Task 1.1

Predicates and reachability

Line	Predicate	Reachability
14	P1=AvBv(C^D)	TRUE
17	P2=E^F^G	P1=FALSE
20	P3=Hvl	P1=FALSE ^ P2=FALSE
24	P4=J^K	P1=FALSE ^ P2=FALSE

P1:

Predicate coverage= 2TRs

- P1=True
- P1=False

Clause coverage: 8 TRs

The following result for CACC is based on the truth table on the right:

Major Clause	Set of possible tests
A	(6,14), (6,15), (6,16), (7,14), (7,15), (7,16), (8,14), (8,15), (8,16)
B	(10,14), (10,15), (10,16), (11,14), (11,15), (11,16), (12,14), (12,15), (12,16)
C	(13,15)
D	(13,14)

Truth Table:

Row#	A	B	C	D	P	PA	PB	PC	PD
1	T	T	T	T	T				
2	T	T	T		T				
3	T	T		T	T				
4	T	T			T				
5	T		T	T	T				
6	T		T		T	T			
7	T			T	T	T			
8	T				T	T			
9		T	T	T	T				
10		T	T		T		T		
11		T		T	T		T		
12		T			T		T		
13			T	T	T			T	T
14			T			T	T		T
15				T		T	T	T	
16						T	T		

A= major

Test number	Test	Input	Expected output	PC	CC	CACC
6	T,F,T,T	15,q,false	-1	½ (T)	4/8	½ (T)
14	F,F,T,F	26,m,false	500	½ (F)	2/8	½ (F)
11	F,T,F,T	96,m,false	-1	-	2/8	-
Tot:				2/2	8/8	2/2

Hektor Dahlberg 41685 Software Testing Assignment 2

P2:

Predicate coverage= 2TRs

- P2=True
- P2=False

Clause coverage: 6 TRs

The following result for CACC is based on the truth table on the right:

Major Clause	Set of possible tests
E	(1,5)
F	(1,3)
G	(1,2)

Truth Table:

Row#	E	F	G	P	PE	PF	PG
1	T	T	T	T	T	T	T
2	T	T					T
3	T		T			T	
4	T						
5		T	T		T		
6		T					
7			T				
8							

E= Major

Test Number	Test	Input	Expected output	PC	CC	CACC
1	T,T,T	24,m,False	2000	½	3/6	½
5	F,T,T	26,m,False	500	½	1/6	½
4	T,F,F	24,f,True	300	-	2/6	-

Hektor Dahlberg 41685 Software Testing Assignment 2

P3:

Predicate coverage= 2TRs

- P3=True
- P3=False

Clause coverage: 4 TRs

The following result for CACC is based on the truth table on the right:

Major Clause	Set of possible tests
H	(2,4)
I	(3,4)

Truth Table:

Row#	H	I	P	PH	PI
1	T	T	T		
2	T		T	T	
3		T	T		T
4				T	T

Test Number	Test	Input	Expected output	PC	CC	CACC
1	T,T	24,f,True	300	½	2/4	½
4	F,F	26,m,False	500	½	4/4	-
2	T,F	26,m,True	300	-	-	½

Hektor Dahlberg 41685 Software Testing Assignment 2

P4:

Predicate coverage= 2TRs

- P4=True
- P4=False

Clause coverage: 4 TRs

The following result for CACC is based on the truth table on the right:

Major Clause	Set of possible tests
J	(1,3)
K	(1,2)

Truth Table:

Row#	J	K	P	PJ	PK
1	T	T	T	T	T
2	T				T
3		T		T	
4					

Test Number	Test	Input	Expected output	PC	CC	CACC
1	T,T	47,m,True	200	1/2	2/4	½
3	F,T	44,m,True	300	1/2	1/4	1/2
2	T,F	66,m,True	300	-	1/4	-

