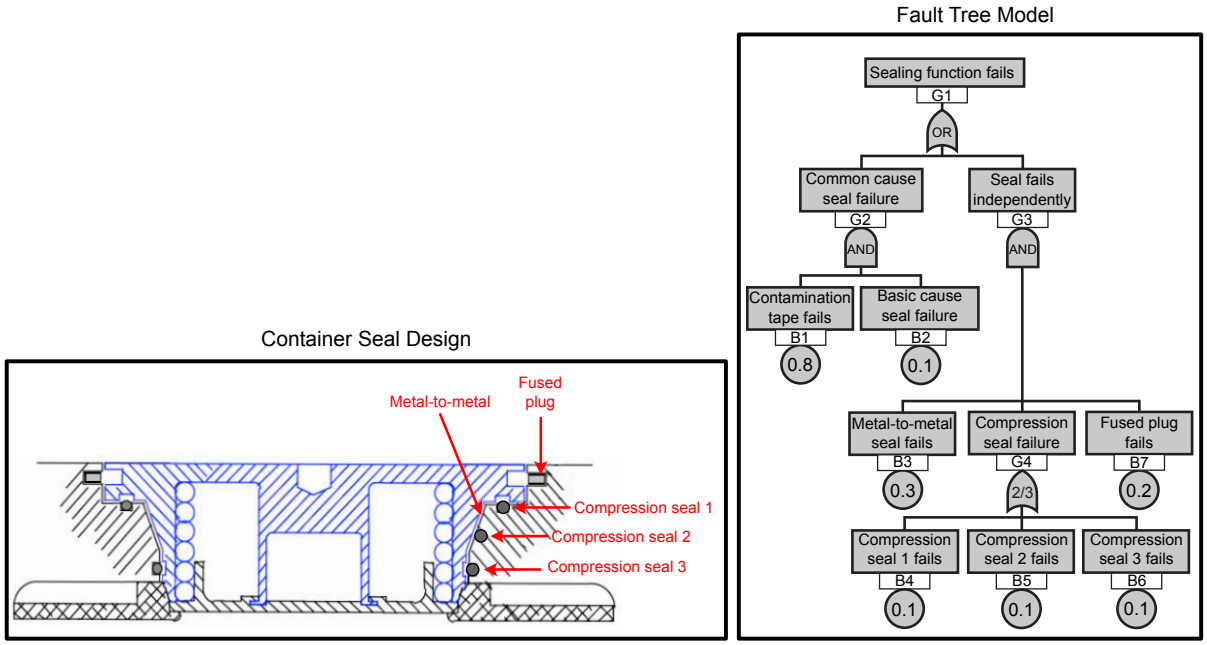


Figure 1: On the left, container seal design system. On the right the associated Fault Tree model F.

1. List all minimal cut sets in F. *(1 point)*

   There are 4 MCSs: $\{B1, B2\}$, $\{B3, B4, B5, B7\}$,$\{B3, B5, B6, B7\}$,$\{B3, B4, B6, B7\}$

2. Compute the exact failure probability of the gate G4, *compression seal failure* ($p(G4)$). *(1 point)*

   Let $p = p_{B4} = p_{B5} = p_{B6}$, then: $p(G4) = 1 - 3 * (p)(1-p)^2 - (1-p)^3 = 0.028$

3. Compute the exact failure probability of the top event (P). *(2 points)*

   $P = p_{B1} * p_{B2} + p_{B3} * p_{B7} * p(G4) - (p_{B1} * p_{B2}) * (p_{B3} * p_{B7} * p(G4))$
   $P = 0.8 * 0.1 + 0.3 * 0.028 * 0.2 - (0.8 * 0.1 * 0.3 * 0.028 * 0.2) = 0.0815$

4. Approximate the failure probability of the top event (P) using the *minimal cut sets method*, and compare this approximation with the exact solution, what can you conclude? *(2 points)*

The approximated answer using the minimal cut sets method is:
$P^* = p_{B1} * p_{B2} + p_{B3} * p_{B4} * p_{B5} * p_{B7} + p_{B3} * p_{B5} * p_{B6} * p_{B7} + p_{B3} * p_{B4} * p_{B6} * p_{B7}$
$P^* = 0.8 * 0.1 + 3 * 0.3 * 0.2 * 0.1^2 = 0.0818 > 0.0815$
Conclusion: The MCSs method is more conservative.

5. Based on the fault tree provided in Figure 1, provide at least one strategy to mitigate the occurrence of the top event *sealing function fails. (2 points)*

The gate $G2$ is more vulnerable than the gate $G3$. Thus, I would reduce the failure probability of basic events (e.g., contamination tape fails), by increasing the quality of the component.

3 Exponential distributions

1. Consider a random variable modeling a component failure time (measured in weeks) and that is exponentially distributed expected value $\frac{1}{5}$. What is the probability that the component experiences no failure within the first four weeks? *(2 points)*

   Expected value $\frac{1}{5}$, so $\lambda = 5$, so $P[X > 4] = e^{-4 \cdot 5} = e^{-20}$

   Answer (a): $P[X > 20 | X > 15] = P[X > 5]$ so not complete

   Answer (a): $P[X > 20 | X > 15] = P[X > 5]$ follows directly from the definition of memorilesness. It then follows that (d) is also correct. Moreover, (c) is correct because these probabilities are all 0.

