

Github Repository: <https://github.com/PashaBarahimi/Software-Testing-Course-Projects>

Last Commit: cdcffe3e45012ef8e7e5910b13e9c4e33f0269bd

سوال اول

$$p = (a \wedge \neg b) \vee (\neg a \wedge \neg c) \vee (a \wedge \neg c)$$

1. Clause-ها

در این predicate، سه clause داریم که a و b و c هستند.

2. جدول درستی

#	a	b	c	P
1	T	T	T	F
2	T	T	F	T
3	T	F	T	T
4	T	F	F	T
5	F	T	T	F
6	F	T	F	T
7	F	F	T	F
8	F	F	F	T

محاسبه شرایط major clause بودن سه clause مسئله:

یادآوری:

$$X \oplus Y = (\neg X \wedge Y) \vee (X \wedge \neg Y)$$
$$P_x = P_{x=True} \oplus P_{x=False}$$

شرایط a:

$$\begin{aligned} P_{a=True} &= (True \wedge \neg b) \vee (False \wedge \neg c) \vee (True \wedge \neg c) \\ &= \neg b \vee False \vee \neg c = \neg b \vee \neg c \\ &= \neg(b \wedge c) \end{aligned}$$

$$\begin{aligned} P_{a=False} &= (False \wedge \neg b) \vee (True \wedge \neg c) \vee (False \wedge \neg c) \\ &= False \vee \neg c \vee False \\ &= \neg c \end{aligned}$$

$$\begin{aligned} P_a &= P_{a=True} \oplus P_{a=False} \\ &= \neg(b \wedge c) \oplus \neg c \\ &= ((b \wedge c) \wedge \neg c) \vee (\neg(b \wedge c) \wedge c) \\ &= (b \wedge False) \vee ((\neg b \vee \neg c) \wedge c) \\ &= False \vee ((\neg b \wedge c) \vee (\neg c \wedge c)) \\ &= (\neg b \wedge c) \vee False \\ P_a &= \neg b \wedge c \end{aligned}$$

شرایط b:

$$\begin{aligned}P_{b=True} &= (a \wedge False) \vee (\neg a \wedge \neg c) \vee (a \wedge \neg c) \\&= (\neg a \wedge \neg c) \vee (a \wedge \neg c) \\&= (\neg a \vee a) \wedge \neg c \\&= \neg c\end{aligned}$$

$$\begin{aligned}P_{b=False} &= (a \wedge True) \vee (\neg a \wedge \neg c) \vee (a \wedge \neg c) \\&= a \vee (\neg a \wedge \neg c) \vee (a \wedge \neg c) \\&= a \vee \neg c\end{aligned}$$

$$\begin{aligned}P_b &= P_{b=True} \oplus P_{b=False} \\&= \neg c \oplus a \vee \neg c \\&= (c \wedge (a \vee \neg c)) \vee (\neg c \wedge \neg(a \vee \neg c)) \\&= (c \wedge a) \vee (\neg c \wedge (\neg a \wedge c)) \\&= (c \wedge a) \vee False\end{aligned}$$

$$P_b = c \wedge a$$

شرایط c:

$$\begin{aligned}P_{c=True} &= (a \wedge \neg b) \vee (\neg a \wedge False) \vee (a \wedge False) \\&= a \wedge \neg b\end{aligned}$$

$$\begin{aligned}P_{c=False} &= (a \wedge \neg b) \vee (\neg a \wedge True) \vee (a \wedge True) \\&= (a \wedge \neg b) \vee \neg a \vee a \\&= (a \wedge \neg b) \vee True \\&= True\end{aligned}$$

$$\begin{aligned}P_c &= P_{c=True} \oplus P_{c=False} \\&= (a \wedge \neg b) \oplus True \\&= \neg(a \wedge \neg b) \\P_c &= \neg a \vee b\end{aligned}$$

3. GACC (General Active CC)

در GACC، همه جفت‌هایی که مقدار major clause متفاوت دارند را حساب می‌کنیم.

حالت major clause بودن a:

$$P_a = \neg b \wedge c = True \rightarrow rows = [3, 7]$$

$$TR = \{(3,7)\}$$

حالت major clause بودن b:

$$P_b = c \wedge a = True \rightarrow rows = [1, 3]$$

$$TR = \{(1,3)\}$$

حالت major clause بودن c:

$$P_c = \neg a \vee b = True \rightarrow rows = [1, 2, 5, 6, 7, 8]$$

$$TR = \{(1,2), (1,6), (1,8), (5,2), (5,6), (5,8), (7,2), (7,6), (7,8)\}$$

4. RACC (Restricted Active CC)

RACC زیرمجموعه CACC (Correlated Active CC)، و CACC زیرمجموعه GACC می باشد.

CACC، جفت هایی از GACC است که مقدار predicate متفاوتی دارند.

در RACC، همه minor clause های جفت ها باید مقدار یکسانی داشته باشند.

حالت major clause بودن a:

$$TR = \{(3,7)\}$$

حالت major clause بودن b:

$$TR = \{(1,3)\}$$

حالت major clause بودن c:

$$TR = \{(1,2), (5,6), (7,8)\}$$

5. RICC (Restricted Inactive CC)

در ابتدا GICC (General Inactive CC) را محاسبه می کنیم.