Task 1.2

Test number	Test	Input	Expected output	PC	CC	CACC
6	T,F,T,T	15,q,false	-1	½ (T)	4/8	½ (T)
14	F,F,T,F	26,m,false	500	½ (F)	2/8	½ (F)
11	F,T,F,T	96,m,false	-1	-	2/8	-
Tot:				2/2	8/8	2/2
Test Number	Test	Input	Expected output	PC	CC	CACC
1	T,T,T	24,m,False	2000	½	3/6	½
5	F,T,T	26,m,False	500	½	1/6	½
4	T,F,F	24,f,True	300	-	2/6	-
Test Number	Test	Input	Expected output	PC	CC	CACC
1	T,T	24,f,True	300	½	2/4	½
4	F,F	26,m,False	500	½	4/4	-
2	T,F	26,m,True	300	-	-	½
Test Number	Test	Input	Expected output	PC	CC	CACC
1	T,T	47,m,True	200	1/2	2/4	½
3	F,T	44,m,True	300	1/2	1/4	1/2
2	T,F	66,m,True	300	-	1/4	-

Task 1.3

FINAL TESTS:

Test number	Input	Output
1	15,q,false	-1
2	26,m,false	500
3	96,m,false	-1
4	24,m,false	2000
5	24,f,true	300
6	26,m,true	500
7	47,m,True	200
8	44,m,True	300
9	66,m,True	300

Task 2.1

Interface based	1	2	3	4	5	6
AGE:	x<18	x>95	18-24	25-45	46-64	65-95
SEX	Not m,f	m	f			
MARRIED	true	false				

Task 2.2

PAIR-WISE:

		Test number	FINAL TESTS
[a1, s1]	[s1, m1]	1	[a1, s1, m1]
[a1, s2]	[s1, m2]	2	[a1, s2, m1]
[a1, s3]	[s2, m1]	3	[a1, s3, m1]
[a2, s1]	[s2, m2]	4	[a2, s1, m2]
[a2, s2]	[s3, m1]	5	[a2, s2, m2]
[a2, s3]	[s3, m2]	6	[a2, s3, m2]
[a3, s1]		7	[a3, s1, m1]
[a3, s2]		8	[a3, s2, m2]
[a3, s3]		9	[a3, s3, m1]
[a4, s1]		10	[a4, s1, m2]
[a4, s2]		11	[a4, s2, m1]
[a4, s3]		12	[a4, s3, m2]
[a5, s1]		13	[a5, s1, m1]
[a5, s2]		14	[a5, s2, m2]
[a5, s3]		15	[a5, s3, m1]
[a6, s1]		16	[a6, s1, m2]
[a6, s2]		17	[a6, s2, m1]
[a6, s3]		18	[a6, s3, m2]

Base choice: a4, s2, m1

Test number	Test
11	[a4, s2, m1]
19	[a4, s2, m2]
20	[a4, s1, m1]
21	[a4, s3, m1]
2	[a1, s2, m1]
22	[a2, s2, m1]
23	[a3, s2, m1]
24	[a5, s2, m1]
17	[a6, s2, m1]

Each choice:

Test number	Input	Each choice
1	a1, s1, m1	3/11
5	a2, s2, m2	6/11
9	a3, s3, m1	8/11
10	a4, s1, m2	9/11
24	a5, s2, m1	10/11
18	a6, s3, m2	11/11

Task 2.3

Functionality based approach:

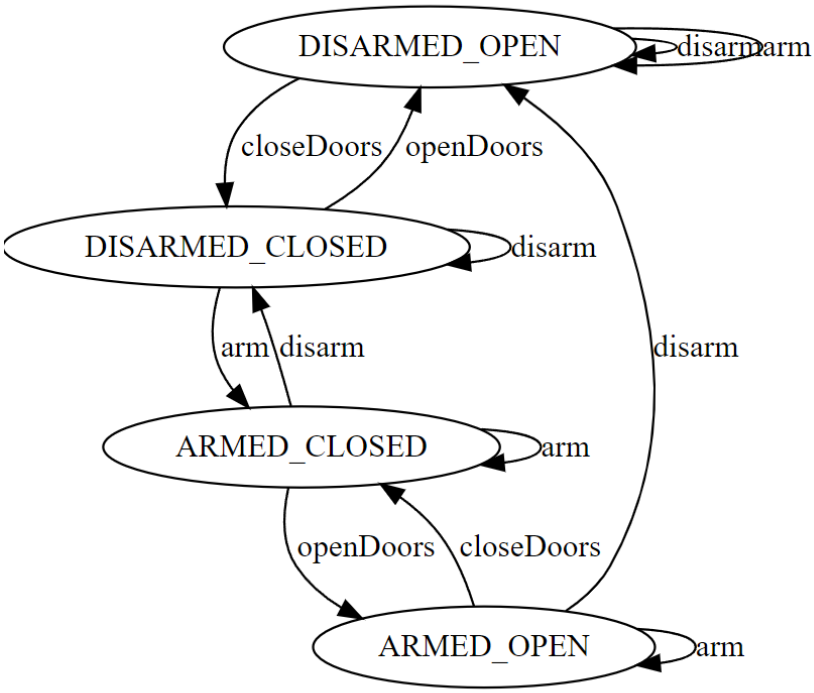
	True	False	Test number true:	Test True:	Test number false:	Test False:
Underaged	True	False	1 (3)	17, f, true	7 (9)	21, f, true
Teenage	True	False	2 (8)	21, m, false	8 (14)	45, m, false
Families with children	True	False	3 (21)	35, f, true	9 (25)	20, f, false
Unmarried middle aged	True	False	4 (19)	35, m, false	10 (15)	60, f, true
retired	True	False	5 (17)	70 ,m, true	11 (11)	25, m, true
Too old to drive	True	False	6 (5)	999, m, false	12 (2)	16, m, true

Task 2.4

Final tests:

number		Input	Expected output	Each choice	Pair wise	Base choice	Functionality based
1	[a1, s1, m1]	17, q, true	-1	3/11	1/18	-	-
2	[a1, s2, m1]	17, m, true	-1	-	2/18	1/9	1/12
3	[a1, s3, m1]	17, f, true	-1	-	3/18	-	2/12
4	[a2, s1, m2]	999, q, false	-1	-	4/18	-	-
5	[a2, s2, m2]	999, m, false	-1	6/11	5/18	-	3/12
6	[a2, s3, m2]	999, f, false	-1	-	6/18	-	-
7	[a3, s1, m1]	21, q, true	-1	-	7/18	-	-
8	[a3, s2, m2]	21, m, false	2000	-	8/18	-	4/12
9	[a3, s3, m1]	21, f, true	300	8/11	9/18	-	5/12
10	[a4, s1, m2]	35, q, false	-1	9/11	10/18	-	-
11	[a4, s2, m1]	35, m, true	300	-	11/18	2/9	6/12
12	[a4, s3, m2]	35, f, false	300	-	12/18	-	-
13	[a5, s1, m1]	60, q, true	-1	-	13/18	-	-
14	[a5, s2, m2]	60, m, false	400	-	14/18	-	7/12
15	[a5, s3, m1]	60, f, true	200	-	15/18	-	8/12
16	[a6, s1, m2]	70, q, false	-1	-	16/18	-	-
17	[a6, s2, m1]	70, m, true	300	-	17/18	3/9	9/12
18	[a6, s3, m2]	70, f, false	300	10/11	18/18	-	-
19	[a4, s2, m2]	35, m, false	500	-	-	4/9	10/12
20	[a4, s1, m1]	35, q, true	-1	-	-	5/9	-
21	[a4, s3, m1]	35, f, true	300	-	-	6/9	11/12
22	[a2, s2, m1]	999, m, true	-1	-	-	7/9	-
23	[a3, s2, m1]	21, m, true	300	-	-	8/9	-
24	[a5, s2, m1]	60, m, true	200	11/11	-	9/9	-
25	[a3, s3, m2]	21, f, false	300	-	-	-	12/12

