Sensorveiledning Software testing V2024

Part 2 - EP / BVA

Imagine that you are going to test a program, where the input values are integers from the interval 1 - 20 (both limits included). To facilitate the test work, we shall divide the input values into smaller subsets.

a) What are the two conditions (regardless of the program to be tested) that must be met for the subsets of input values to constitute an equivalence partitioning?

1. Delmengdene må være disjunkte, dvs. de kan ikke ha felles elementer.
   $$A_i \cap A_j = \emptyset \; \forall \; i \neq j$$
2. Delmengdene må til sammen utgjøre hele mengden de er utgått fra.
   $$A = A_1 \cup A_2 \cup A_3 \cup \ldots \cup A_n$$

b) Below you see two proposals for equivalence partitioning:

Proposal 1:

- Two valid equivalence partitions:
  - 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
  - 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
- Two invalid equivalence partitions:
  - Integers less than 1
  - Integers greater than 20

Proposal 2:

- Two valid equivalence partitions:
  - The odd numbers 1, 3, 5, 7, 9, 11, 13, 15, 17, 19
  - The even numbers 2, 4, 6, 8, 10, 12, 14, 16, 18, 20
- One invalid equivalence partitions:
  - Integer outside the range 1 - 20 (i.e. numbers less than 1 and greater than 20).

Justify whether the conditions you stated under point a) are met for the two different proposals for equivalence partitioning.

Svar: Ja, de to betingelsene er oppfylt for begge forslagene.
Ingen heltall er med i mer enn én ekvivalensklasse.
I hver av forslagene utgjør de gyldige og ugyldige ekvivalensklassene til sammen alle heltallene.

c) Even if a set of input values can be divided into equivalence partitions, it is not certain that we can use them as a test design technique on the program we are going to test.
What condition must the *program* meet in order for us to be able to use a concrete equivalence partitioning as a test design technique?

Svar:

For alle ekvivalensklassene gjelder følgene: Dataprogrammet må behandle alle elementene innenfor ekvivalensklassen likt.

d) Assume the program fulfills the condition from point c) to be able to use the equivalence partitioning given in Proposal 1 (see point b). Which test values would you select to get the best possible equivalence partitioning coverage with the fewest possible test values?

Svar:

Vi må velge ett element fra hver ekvivalensklasse, f.eks. -1, 5, 15 og 25.

e) Equivalence partitioning is often used in conjunction with *boundary value analysis*. Boundary value analysis cannot, however, be applied to all types of equivalence partitionings. Which necessary, but not always sufficient, condition must be met for the equivalence partitions to contain boundary values?

Svar:

Elementene må kunne sorteres etter en ordning slik at vi kan avgjøre hvilke elementer som er nærmest grensene mellom ekvivalensklassene.

f) Enter any boundary values for one or both of the proposals for equivalence partitioning described in point b).

Svar:

Proposal 1: 0, 1, 10,11, 20,21 (På grensen mellom alle partisjonene, ugyldige og gyldige.)

Proposal 2: 0, 1, 20,21. Det er kun grenseverdier mellom den ugyldige og de gyldige partisjonene: 0, 1, 10,11, 20,21 (De gyldige ekvivalensklassene har **ikke** grenseverdier seg imellom, da det er egenskapen par/odde som avgjør hvilken ekvivalensklasse tallene tilhører):

Part 2 - Decision tables

NB! In this task, you can answer both in the word processor and possibly with sketches/drawings. In that case, use the sketch paper that has been handed out to you in the exam room. See instructions under the taskbar.

Several positions as associate professor in information technology are to be announced at a university in Norway.

In the announcement text, the following requirements for the applicant have been set:

- The applicant must have a PhD within the relevant subject area.
- The applicant must be able to document pedagogical competence at university level.
- The applicant must have language skills in Norwegian, equivalent to level B2 (the so-called Bergen test).

In order to streamline the processing of applicants, a computer program will be used to filter out the qualified applicants, as well as applicants who are qualified based on given conditions. Both those qualified and those who are conditionally qualified will then be called for an interview.

