



به نام خدا
تمرین کامپیوتری شماره 4



810198537

ملیکا حیدری دستجردی

دانشکده مهندسی برق و

810199461

علی عطاءاللهی

کامپیوتر

بخش اول - پیاده سازی تست

Github repository link: <https://github.com/Melikad/SoftwareTesting-CA4>

بخش دوم - گزارش کار

سوال (1)

بخش اول:

Clauses: a, b, c

بخش دوم:

row	a	b	c	$(a \wedge \sim b)$	$(\sim a \wedge \sim c)$	$(a \wedge \sim c)$	$(a \wedge \sim b) \vee (\sim a \wedge \sim c) \vee (a \wedge \sim a)$
1	T	T	T	F	F	F	F
2	T	T	F	F	F	T	T
3	T	F	T	T	F	F	T
4	T	F	F	T	F	T	T

5	F	T	T	F	F	F	F
6	F	T	F	F	T	F	T
7	F	F	T	F	F	F	F
8	F	F	F	F	T	F	T

بخش سوم:

Major Clause Condition:

$$P_x = P_{x=true} \oplus P_{x=false}$$

★ GACC:

Major Clause = a: rows: 3, 7 {(3, 7)}	Major Clause = b: rows: 1, 3 {(1, 3)}	Major Clause = c: rows: 1, 2, 5, 6, 7, 8 {(1, 2), (1, 6), (1, 8), (5, 2), (5, 6), (5, 8), (7, 2), (7, 6), (7, 8)}
---	---	---

★ RACC:

Major Clause = a: rows: 3, 7 {(3, 7)}	Major Clause = b: rows: 1, 3 {(1, 3)}	Major Clause = c : rows: 1, 2, 5, 6, 7, 8 {(1, 2), (5, 6), (7, 8)}
---	---	--

★ RICC:

Major Clause = a: rows: 1, 2, 4, 5, 6, 8 {(1, 5), (2, 6), (4, 8)}	Major Clause = b: rows: 2, 4, 5, 6, 7, 8 {(2, 4), (5, 7), (6, 8)}	Major Clause = c : rows: 3, 4 {(3, 4)}
---	---	--

سوال دوم:

برای هر یک از سه پارامتر، حالت‌های تعیین‌کننده را طبق جداول زیر جدا می‌کنیم.

quantity:

0	1	2	3	
Less than 0	Equal to 0	Equal to 1	Greater than 1	Q

-> از لحاظ منطقی حالت‌های کمتر و مساوی صفر قابل قبول نیستند و توجیه ندارند. (Q_0, Q_1)

unit_price:

0	1	2	
Less than 0	Equal to 0	Greater than 0	U

-> از نظر منطقی قیمت صفر و کمتر از آن قابل قبول نیست. (U_0, U_1)

discount:

0	1	2	3	4	
Less than 0	Equal to 0	Between 0 and 1	Equal to 1	Greater than 1	D

-> درصد تخفیف نباید کمتر از صفر باشد. (D_0)

حال چند تست طبق روش Pair-wise coverage ایجاد می‌کنیم:

1. $\sim Q_0, U_0, D_0$
2. $\sim Q_0, U_1, D_1$
3. $\sim Q_0, U_2, D_2$
4. $\sim Q_0, U_0, D_3$
5. $\sim Q_0, U_1, D_4$
6. $\sim Q_1, U_0, D_0$
7. $\sim Q_1, U_1, D_1$
8. $\sim Q_1, U_2, D_2$

9. ~ Q1, U0, D3
10. ~ Q1, U1, D4
11. ~ Q2, U0, D0
12. ~ Q2, U1, D1
13. ~ Q2, U2, D2 -> ورودی معتبر
14. ~ Q2, U0, D3
15. ~ Q2, U1, D4
16. ~ Q3, U0, D0
17. ~ Q3, U1, D1
18. ~ Q3, U2, D2 -> ورودی معتبر
19. ~ Q3, U0, D3
20. ~ Q3, U1, D4

به همین ترتیب، تست‌ها به شکل زیر خواهند بود:

1. assertEquals("Invalid input", calculate_total_cost(-1, -1, -1))
2. assertEquals("Invalid input", calculate_total_cost(-1, 0, 0))
3. assertEquals("Invalid input", calculate_total_cost(-1, 1, 0.77))
4. assertEquals("Invalid input", calculate_total_cost(-1, -1, 1))
5. assertEquals("Invalid input", calculate_total_cost(-1, 0, 1.7))
6. assertEquals("Invalid input", calculate_total_cost(0, -1, -1))
7. assertEquals("Invalid input", calculate_total_cost(0, 0, 0))
8. assertEquals("Invalid input", calculate_total_cost(0, 1, 0.77))
9. assertEquals("Invalid input", calculate_total_cost(0, -1, 1))
10. assertEquals("Invalid input", calculate_total_cost(0, 0, 1.7))
11. assertEquals("Invalid input", calculate_total_cost(1, -1, -1))
12. assertEquals("Invalid input", calculate_total_cost(1, 0, 0))
13. assertEquals("Invalid input", calculate_total_cost(1, 1, 0.77))
14. assertEquals("Invalid input", calculate_total_cost(1, -1, 1))
15. assertEquals("Invalid input", calculate_total_cost(1, 0, 1.7))

```
16.assertEqual("Invalid input", calculate_total_cost(1.7, -1, -1))
17.assertEqual("Invalid input", calculate_total_cost(1.7, 0, 0))
18.assertEqual("Invalid input", calculate_total_cost(1.7, 1, 0.77))
19.assertEqual("Invalid input", calculate_total_cost(1.7, -1, 1))
20.assertEqual("Invalid input", calculate_total_cost(1.7, 0, 1.7))
```

- باید دقت شود که اگر شرط اول کد پاس شود و مقدار تخفیف معتبر نباشد، ما ارور نمی‌دهیم و این یک باگ است.