Task 3.2



Task 3.3

Random/NoSteps	2	16	32	64
Action coverage	2/4	4/4	4/4	4/4
State Coverage	2/4	4/4	4/4	4/4
Transition Coverage	2/13	13/13	13/13	13/13
Edge pair coverage	1/43	25/43	29/43	36/43

Greedy/NoSteps	2	16	32	64
Action coverage	2/4	4/4	4/4	4/4
State Coverage	1/4	4/4	4/4	4/4
Transition Coverage	2/13	13/13	13/13	13/13
Edge pair coverage	1/43	25/43	33/43	38/43

All Round/NoSteps	2	16	32	64
Action coverage	2/4	4/4	4/4	4/4
State Coverage	2/4	0/4	0/4	4/4
Transition Coverage	2/13	13/13	13/13	13/13
Edge pair coverage	0/43	26/43	31/43	36/43

Couldn’t find any amount steps that would give full edge pair coverage.

TASK 3.6

ii:

In AllRoundTester and randomTester both failed due to DISARMED_CLOSED during arm returned 1, which it should do according to the state machine from the task while it returned 0.

In greedyTester it failed due to DISARMED_OPEN during closeDoors returned 0, which it should do according to the state machine from the task while it returned -1.

AllRoundTester:

done Random reset(true)

done (DISARMED_OPEN, disarm, DISARMED_OPEN)

done (DISARMED_OPEN, arm, DISARMED_OPEN)

done (DISARMED_OPEN, arm, DISARMED_OPEN)

done (DISARMED_OPEN, disarm, DISARMED_OPEN)

done (DISARMED_OPEN, arm, DISARMED_OPEN)

done (DISARMED_OPEN, arm, DISARMED_OPEN)

done (DISARMED_OPEN, closeDoors, ARMED_CLOSED)

done (ARMED_CLOSED, arm, ARMED_CLOSED)

done (ARMED_CLOSED, openDoors, ARMED_OPEN)

done (ARMED_OPEN, closeDoors, ARMED_CLOSED)

done (ARMED_CLOSED, disarm, DISARMED_CLOSED)

done (DISARMED_CLOSED, openDoors, DISARMED_OPEN)

done (DISARMED_OPEN, disarm, DISARMED_OPEN)

done (DISARMED_OPEN, closeDoors, DISARMED_CLOSED)

FAILURE: failure in action arm from state DISARMED_CLOSED due to java.lang.AssertionError: expected:<1> but was:<0>

randomTester:

done Random reset(true)

done (DISARMED_OPEN, closeDoors, DISARMED_CLOSED)

done (DISARMED_CLOSED, openDoors, DISARMED_OPEN)

done (DISARMED_OPEN, closeDoors, DISARMED_CLOSED)

done (DISARMED_CLOSED, disarm, DISARMED_CLOSED)

done (DISARMED_CLOSED, disarm, DISARMED_CLOSED)

FAILURE: failure in action arm from state DISARMED_CLOSED due to java.lang.AssertionError: expected:<1> but was:<0>

greedyTester:

done (DISARMED_OPEN, arm, DISARMED_OPEN)

done (DISARMED_OPEN, closeDoors, ARMED_CLOSED)

done (ARMED_CLOSED, openDoors, ARMED_OPEN)

done (ARMED_OPEN, arm, ARMED_OPEN)

done (ARMED_OPEN, disarm, DISARMED_OPEN)

done (DISARMED_OPEN, disarm, DISARMED_OPEN)

done (DISARMED_OPEN, disarm, DISARMED_OPEN)

FAILURE: failure in action closeDoors from state DISARMED_OPEN due to java.lang.AssertionError: expected:<0> but was:<-1>

Task 3.7

a. Which was able to achieve better coverage faster for abstract tests

   Greedy was the fastest.

b. Which was able to detect the error faster for the concrete tests

   Greedy tester found an error fastest.

c. Give a possible explanation for the above.

   Random could have found first as it randomized, but greedy found it faster, why it was faster then all round tester I'm not sure about.