

IN3240 - Oblig 2

29. April 2019

Gruppemedlemmer:

Nadya Gileva (nadezdag)

Johan Hoi (jchoi)

Marius Haugen (marius5)

Emil H. Unhjem (emilhun)

Excercise 1

Se vedlegg

Excercise 2

2.1

I + II) equivalence partitions, både gyldige og ugyldige + boundary values

I dette programmet kan vi bruke equivalence partitions til å sjekke om både gyldige og ugyldige datoer blir behandlet riktig av programmet. En equivalence partitioning for dette programmet kunne for eksempel sett slik ut:

Invalid partition	Valid Januar	Valid Februar	Valid Mars	Valid (osv.)	Invalid partition
Måned: 0 Dag: 0	Måned: 1 1 - 31	Måned: 2 1 - 29	Måned: 3 1 - 31	Måned: x 1 - xx	Måned: 13 Dag: 32

Her tester vi alle gyldige og ugyldige datoer i programmet. Ugyldige datoer er naturligvis datoer som 0/0, 0/1, 1/0, 32/13, 32/1, 1/13, eller minustall som -8/-3. Dersom slike datoer blir skrevet inn bør programmet reagere på det. De gyldige datoene bør også

testes grundig ettersom at ikke alle måneder har like mange dager. Februar kommer aldri opp i 30 eller 31 dager, for eksempel.

Her vil programmets boundary values være avhengig av hvilken måned det er snakk om (f.eks. har ikke Februar en 30'de dag), men felles for alle måneder er at ingen av dem har en 0'te dag eller 32're dag. Det finnes heller ikke en 0'te eller 13'de måned. Når det kommer til gyldige verdier så vet vi at månedene 1 – 12 er gyldige og vi vet at alle måneder har minst 1 - 29 dager (dersom skuddår regnes med). En tabell over boundary values kunne sett slik ut:

Invalid	Valid	Invalid
Måned: 0 Dag: 0	Måned: 1 – 12 Dag: 1 – 29 (30, 31)	Måned: 13 Dag: 32

2.2

- Trykker bruker på «see your horoscope» uten å ha skrevet inn noe (eller så lengde det mangler info i et av feltene/feil info i feltene), kommer det ikke en informativ feilmelding (minor)
- Det er ikke lov å være født 21. desember (21/12) av en eller annen grunn (major)
- Det er lov å være født 31. juni, men det er ikke en ekte dag. (major)
- Med tanke på at vi har funnet to feil som innebærer feil ved datoer kan det tenkes at det finnes flere av disse. Det kan derfor være relevant under fremtidig testing å se mer på datoer i programmet.

Det viktigste er å fikse de feilene som gjør at programmet ikke fungerer som det skal. Vi ville derfor prioritert å fikse feilene hvor det er lov å være født på datoer som ikke eksisterer og de hvor lovlige fødselsdager ikke blir akseptert som en ekte fødselsdag, feil som vi ville definert som major defekter. Grensesnittfeil (som dårlige tilbakemeldinger ved feil input) tenker vi kan lett fikses og det haster ikke å fikse slike feil med en gang. Feil ved grensesnittet påvirker heller ikke hvordan programmet

fungerer, nødvendigvis, og slike feil ville vi definert som minor. Nummerert ville vi fikset feilene slik:

1. Feilen hvor det ikke er lov å være født 21. Desember (major)
2. Feilen hvor det er lov å være født 31. Juni, en dato som ikke eksisterer (major)
3. Andre uoppdagede feil ved datoer (major)
4. Grensesnittfeilen som gir dårlige tilbakemeldinger (minor)

Oppgave 2.3

IEEE 829 STANDARD: TEST INCIDENT REPORT

Test incident report identifier: Tid: 14.06 - 14.30, Dato: 23/04/19, black-box testing

