



تمرین سوم مهندسی نرم افزار

اعضا:

نیما سالم ۹۷۱۰۶۰۰۲

امیرمحمد قاسمی ۹۷۱۰۰۴۹۵

یگانه قره‌داغی ۹۷۱۰۶۲۱۶

حمیدرضا کامکاری ۹۷۱۱۰۱۷۷

سید علیرضا هاشمی ۹۷۱۰۲۶۲۲

سوال اول

بخش الف)

$$TR = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

$$TP = \{[0, 1, 2, 3, 4, 5, 6, 10], \\ [0, 1, 2, 3, 4, 7, 8, 2, 10], \\ [0, 1, 2, 3, 4, 7, 9, 2, 10]\}$$

بخش ب)

$$TR = \{(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 10), (4, 7), (7, 8), (8, 2), (2, 10), (7, 9), (9, 2)\}$$

$$TP = \{[0, 1, 2, 3, 4, 5, 6, 10], \\ [0, 1, 2, 3, 4, 7, 8, 2, 10], \\ [0, 1, 2, 3, 4, 7, 9, 2, 10]\}$$

بخش ج)

$$TR = \{[0, 1, 2], [1, 2, 3], [1, 2, 10], [2, 3, 4], [3, 4, 5], [4, 5, 6], [5, 6, 10], [3, 4, 7], [4, 7, 8], [7, 8, 2], [8, 2, 10], [4, 7, 9], [7, 9, 2], [9, 2, 10], \}$$

$$TP = \{[0, 1, 2, 10], \\ [0, 1, 2, 3, 4, 7, 8, 2, 10], \\ [0, 1, 2, 3, 4, 7, 9, 2, 10], \\ [0, 1, 2, 3, 4, 7, 8, 2, 3, 4, 7, 9, 2, 3, 4, 5, 6, 10]\}$$

سوال دوم

بخش الف)

کلازهای موجود ۴ تا هستند و عبارتند از: $a - b - c - d$
تاثیر کلازهای a و b روی P مشابه یکدیگر است؛ پس از بین این دو کلاز، تنها تاثیر یکی از آنها را بررسی می‌کنیم:

تاثیر کلاز a

اگر a درست باشد، باقی عبارت به صورت:

$(b \text{ or } c) \text{ and } d$

می‌شود.

اگر a غلط باشد، باقی عبارت به صورت:

$c \text{ and } (!b \text{ or } d)$

می‌شود.

با xor گرفتن از دو عبارت بالا، به حالتی می‌رسیم که در آنها، کلاز a عبارت P را تعیین می‌کند:

شماره حالت	b	c	d	$(b \text{ or } c) \text{ and } d$
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	0
8	1	1	1	1

شماره حالت	b	c	d	c and (!b or d)
1	0	0	0	0
2	0	0	1	0
3	0	1	0	1
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0
7	1	1	0	0
8	1	1	1	1

با xor گرفتن از دو عبارت بالا، می‌فهمیم که مقدار کلاز a در حالت‌های ۳ و ۶ تاثیرگذار است:

شماره حالت	(b or c) and d	c and (!b or d)	حاصل xor
1	0	0	0
2	0	0	0
3	0	1	1
4	1	1	0
5	0	0	0
6	1	0	1
7	0	0	0
8	1	1	0

تأثیر کلاز c

وقتی c درست باشد:

شماره حالت	a	b	d	(!a and !b) or d
1	0	0	0	1
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	0
8	1	1	1	1

وقتی c غلط باشد:

شماره حالت	a	b	d	(a and b) and (!a and !b) or d
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	0
5	1	0	0	0
6	1	0	1	0
7	1	1	0	0
8	1	1	1	1

حاصل xor دو حالت: حالت‌های ۱ و ۲ و ۴ و ۶

تأثیر کلاز d

وقتی d درست باشد:

شماره حالت	a	b	c	(a and b) or c
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

وقتی d غلط باشد:

شماره حالت	a	b	c	((a and b) or c) and (!a and !b)
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	0
5	1	0	0	0
6	1	0	1	0
7	1	1	0	0
8	1	1	1	0

حاصل xor دو حالت: حالت‌های ۴ و ۶ و ۷ و ۸

بخش ب)

a	b	c	d	مقدار نهایی
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

بخش ج)

Predicate Coverage:

TRUE:

$a=1 \wedge b=1 \wedge c=1 \wedge d=1$

FALSE:

$a=0 \wedge b=0 \wedge c=0 \wedge d=0$

Clause Coverage:

test TRUE value of each clause:

$a=1 \wedge b=1 \wedge c=1 \wedge d=1$

test FALSE value of each clause:

$a=0 \wedge b=0 \wedge c=0 \wedge d=0$

سوال سوم

بخش الف)

```
procedure iseven(x:int) return (q:int)
{
    q := 0
    if (x % 2 == 0)
        q := 1
    return q
}
```

$\forall x, q: Z \bullet (q = 0 \wedge x \% 2 = 0) \vee (q \neq 0 \wedge x \% 2 \neq 0)$

$(q = 0) \rightarrow a$

$(x \% 2 = 0) \rightarrow b$

$p: (a \wedge b) \vee (\neg a \wedge \neg b)$

بخش ب)

```
procedure sum (x:int, y:int) return (q:int)
{
    q = x + y
    return q
}
```

$\forall x, y, q : Z \bullet (x + y = q)$

$(x + y = q) \rightarrow a$

$p : a$

سؤال چهارم

بخش الف)

تمامی متغیرها عبارتند از x و arr که به عنوان ورودی داده شده‌اند و متغیر i که شمارنده‌ایست که در حل مسئله و به دست آوردن خروجی کمک می‌کند.

بخش ب)

خصوصیات مربوط به واسط زمانی‌هست که به صورت blackbox به تابع می‌نگریم برای مثال اینکه آرایه تهی‌هست یا نه و یا اینکه x مقدار دارد یا نه مربوط به واسط هست.

خصوصیات مربوط به عملکرد به نحوه پیاده‌سازی داخل تابع بر می‌گردد برای مثال اینکه x در کدام جایگاه آمده ممکن است در سرعت اجرای تابع تاثیر بگذارد. بنابراین یکی از خصوصیات می‌تواند این باشد که x در نیمه اول آرایه آمده‌است یا نه.

مورد دیگری که می‌توانیم در نظر بگیریم تعداد رخدادهای x در آرایه‌است. برای مثال یک روش ممکن است ابتدا تمامی مکان‌هایی که x آمده را پیدا کند و بین آن‌ها مینیمم بگیرد در حالیکه روش دیگری از ابتدای آرایه شروع کند و به اولین x ای که رسید گزارش دهد. بنابراین در این حالت این یک خصوصیت است که بسته به رفتار تابع می‌تواند تغییر کند.

بنابراین ۴ تا criterion به صورت زیر تعریف می‌شوند:

A = is the array empty or not?

B = is 'x' empty or not?

C = is 'x' available in the first half of the array?

D = How many times has x repeated in the array?

بخش ج)

برای بلوک بندی مقادیر زیر را تعیین می‌کنیم:

A: ($a1$ = array is empty) ($a2$ = array is not empty)

B: ($b1$ = x is empty) ($b2$ = x is not empty)

C: ($c1$ = x is available in the first half) ($c2$ = x is not available in the first half)

D:

- ($d1$ = x has not repeated at all)

- ($d2 = x$ has repeated in $[avg - sq, avg + sq]$ times where 'avg' is the average and sd is the standard deviation of the number of times 'x' occurs)
- ($d3 = x$ has repeated at least once but not 'avg' times)

همانطور که می‌بینید تعداد مقادیر A و B و C برابر ۲ تاست ولی D مقادیرش به سه تا بلوک تقسیم شده که بلوک وسطی از همه بیشتر ممکن است.

بخش د)

ابتدای کار حالت‌های پایه را به ازای هر خصوصیت می‌نویسیم. این حالت‌های پایه حالت‌هایی هستند که از همه مرسوم‌ترند:

$$A = a2, B = b2, C = c2, D = d2$$

حالا یک تست به صورت

(a2, b2, c2, d2)

تعریف می‌کنیم، سپس به ازای هر متغیر نیز بقیه حالت‌ها را ثابت می‌گیریم و فقط همان متغیر را عوض می‌کنیم تا به تست‌های زیر برسیم:

- (a1, b2, c2, d2) <Change A> IMPOSSIBLE
- (a2, b1, c2, d2) <Change B> IMPOSSIBLE
- (a2, b2, c1, d2) <Change C>
- (a2, b2, c2, d1) <Change D>
- (a2, b2, c2, d3) <Change D>

توجه کنید از بین تست‌های اشاره شده تست اول غیر ممکن است چرا که اگر آرایه خالی باشد تعداد تکرار صفر باید صفر باشد و لذا $D=d1$. همینطور وقتی 'x' به درستی تعریف نشده هم باید $D=d1$ باشد.