**Caja blanca**

Se tiene los detalles de implementacion para diseñar los casos de prueba

Cobertura de enunciados o caminos básicos

Cobertura de sentencias

Cobertura de decisión

Cobertura de condición

Cobertura de decisión/condición

Cobertura multiple

Hay muchas mas pero se debe acotar en estas para el efecto del practico

Cobertura es la estrategia para recorrer los diferentes caminos de diagrama de flujo de nuestro código

COBERTURA DE ENUNCIADO O CAMINO BASICOS

Se intenta cubrir todos los caminos independientes, una vez realizada este casos de prueba se garantiza que se recorrieron todos los caminos de la implementación

La complejidad de un escenarios nos da una ida de cuantos casos de prueba son necesarios

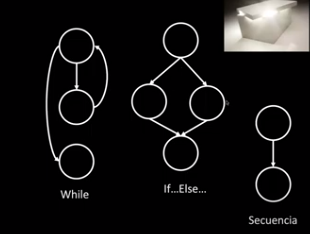
Para la prueba de camiono básico

Se requiere representar la ejecución se representa mediante grafos de flujo(OJO CON LOS METODOS RECURSIVOS)

Se calcula la complejidad ciclomatica

Dado un grafo de flujo se pueden generar casos de prueba

COMO SE GENERAN LOS CASOS DE PRUEBA



El switch es un if else con varias ramas

Es el método mas sencillo para la idea de cobertura pero es demasiado trivial

Complejidad ciclomatica=caminos independientes=Casos de prueba

M->Complejidad ciclomatica

E->Numero de aristas del grafo

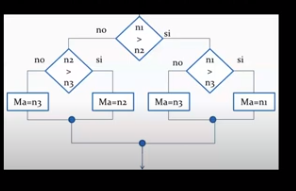
N->Numero de nodos del grafo

P->Numero de componentes conexos,nodos salida

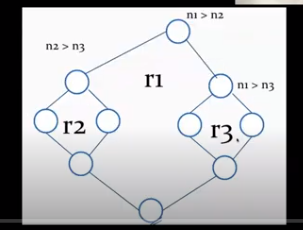
M=E-N+3\*P

M=Numero de regiones +1 forma visual de calcularlo

Ejemplo



Grafo



Como se diseña cao de pruba

Para garantizar todos los caminos independientes en este caso 4

Se pueden escribir mas casos de prueba del minimo pero el chiste es aplicar la economía de detección defectos

|  |  |  |  |
| --- | --- | --- | --- |
| Tc1 | Tc2 | Tc3 | Tc4 |
| N1=1 | Nq=1 | N1=2 | N1=4 |
| .. | .. | .. | .. |

Nos sirve para desarrollar los casos de prueba pero no son casos de prueba

Una heurística que nos puede servir Es que si

Complejidad |Evalucion de risego

1-10 |Programa mas simple sin mucho riesgo

11-20 | Mas complejo pero riesgo moderado

21-50 |Complejo, muy alto riesgo

50 | No es testeable.Se puede optimizar el código, modularizar, establecer tc priorizadas

PASOS

Obtner el grafo de flujo

Obtenre la complejidad

Definir los camino independientes

Determinar los casos de prueba que permitan la ejecución de cada camino independiente

Ejecutar cada caso de prueba y comprobar que los resultaso son los esperados

COBERTURA DE SENTENCIAS

Ejemplo

IF(a>0&&c==1)

x=x+1

END IF

IF(B==3||D<0)

Y=0

End

Cuantas senticias hay?

2 := x=x+1 ^ y=0

Una sentencia es cualquier instrucción mientras no sean estructura de control

El objetivo de esta cobertura es ejecutar todas las sentencias en el menor numero de casos de prueba

En este caso caso es ejecutar solamente un caso de prueba

Palabra clave “no es excluyentes” “son independientes”

|  |
| --- |
| Tc1 |
| A=5 |
| C=1 |
| B=3 |
| D=-3 |

Con estos valores se realizaría un solo caso de prueba para cubrir todas las sentencias

No van a ser asignaciones de valores numéricas

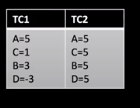
Hay que tener en cuenta las reglas de negocio, para ver si una combinación de variables es aplicable.