Oppsummering (Summary): Mens vi black-box testet Horoscope19-programmet med noen inputs oppdaget vi 3 defekter av varierende alvorlighetsgrad. Vi ville teste hva som ville skje dersom vi skrev ingenting/andre symboler enn tall inn i skrivefeltene, om ikke-eksisterende datoer ble akseptert og om det fantes eksisterende datoer som ble avvist av programmet. Vi oppdaget defekter i alle områdene vi ville undersøke nærmere. Defekter oppsummert:

- Når ingenting/andre enn symboler enn tall ble skrevet inn så produserte ikke programmet en informativ feilmelding (minor).
- Når den ikke-eksisterende datoen 31. Juni (31/12) ble skrevet inn så aksepterte programmet datoen som en ekte dato og produserte ikke en feilmelding (major).
- Når datoen 21. Desember (21/12) ble skrevet inn så avviste programmet datoen og produserte en feilmelding for ugyldige datoer (major)

Hendelse 1 beskrivelse (Incident 1 description):

- Input: Ingenting/andre symboler enn tall
- Forventet resultat (Expected Results): Informativ feilmelding

- Faktisk resultat (Actual Results): Ingen feilmelding angående feilen
- Avvik (Anomalies): Ingen feilmelding
- Dato og Tid: 23/04/19, 14.06
- Gjentakelsesforsøk (Attempts to Repeat): 3
- Testere: Emil
- Observerere: Nadya, Marius, Johan

Hendelse 1 påvirkelse (Incident 1 impact): Denne defekten påvirker bare brukervennligheten og er ikke en spesielt alvorlig eller hastende feil som må fikses. Vi vil si at dette er en minor defekt og dens prioritetsnivå kan bli beskrevet som rundt deferred (kan forbli i programmet) eller delayed (systemet er brukbart, men defekten bør fikses). Vi anser det ikke som en immediate defekt (må bli fikset ASAP)

Hendelse 2 beskrivelse (Incident 2 description):

- Input: 31. Juni (31/06)
- Forventet resultat: Feilmelding angående ugyldig dato
- Faktisk resultat: Vanlig visning av horoskopet til krepsen
- Avvik (Anomalies): Ingen feilmelding
- Dato og Tid: 23/04/19, 14.17
- Gjentakelsesforsøk (Attempts to Repeat): 3
- Testere: Emil
- Observerere: Nadya, Marius, Johan

Hendelse 2 påvirkning (Incident 2 impact): Dette er en mer alvorlig defekt. Hvis programmet ikke kan avdekke falske datoer så kan det lede til flere alvorlige defekter. Vi har ikke sett noen alvorlige konsekvenser som følge av denne defekten, men vi mener likevel at den bør fikses umiddelbart. Vi definerte denne defekten som en major feil og mener at den bør hav prioritetsnivået immediate (bør fikses umiddelbart).

Hendelse 3 beskrivelse (Incident 3 description):

- Input: 21. Desember (21/12)
- Forventet resultat: Visning av horoskopet for skytten
- Faktisk resultat: Feilmelding angående ugyldige datoer
- Avvik (Anomalies): Feilmelding dukket opp på tross av at det ikke var en feil
- Dato and Tid: 23/04/19, 14.30
- Gjentakelsesforsøk (Attempts to Repeat): 3
- Testere: Emil
- Observerere: Nadya, Marius, Johan

Hendelse 3 påvirkning (Incident 3 impact): Enda en alvorlig feil. Programmet bør kjenne igjen alle gyldige fødselsdager. På tross av at det er en alvorlig feil, har vi ikke funnet noen alvorlige konsekvenser som følge av denne feilen. Vi definerer denne defekten som en major feil og vi mener at den bør få prioritetsnivået immediate (bør fikses ASAP)

Excercise 3

For å lage decision table må vi først definere kondisjoner, regler og antall mulige kombinasjoner. Vi kan se at vi har 3 kondisjoner:

1. Bil er en elektrisk bil
2. Bil er en petrol/diesel bil
3. Det er rush hour

Hver kondisjon kan være enten True eller False. Så antall kombinasjoner = $2*2*2 = 8$

Condition	R1	R2	R3	R4	R5	R6	R7	R8
Electric	F	F	F	F	T	T	T	T
Petrol/Diesel	F	F	T	T	F	F	T	T

Rush hour	F	T	F	T	F	T	F	T
Effects (outputs)								
No fee	X	X			Y		X	X
Fee 30kr						Y		
Fee 50kr			Y					
Fee 100kr				Y				

Forenklet tabell

Vi kan se nærmere på R1 og R2, R7 og R8. Selv om disse reglene har forskjellige true/false verdier i Rush hour, de bestemmer ikke resultat som blir samme. Så vi kan slå sammen R1 og R2, R7 og R8.