2. Give an example of a probability distribution that is not memoriless. (As for the other questions: explain your answer.) *(2 points)*

   Numerous possibilities: normal, deterministic, etc.

4 Blackbox Testing.

The function `int days_since_corona(int day, month, year)` takes three integers that are interpreted as a date. The program yields the number of days since March 11, 2020 (= when the WHO officially declared Covid as a pandemic). Dates that result in a negative number of days are not allowed.

We assume that this function has been type checked, so you do not have to check for non-integer values, or the wrong number of arguments.

1. Use the equivalence partitioning technique to divide the input into suitable equivalence classes and give a test suite that covers all equivalence classes. *(3 points)*

Equivalence classes:

Nr	*Description*	*Valid/Invalid*	*Expected output*
1	year is less than 2020	invalid	(invalid year) error
2	month is not in [1..12]	invalid	(invalid month) error
3	day is not in [1..daysInMonth(month, year)]	invalid	(invalid date / invalid day) error
4	date is valid, but before March 11, 2020	invalid	(negative date) error
5	date is valid, and March 11, 2020 or later	valid	number of days since March 11, 2020

Tests:

Test case	*d*	*m*	*y*	*Exp. output*	*Equivalence classes*
1	1	1	2019	error	1
2	1	20	2020	error	2
3	40	1	2020	error	3
4	1	3	2020	error	4
5	20	4	2020	40	5

2. Extend the test suite using the principle of boundary value analysis. *(3 points)*

Test case	*d*	*m*	*y*	*Exp. output*	*Equivalence classes*
6	1	1	$+\infty$	$+\infty$	1, 5
7	1	12	2020	(some value)	2, 5
8	31	3	2020	20	3, 5
9	11	3	2020	0	4, 5
10	10	3	2020	error	4, 5

3. Extend the test suite using the principle of error guessing. *(1 point)*

Test case	*d*	*m*	*y*	*Exp. output*	*Reasoning*
11	29	2	2024	error	Leap year (1/2)
12	28	2	2024	(some value)	Leap year (2/2)
13	31	4	2020	error	Months may have 31 days incorrectly

Present each test suite in a table.

5 Whitebox Testing.

We consider the following transformation on programs. Given a program A, we obtain a program A' by replacing all statements of the form

```
if C & D
then S
endif
```

by

```
if C
then
    if D
    then S
    endif
endif
```

1. Do these transformations preserve decision coverage? *(3 points)*

   (I.e. if a test suite T has 100% decision coverage on program A, does T have 100% decision coverage for the transformed program A' as well?) If your answer is "no", present a program A, its transformation A', and a test suite T such that T has 100% coverage on A, but not on A'.

   No.
   We give a counter-example. The inputs $\{C = \text{TRUE}, D = \text{TRUE}\}$ and $\{C = \text{FALSE}, D = \text{TRUE}\}$ results in decision coverage in A. However, in A' the outcome where decision "if D" is false is not covered (this could be achieved by adding input $\{C = \text{TRUE}, D = \text{FALSE}\}$). Thus, decision coverage is not preserved.

2. Same question, but for condition coverage. *(3 points)*

   If we assume that D is only evaluated when $C = \text{TRUE}$, then yes. Since C is the first condition, it is still fully covered in A'. Because of our assumption, D must have been fully covered in A while $C = \text{TRUE}$, simultaneously; consequently, D is still fully covered in A'.
   Otherwise, no. We give a counter-example. The inputs $\{C = \text{TRUE}, D = \text{FALSE}\}$ and $\{C = \text{FALSE}, D = \text{TRUE}\}$ results in condition coverage in A, but not in A'.

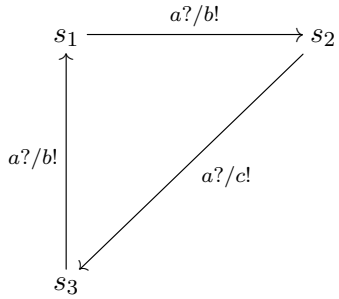


Figure 2: The Mealy Machine A; s_1 is initial.

6 Mealy Machine testing

Consider the Mealy Machine A in Figure 2.

1. If possible, give test suites of A with the following properties. If such a test suite exists, explain why your test suite meets the desired properties. If such a test suite does not exist, explain why.

   (a) A test suite T_1 consisting of 1 test that is sound. *(1 point)*

   Input: $a?$
   Expected output: $b!$

   (b) A test suite T_2 consisting of 1 test that is complete. *(2 points)*

   We assume that we are looking for completeness given that implementations have the same number of states as A or less (otherwise, a complete test suite with 1 test does not exist; we cannot give a test suite with a contradiction, which requires at least 2 tests).
   We identify $a?\,a?$ as a distinguishing sequence. We use this for creating transition tests.
   When we create transition tests, we observe that the transition test for the transition from s_1 to s_3 has all other tests as a prefix. We can therefore leave out those other tests. We are left with
   Input: $a?\,a?\,a?\,a?\,a?$
   Expected output: $b!\,c!\,b!\,b!$
   (Making use of the assumption that an implementation is reduced and strongly connected, one can argue that the last input+expected output is not necessary.)