- Applicants who meet all three requirements are qualified.
- Applicants who cannot document pedagogical competence at the university level, but who meet the other two requirements, are qualified provided that they pass a course in university pedagogy within two years.
- Applicants who cannot document sufficient Norwegian skills, but who meet the other two requirements, are qualified provided that they pass the Bergen test (level B2) within two years.
- Applicants who do not fall under any of the points above are not qualified and should be rejected immediately.

In connection with the development and testing of the computer program that will process the applications automatically, you are to use decision tables to select use cases to be run during the system testing of the program.

a) Draw a decision table that shows all possible combinations of requirement fulfillment in the application. The decision table should include an action part that shows the outcome of the application, based on the points above.

Svar:

Betingelser:

PhD: The applicant must have a PhD within the relevant subject area.

Ped: The applicant must be able to document pedagogical competence at university level.

B2: The applicant must have language skills in Norwegian, equivalent to level B2 (the so-called Bergen test).

Aksjoner:

Q: Applicants who meet all three requirements are qualified.

Q÷Ped: Applicants who cannot document pedagogical competence at the university level, but who meet the other two requirements, are qualified provided that they pass a course in university pedagogy within two years.

Q÷B2: Applicants who cannot document sufficient Norwegian skills, but who meet the other two requirements, are qualified provided that they pass the Bergen test (level B2) within two years.

Not Q: Applicants who do not fall under any of the points above are not qualified and should be rejected immediately.

	R1	R2	R3	R4	R5	R6	R7	R8
PhD	true	true	true	true	false	false	false	false
Ped	true	true	false	false	true	true	false	false
B2	true	false	true	false	true	false	true	false
Aksjoner	Q	Q÷B2	Q÷Ped	Not Q	Not Q	Not Q	Not Q	Not Q

b) Reduce the number of rules in the decision table as much as possible without losing information. Explain how you made the reduction.

Svar:

Regel 5, 6, 7 og 8 kan slås sammen da alle søkere som ikke har PhD fra relevant fagområde uansett ikke er kvalifisert, uavhengig av om de har Ped og/eller B2.

	R1	R2	R3	R4	R5 - 8
PhD	true	true	true	true	false
Ped	true	true	false	false	-
B2	true	false	true	false	-
Aksjoner	Q	Q÷B2	Q÷Ped	Not Q	Not Q

c) Consider the following three different use cases:

- Applicant with a PhD within the relevant subject area, documented pedagogical competence, and sufficient language skills.
- Applicant with a PhD within the relevant subject area, but without documented pedagogical competence, and who also lacks sufficient Norwegian skills.
- Applicant with documented pedagogical competence, and sufficient Norwegian skills, but with a PhD outside the relevant subject area.

Do we need more use cases to achieve full decision table coverage? If yes, specify the use case(s) and the rule(s) in the decision table they represent.

Svar:

Use casene dekker reglene 1, 4 og 5-8. Vi mangler imidlertid use caser for regel 2 og 3:

- Regel 2: Applicant with a PhD within the relevant subject area, documented pedagogical competence, but without sufficient language skills.
- Regel 3: Applicant with a PhD within the relevant subject area, and sufficient language skills, but without documented pedagogical competence.

d) It turns out that the number of applicants for the positions is very high. To reduce the number to be called in for an interview, the department is considering adding an additional requirement:

- Only applicants with a minimum of two years of teaching experience at the university level will be considered.

How many possible different combinations of requirements fulfillment do we get in the decision table now (*before* any rule reductions)?

Svar:

Med fire betingelser får vi $2^4 = 16$ ulike kombinasjoner.

e) Decision tables are considered a test design tool. However, they are suitable for more than designing use cases. Where in the software life cycle are decision tables a good tool to ensure the foundation on which the program is to be built, and what is this form of testing called?