Condition	R1	R2	R3	R4	R5	R6
Electric	F	F	F	T	T	T
Petrol/Diesel	F	T	T	F	F	T
Rush hour	-	F	T	F	T	-
Effects (outputs)						
No fee	X			Y		X
Fee 30kr					Y	
Fee 50kr		Y				
Fee 100kr			Y			

Excercise 4

4.1 Se vedlegg for denne oppgaven [Se diagram 4.3 i vedlegg]

4.2

i) Du måler state coverage ved å finne en sti gjennom alle tilstandene i programmet. Hvis du finner den korteste mulig veien gjennom alle tilstandene kan vi si at vi har 100% state-coverage

ii) For å måle transition-coverage må vi finne en sti fra startpunktet til endepunktet og telle opp antall overganger som ble gjort. Når vi har telt opp antall overganger kan vi regne ut transition-coverage ved følgende.

$$\frac{\text{Antall overganger i stien}}{\text{Antall mulige overganger}} \times 100\%$$

4.3

Det er en sammenheng mellom transition-coverage og state-coverage. Her vil det si at hvis du har testet for 100% transition-coverage har du også testet for 100% state-coverage, men ikke omvendt. Det gir mening fordi det er mulig å være innom alle tilstandene i programmet uten å ha brukt alle overgangene, men det er ikke mulig å ha brukt alle overgangene uten å ha brukt alle tilstandene. Derfor sier vi også at transition-coverage er “*sterkere*” enn state-coverage.

Som vi ser i diagrammet er det mange overganger med tanke på at man alltid har en mulighet til å gå tilbake, så derfor blir tilstandssekvensen for å dekke 100% transition coverage veldig lang. Følgende er et eksempel på en tilstandssekvens som dekker 100% transition-coverage fra diagrammet:

SS-S1-SS-S1-S2-S3-S6-S3-S7-S8-S10-S8-S11-ES-SS-S4-SS-S4-S5-SS-S4-S5-S3-S7-S9-S10-S9-S11-ES-SS-S4-S5-S3-S7-S9-SS

$$\frac{38}{13} \times 100\% \approx 292\% \text{ state coverage}$$

Denne lange sekvensen gir oss 292% state-coverage og gir oss mer enn 100% transition-coverage, ettersom vi må bruke flere overganger mer enn en gang. Hvis vi bruker formelen fra tidligere får vi en transition-coverage på

$$\frac{33}{27} \times 100\% \approx 122\% \text{ transition coverage}$$

Dette gjelder da selvfølgelig kun viss vi teller en state eller transition mer enn en gang.

For å vise forholdet mellom state-coverage og transition-coverage har vi lagt til et ekstra diagram i denne oppgaven som viser hvordan man kan oppnå 100% state coverage uten å ha 100% transition-coverage [Se diagram 4.4 i vedlegg]

Excercise 5

5.1

- Sjekker inn med bookingnr,
 - noe går galt / feil bookingnr
- Sjekke kredittkort
 - Kredittkort ikke godkjent

Et interessant aspekt ved programmet er at man alltid skal kunne avbryte (cancel), noe som kan være interessant å teste. Dette har vi testet i ett tilfelle (8a i kredittkorttabellen), men av praktiske hensyn så har vi ikke testet alle avbryt-tilfellene.