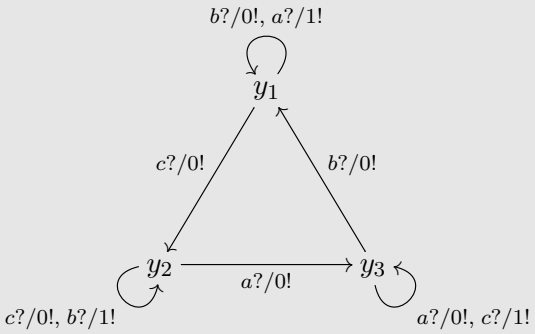
2. Do the following sequences / sets exist for all Mealy machines?

   - If so, explain why.
   - If not, provide a counter example (for readability, provide your counter example below the table).

Present your solutions in the table below. *(3 points)*

sequence	**exists? y/n**	**explanation**
state tours		
distinguishing sequences		
W-sets		

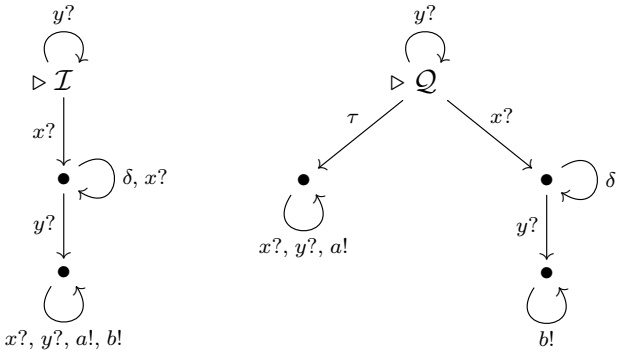
State tours exist for all Mealy machines, because Mealy machines are strongly connected.
DSs do not always exist. Counter-example:



Note that the counter-example is input-enabled, reduced, strongly connected, and deterministic!
W-sets always exist. This is because Mealy machines must be reduced, meaning that for each pair of states there must exist at least one input sequence for which the two states give different outputs (otherwise, the states would be equivalent). By including all such input sequences in a set, one has constructed a W-set.

7 IOCO

Consider QTSs $\mathcal{I}$ and $\mathcal{Q}$:



1. Given that $L^I_{\mathcal{I}} = L^I_{\mathcal{Q}} = \{\ x?,\ y?\ \}$ and $L^O_{\mathcal{I}} = L^O_{\mathcal{Q}} = \{\ a!,\ b!\ \}$, evaluate $\mathcal{I} \sqsubseteq_{\text{ioco}} \mathcal{Q}$. If the outcome is $\bot$, give a trace from $\mathcal{Q}$ that prevents $\mathcal{I} \sqsubseteq_{\text{ioco}} \mathcal{Q}$ from evaluating to $\top$. *(3 points)*

   $\mathcal{I}$ is not well-formed (its first state is quiescent but misses an outgoing δ transition), so $\mathcal{I} \sqsubseteq_{\text{ioco}} \mathcal{Q}$ is, in fact, undefined.
   If this is overlooked, it can be shown that $\mathcal{I} \not\sqsubseteq_{\text{ioco}} \mathcal{Q}$ with $\sigma = x?\, y?\, a!$: the output set is $\{a!, b!\}$ for $\mathcal{I}$ and $\{a!\}$ for $\mathcal{Q}$.

2. Let $\mathcal{A}$ and $\mathcal{B}$ be two well-formed QTSs such that $\mathcal{A}$ is input-enabled. Prove or disprove that

$$traces_{\mathcal{A}} = traces_{\mathcal{B}} \quad \Rightarrow \quad \mathcal{A} \sqsubseteq_{\text{ioco}} \mathcal{B}$$

   (4 points)

   Suppose that

$$\exists \sigma \in \text{Traces}(\mathcal{A}) \,.\, \text{out}_{\mathcal{A}}(\sigma) \not\subseteq \text{out}_{\mathcal{B}}(\sigma)$$

   when the premise $traces_{\mathcal{A}} = traces_{\mathcal{B}}$ holds.
   Then, by the definition of out, it must be the case that

$$\exists \sigma' \in \text{Traces}(\mathcal{A}), a \in L_O \cup \{\delta\} \,.\, \sigma'\, a \in \text{Traces}(\mathcal{A}) \wedge \sigma'\, a \notin \text{Traces}(\mathcal{B})$$

   Using the premise for a substitution, we get

$$\exists \sigma' \in \text{Traces}(\mathcal{A}), a \in L_O \cup \{\delta\} \,.\, \sigma'\, a \in \text{Traces}(\mathcal{A}) \wedge \sigma'\, a \notin \text{Traces}(\mathcal{A})$$

   This expression contains a contradiction, and can never be true. Therefore, the hypothesis holds.