Svar:

I planleggingsfasen av prosjektet er de veldig godt egnet til å granske **(rewiew)** beskrivelsen av programmets funksjonalitet i kravspesifikasjonen, og derved egnet til å belyse alle mulige konsekvenser av ulike kombinasjoner av binære betingelser.

Denne formen for granskning (review) sorterer inn under **statisk testing**.

Part 3 - State transition

Note that in this task you shall answer both in the text editor and with sketches/drawings. Use the sketching paper handed to you in the exam room. See instructions on filling out the sketch sheets in the link below the task bar.

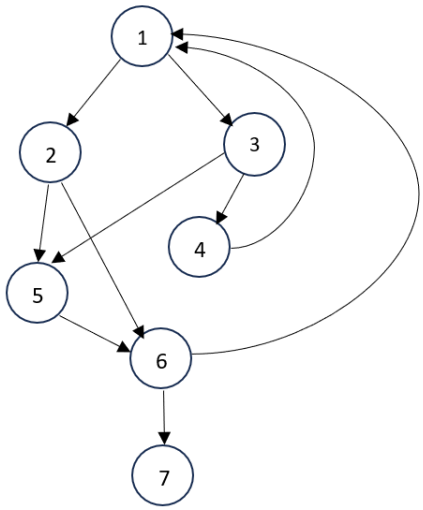
In the second compulsory assignment, you were testing av ticket machine.



The interaction between the user and the ticket machine module was described as follows.
(For practical reasons, the interaction is somewhat simplified in relation to the compulsory assignment.)

a) Draw a state transition diagram that cover all the states and transitions.

Svar:



b) Consider following use case:

" The traveler chooses to recharge an existing travel card. The system recognizes the card, and the traveler chooses to deposit 100 NOK, and then confirms the purchase."

Write down the path through the state transition diagram representing the use case.
What is the state coverage and transition coverage for this test case?

Svar:

S1→S3→S5→S6→S7

State coverage: 5/7 (Beregnes ut fra stien de har valgt

Transition coverage: 4/10 (Beregnes ut fra stien de har valgt=

c) Is there a path through the state transition diagram that provides 100% state coverage? If there are multiple paths of different lengths that all provide 100% state coverage, choose the shortest.

Svar:

S1→S3→S4→S1→S2→S5→S6→S7

d) What is the transition coverage of your path in point c).

Svar:

Transition coverage: 7/10

e) How many use cases do we need to achieve 100% transition coverage?
Write down the relevant path(s) through the state transition diagram that represents the use case(s).

Svar:

Vi trenger kun én:

S1→S3→S4→S1→S3→S5→S6→S1→S2→S6→S1→S2→S5→S6→S7

f) If a use case has 100% transition coverage, is it then possible to determine what the state coverage is? Justify your answer.

Svar:

Hvis vi har 100% transition coverage har vi også 100% state coverage.

100% transition coverage garanterer 100% state coverage, men ikke omvendt!

Part 3 - Short question

One of the seven testing principles is called *Defect clustering*.
What does this principle tell us, and how should it affect the testing activities?

Svar: (de trenger ikke å svare alt dette, kun de viktigste poengene)
Programmeringsfeilene er **ikke spredt jevnt utover koden**. De har en tendens til i å samle seg klynger. Undersøkelser har vist at omtrent **80 % av feilene befinner seg i ca. 20 % av koden.**
Hva årsaken til dette er, kan vi bare spekulere i, men det vi vet, er at gitte moduler er utviklet under **samme betingelser**. De kan være utviklet under samme tidspress, samme kompetanse eller mangel på kompetanse, samme motivasjon og mer. Det faktum at enkelte moduler er mer kompliserte enn andre, spiller også inn.
Visstheten om at feilene har en tendens til å samle seg i klynger, kan hjelpe oss med å fokusere testinnsatsen på de modulene der vi allerede har oppdaget feil.
Det er som med rotter: Har vi først sett én, er det gjerne flere av dem. Feilene har også i seg selv en smittende effekt. **Én enkelt feil kan lede til nye feil**, som igjen leder til nye, slik at feilene forplanter seg som rotter utover i koden.
Et statisk analyseverktøy, som analyserer koden uten å kjøre den, vil kunne være et **nyttig hjelpemiddel** i jakten på områder med dårlig kodekvalitet og/eller unødig høy kompleksitet. Slike områder bør vi **gå nærmere etter i sømmene**.