5.2

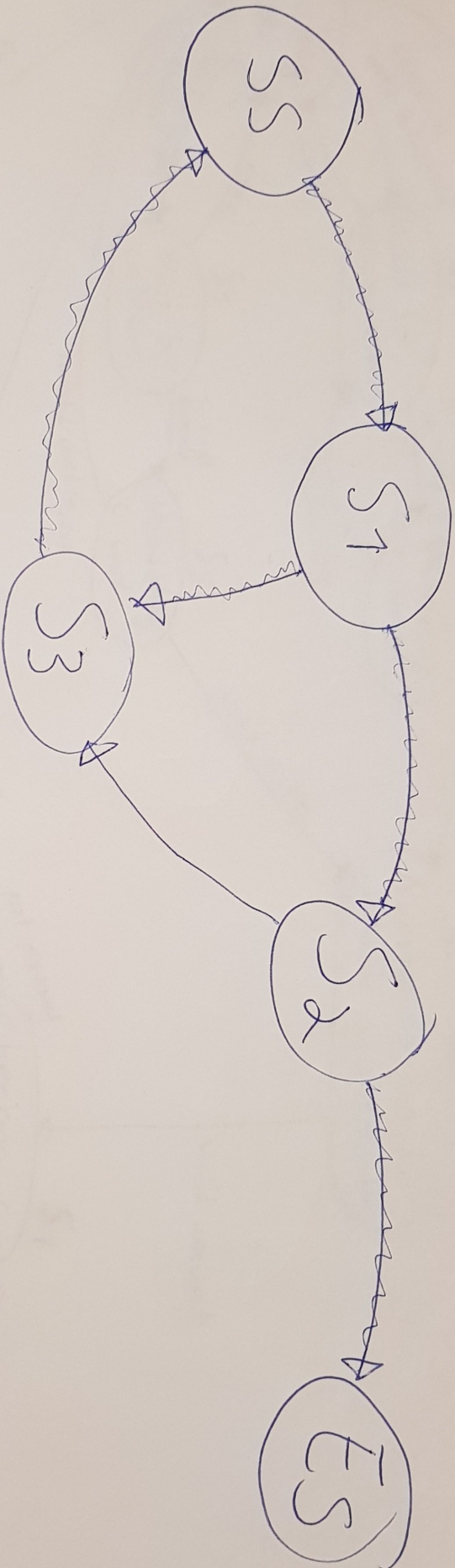
1. Testing av bookingnummer

Main Success Scenario A: Actor S: System	Step	Description
	0	S: Awaits userinput
	1	A: Choose check-in with booking number
	2	S: Ask for booking number
	3	A: Enters booking number
	4	S: Validates booking number
	5	S: Prompt for amount of luggage items
	6	A: Enters number of items and clicks continue
	7	S: Checks for reserved seat
	8	A: Confirm seat change/reservation
	9	S: Print luggage tags and boarding pass
	10	A: Retrieves luggage tag and boarding pass
	11	S: Wishes customer a happy flight
Extensions	4a	Invalid entry S: Display message and return to step 2
	6a	Invalid number of items S: Display error message and ask for re-try
	8a	Customer declines seat reservation/change S: System arbitrarily chooses seat and proceed

2. Testing av kredittkort

Main scenario A: Actor S: System	Step	Description
	0	S: Awaits userinput
	1	A: Choose check-in with credit card
	2	S: Ask for credit card
	3	A: Enters credit card
	4	S: Validates credit card
	5	S: Prompt for amount of luggage items
	6	A: Enters number of items and clicks continue
	7	S: Checks for reserved seat
	8	A: Decline seat reservation
	9	S: Print luggage tags and boarding pass
	10	A: Retrieves luggage tag and boarding pass
	11	S: Wishes customer a happy flight
Extensions	4a	Invalid entry S: Display message, eject card and return to step 2
	6a	Invalid number of items S: Display error message and ask for re-try
	8a	Customer accept seat reservation, but cancel during selection S: System cancel luggage check-in and returns to step 0

Diagram 4.4



$SS - S_1 - S_3 - SS - S_2 - ES$

Diagram 4.3