COBERTURA DE DECISIÓN

Una decisión es una estructura de control completa en donde se bifurcan caminos. Si hay estrucuturas de control entonces no hay caminos independientes

El objetivo es ver que cada decisión funcione correctamente, osea que se vaya a uno u otro camino según lo que deseo.Es independiente de las sentencias.

Basada en esta definición se definene 2 casos de prueba.En este caso sirve porque las variable dentro de esto los if son independientes entonces en un test case puedo probar un camino por decisión y en el otro tc probamos los otros caminos.Imposible sea por uno por que es por el camino de true and false del if.No importan las condiciones



Si hubera ramas anidados entonces haría falta un nuevo caso de prueba porque no se podrían probar los caminos del if dependiente.

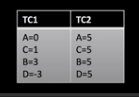
COBERTURA DE CONDICION

Busca encontrar la menor cantidad de casos de prueba que evalúen los diferentes ramas de las condiciones independientemente del resultado de la decisión.

Otro tema a tener en cuenta los if cortocircuitados no se toman en cuenta

Se necesitan 2 casos de prueba porque las condiciones son independientes. Osea que con un caso de prueba evaluo un camino de cada condición y con otro evaluo los demás camino

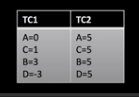
#OCHOOOOOOOOOOOOOO WHATTT?.#



COBERTURA DE DECISIÓN Y CONDICION

Lo que se busca valuar las condiciones verdero y falso y las desiciones verdaderos y falso.

Harian falta 2 casos de prueba ya que se encuentran 2 combinacion de variables que me permitan valuar ambos caminos de las condiciones y decisión en el mismo conjunto de valores de casos de prueba.



Es el mismo en este caso.

Si hacen la cobertura decisión y combinación puede extenderse a las coberturas de decisión y a la cobertura de condición.

COBERTURA MULTIPLE

Busca valuar el combinatorio de los valores de verdad para toda la condición de valor disponibles.

En este caso considerando que las condiciones son independienes con 4 casos de prueba estaría bien.

Un ejemplo distinto seria

IF(a>0&&c==1)

x=x+1

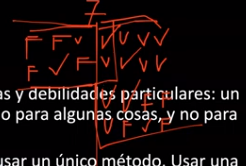
IF(B==3||D<0)

Y=0

END IF

End IF

En este caso de anidamiento



Se necesitan 7 casos de prueba.

Ya que la rama de arriba tiene que ser verdero y verdadero para ir al if de abajo

Ojo una vez mas con las reglas de negocio ya que no permite que aparezacan ciertas combinaciones por la semánticas y ni se tiene en cuenta.

ELEGIR UN METODO

Lo ideal no es usar todos o ninguno sino un conjunto que nos complete lo que deseamos probar

REALIZAR EJERCICO

Datos de los casos de prueba datos específicos para llevar al caso de prueba en lado deseado

Precondiciones algunas mas de las que están aclarads

CORRECCIONES

Aprovechar las condiciones en este casi el estado

CLAAAAAAAAAAAAAAAAAARO LAS PRECONDICINES SON LOS DATOS CARGADOS EN EL SISTEMA POR ESOOOOOOOOOOOOO LA 555 ES INEXISTENE.POR QUE NO ESTA CARGADAAAAAAAAAA.CLAROOOOOOOOO

Si coinciden las combinaciones se copian y pegan

En este caso el or de solicitado y ocupado vendría a ser un XOR por que lo estados son dependientes.

En la mayoría de los ejercicios practicos se pueden usar la misma combinatria de cobertura. Con el mismo conjunto se caracteriza a ambas por lo que se copia y pega. Es lo mas optimo.

Listorti

Son 5 tc para la prueba de sentencia y 6 para las demás-