Part 3 - Short question

What other testing principles do we have, according to ISTQB?

Svar (nummerering og rekkefølgen er ikke så viktig. Ett poeng pr. prinsipp)

Prinsipp 1. Testing kan ikke bevise at koden er feilfri
Prinsipp 2. Fullstendig testing er umulig
Prinsipp 3. Tidlig testing sparer tid og penger
Prinsipp 5. Pass på sprøytemiddel-paradokset
Prinsipp 6. Testing er avhengig av konteksten
Prinsipp 7. Feilslutning vedrørende fravær av feil

Part 3 - Short questions

One of the guest lectures dealt with TDD.

a) What does the acronym TDD stand for?
b) What is the main principle behind TDD?
c) TDD is an iterative process.
Name and explain the various phases in the process.
d) TDD is preferably used at one of the four test levels. Which?

Svar:

a) Test Driven Development
b) Testen skrives før koden utvikles/implementeres
c)
   1. Red – testen skrives først og kjøres *før* koden er implementert
   2. Green – koden utvikles frem til testen passeres, dvs. blir grønn
   3. Refactor – koden omstruktureres og forbedres til den er bra nok (testes etter hver endring for å sikre at den fremdeles fungerer)

   Fasene går i loop: Red → Green → Refactor → Red → Green→ Refactor →…

d) Component test nivå (unit testing)

Part 3 - Short question

a) Explain the following terms:

- Regression testing
- Confirmation testing

b) What is the name of the test type that both regression testing and confirmation testing belong to?

Svar:

a)

Bekreftelsestesting er testingen vi foretar etter at en feil er rettet. Målet er å bekrefte
at feilen faktisk er fjernet, og at den nye versjonen av programmet fungerer som den skal. Det er avgjørende at bekreftelsestesten utføres på nøyaktig samme måte og under samme forhold som testen som i utgangspunktet avslørte feilen. (2)

Regresjonstesting utføres etter enhver form for endring av koden, enten den skyldes feilretting, refaktorisering eller videreutvikling. Her testes ikke bare den delen av koden der endringen har funnet sted, men også de deler av programmet som tidligere har bestått testene, ja eventuelt hele programmet om nødvendig. Målet er å sikre oss mot at nye feil utilsiktet har blitt introdusert, eller at endringen har gitt uønskede bivirkninger.

b)

Testetypen som omfatter Regression testing Confirmation testing er en av de fire testtypene, jf. ISTQB, og heter "**Testing related to changes**" eller på norsk: Testing relatert til endringer. (1)

Part 3 - Short question

a) What two factors are involved in the calculation of risk?

Svar:

a)

- Sannsynlighet for at feil inntreffer
- Konsekvensen av feilen

b) How are risk values used...

1. ... in the planning phase?
2. ... during the dynamic testing?

b)

1. I planleggingsfasen brukes mål for risiko, f.eks. i form av en risikoanalyse, til å avgjøre hvor **mye testing som er nødvendig** og **hvilke områder som bør prioriteres.**
2. Når en feil avdekkes under den dynamiske testingen, tilordnes gjerne en risikofaktor for å **angi alvorlighetsgraden av feilen**.

Part 4 - Essay

Within Continuous Integration, Delivery & Deployment it is common for testers and developers to work side by side in the same team (cf. the guest lecture on 24 April)

Discuss advantages and disadvantages of this practice from different perspectives, such as

- Test independence
- Test levels
- Roles
  - developer
  - tester
  - test leader
- Psychological aspects
- Different mindsets
- ...