RICC جفت هایی از GICC است که minor clause های آن مقدار یکسانی دارند.

برای محاسبه GICC، بین سطرها جفت هایی را انتخاب می کنیم که major clause متفاوت است و predicate یکسانی دارند.

حالت major clause بودن a:

$$P_a = \neg b \wedge c = \text{False} \rightarrow \text{rows} = [1, 2, 4, 5, 6, 8]$$

$$TR = \{(1,5), (2,6), (2,8), (4,6), (4,8)\}$$

حالت major clause بودن b:

$$P_b = c \wedge a = \text{False} \rightarrow \text{rows} = [2, 4, 5, 6, 7, 8]$$

$$TR = \{(2,4), (2,8), (4,6), (5,7), (6,8)\}$$

حالت major clause بودن c:

$$P_c = \neg a \vee b = \text{False} \rightarrow \text{rows} = [3, 4]$$

$$TR = \{(3,4)\}$$

محاسبه RICC:

حالت major clause بودن a:

$$TR = \{(1,5), (2,6), (4,8)\}$$

حالت major clause بودن b:

$$TR = \{(2,4), (5,7), (6,8)\}$$

حالت major clause بودن c:

$$TR = \{(3,4)\}$$

سوال دوم

کد به صورت زیر است:

```
def calculate_total_cost(quantity, unit_price, discount):  
    if quantity <= 0 or unit_price <= 0:  
        return "Invalid input"  
    else:  
        total_cost = quantity * unit_price * (1 - discount)  
        return total_cost
```

در این کد، 3 پارامتر زیر را داریم:

1. quantity

برای این پارامتر می‌توانیم چهار گروه (block) زیر را در نظر بگیریم:

- A0: کوچکتر از 0 (مقدار غیر قابل قبول)
- A1: برابر با 0 (مقدار غیر قابل قبول)
- A2: برابر با 1 (مقدار قابل قبول)
- A3: بیشتر از 1 (مقدار قابل قبول)

2. unit_price

برای این پارامتر سه گروه (block) زیر را در نظر می‌گیریم:

- B0: کوچکتر از 0 (مقدار غیر قابل قبول)
- B1: برابر با 0 (مقدار غیر قابل قبول)
- B2: بیشتر از 0 (مقدار قابل قبول)

3. discount

برای این پارامتر نیز پنج گروه (block) زیر را در نظر می‌گیریم:

- C0: کوچکتر از 0 (مقدار غیر قابل قبول)
- C1: برابر با 0 (مقدار قابل قبول)
- C2: بین 0 و 1 (مقدار قابل قبول)
- C3: برابر با 1 (مقدار قابل قبول)
- C4: بیشتر از 1 (مقدار غیر قابل قبول)

با توجه به اینکه روش PWC انتخاب شده، می‌توانیم تست‌های زیر را انتخاب کنیم:

A0, B0, C0	A0, B2, C1	A0, B1, C2	A0, B0, C3	A0, B0, C4
A1, B1, C0	A1, B0, C1	A1, B2, C2	A1, B1, C3	A1, B1, C4
A2, B2, C0	A2, B1, C1	A2, B0, C2	A2, B2, C3	A2, B2, C4
A3, B0, C0	A3, B2, C1	A3, B2, C2	A3, B2, C3	A3, B1, C4

تست کیس ها به صورت زیر خواهند بود:

```
import unittest

from main import calculate_total_cost

INVALID_INPUT = "Invalid input"

A = [-1, 0, 1, 2]
B = [-1, 0, 1]
C = [-1, 0, 0.5, 1, 2]

class TestCalculateTotalCost(unittest.TestCase):
    def test_invalid_inputs_should_fail(self):
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[0], B[0], C[0]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[0], B[2], C[1]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[0], B[1], C[2]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[0], B[0], C[3]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[0], B[0], C[4]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[1], B[1], C[0]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[1], B[0], C[1]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[1], B[2], C[2]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[1], B[1], C[3]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[1], B[1], C[4]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[2], B[1], C[1]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[2], B[0], C[2]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[3], B[0], C[0]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[3], B[1], C[4]))

    def test_invalid_discount_should_fail(self): # not covered in the code
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[2], B[2], C[0]))
        self.assertEqual(INVALID_INPUT, calculate_total_cost(A[2], B[2], C[4]))

    def test_valid_inputs_should_pass(self):
        self.assertEqual(0, calculate_total_cost(A[2], B[2], C[3]))
        self.assertEqual(2, calculate_total_cost(A[3], B[2], C[1]))
        self.assertEqual(1, calculate_total_cost(A[3], B[2], C[2]))
        self.assertEqual(0, calculate_total_cost(A[3], B[2], C[3]))

if __name__ == '__main__':
    unittest.main()
```

همانطور که در کامنت گفته شده، در بازه صحیح بودن تخفیف در کد چک نشده و باگ دارد. به همین دلیل این 2 تست پاس نمی شوند:

```
-----
Traceback (most recent call last):
  File "d:\Lessons\Test\Software-Testing-Course-Projects\Baloot1\Reports\Extras\CA4\test.py", line 30, in test_invalid_discount_should_fail
    self.assertEqual(INVALID_INPUT, calculate_total_cost(A[2], B[2], C[0]))
AssertionError: 'Invalid input' != 2
-----
Ran 3 tests in 0.001s

FAILED (failures=1)
```