Svar:

Du ulike punktene bør drøftes opp mot arbeidssituasjonen i et team:

- **Test independence:** Bør trekke frem **fordeler** og **ulemper** knyttet til graden av uavhengighet i testingen, innen teamet.

- **Test levels:** På hvilke testnivåer er det en fordel å jobbe side om side, og på hvilket er det ikke?

- **Roles:** Beskrive/drøfte hvordan det å jobbe i team påvirker tre ulike rollene: (3)
    - developer
    - tester
    - test leader
    -
- **Psychological aspects:** Hvordan påvirker det å jobbe i team det psykiske miljøet? F.eks: Blir det lettere eller vanskeligere å formidle positiv/negativ kritikk? Hva med kommunikasjonen? Blir den bedre?

- **Different mindsets:** Utvikler er optimist, tester profesjonell pessimist. Hvordan påvirker det samarbeidet og teamet?

- **Bonus** for å ta med andre aspekter

Se kapittel 5.1 Test organization

https://www.uio.no/studier/emner/matnat/ifi/IN3240/v24/forelesningsvideoer/chapter-5-part-1-24.pdf

Aktuelle slides:

Independence in testing

- A certain degree of independence is often more effective at finding defects and failures. However, the developer can very efficiently find bugs in their own code.

- The level of independence in the testing depends on the objective of testing.

Independence test levels

- Independence levels:

- Tests designed by the same person who wrote the code
- Tests designed by another person from the same team, but same organization
- Tests designed by a person from a separate testing team , but in the same organization
- Tests designed by a person from an outside organization / company (outsourcing the testing)

Advantages & disadvantages

+

Independent testers see other and different defects, and are unbiased.

An independent tester can verify assumptions people made during specification and implementation of the system.

Advantages & disadvantages

-

Isolation from the development team (if treated as totally independent).

Independent testers may be the bottleneck as the last checkpoint.

Developers may lose a sense of responsibility for quality.

Advantages & disadvantages

Note

Testing tasks may be done by people in a specific testing role, or may be done by someone in another role, such as:

- project manager
- quality manager
- developer
- business and domain expert
- infrastructure or IT operations

(this can be both good and bad)

Tasks of the test leader and tester

There is wide variation in the roles that people within the test team play. The two most common roles are

- **test leader**
- **tester**

Test leader = test manager / test coordinator.
The test leader plans, monitors and controls the *testing activities and tasks*.

The tester
The tester reviews and contributes to test plan, analyses, designs, prepares, implements and executes tests.

A good tester needs:

- Curiosity
- Professional pessimism
- Attention to details
- Good communication skills
- Experience at error guessing

- To communicate defects and failures in a constructive way: fact-focused reports and review of findings

Tips and tricks

- Be clear and objective
- Confirm that:
  - You have understood the requirements
  - The person who is going to fix the bug has understood the problem

1. Test organization
- 1.1 Test organization and independence
- 1.2 Tasks of the test leader and tester

2. Test planning and estimation
- 2.1 Test planning
- 2.2 Test planning activities
- 2.3 Entry criteria
- 2.4 Exit criteria
- 2.5 Test estimation
- 2.6 Test strategy, test approach

3. Test progress monitoring and control
- 3.1 Test progress monitoring
- 3.2 Test reporting
- 3.3 Test control

4. Configuration management

5. Risk and testing
- 5.1 Project risks
- 5.2 Product risks

6. Incident management

Skills

People involved in testing need

- **Application or business domain**: A tester must understand the intended behavior and the problem the system will solve in order to spot improper.
- **Technology**: A tester must be aware of issues, limitations and capabilities of the chosen implementation technology, in order to locate problems and the 'likely to fail' functions and features.
- **Testing:** A tester must know the testing topics discussed in this book - in order to carry out the test tasks assigned.

Skills

People involved in testing

need basic professional and social qualifications such as

- literacy
- the ability to prepare and deliver written and verbal reports
- the ability to communicate effectively
- contribute to a positive relationship between developer and tester