Universidade Federal do Rio de Janeiro
Projeto de Teste de Software

Cobertura de Requisitos Lógicos

Alunos:
César Augusto Julio da Silva
João Ricardo Campbell Maia
Rodrigo Delpreti de Siqueira

Cobertura de Predicados

Primeiramente, vamos identificar todos os predicados do programa:
1 - (i < N)
2 - (j < M)
3 - (m == 0 && flag)
4 - (c <= lastc && c < M)

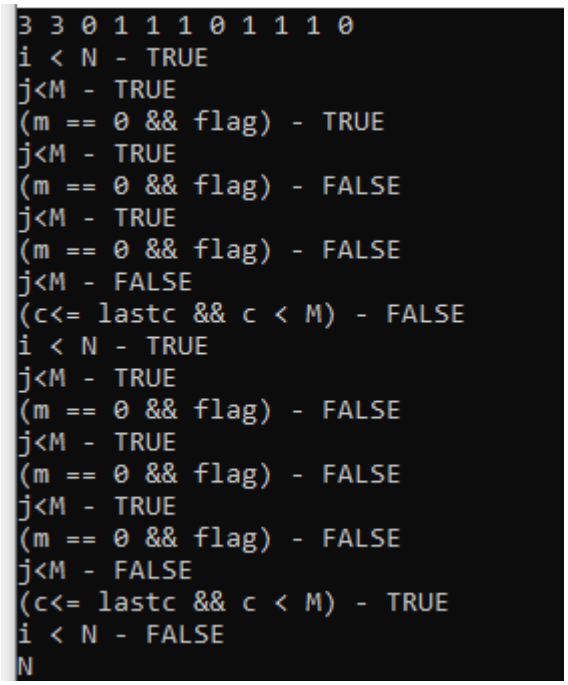
Sendo assim, a partir da cobertura de predicados, os requisitos são:

1 - P1 = T, P1 = F
2 - P2 = T, P2 = F
3 - P3 = T, P3 = F
4 - P4 = T, P4 = F

Ou seja, precisamos fazer com que cada predicado assuma o valor verdadeiro e falso pelo menos uma vez.

Podemos modificar o programa para permitir a fácil verificação do valor dos predicados, apenas inserimos um print com o valor de cada comparação

CT1:
3 3 0 1 1 1 0 1 1 1 0 - Output N



Percebemos que todos os predicados foram cobertos com apenas um caso de teste, logo não é necessária a criação de mais casos de testes.

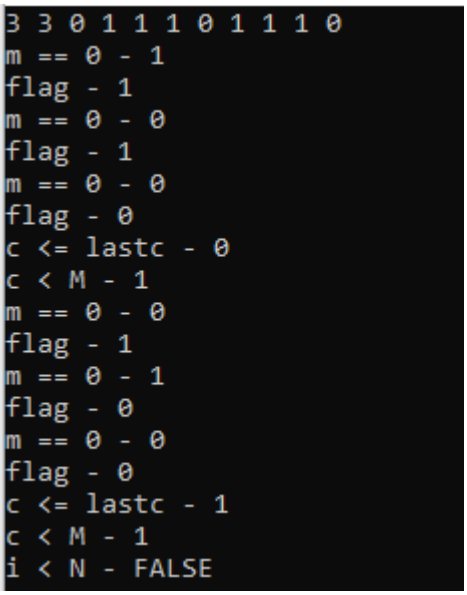
Cobertura de Cláusulas

Para a cobertura de cláusulas, vamos identificar todas as cláusulas do programa:

1 - i < N
2 - j < M
3 - m == 0
4 - flag
5 - c <= lastc
6 - c < M

Os requisitos serão todas as cláusulas apresentarem pelo menos um valor verdadeiro e um falso, aproveitando nosso último caso de teste e fazendo outra adaptação no código para mostrar o resultado de cada cláusula da expressão, temos:

CT1:
3 3 0 1 1 1 0 1 1 1 0 - Output N



Restou apenas uma claúsula que não foi satisfeita, c < M == FALSE

CT2: 4 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 - Output N
Nesse caso possuímos os dois valores para a cláusula c < M.

Sendo assim, todas as nossas cláusulas estão cobertas.

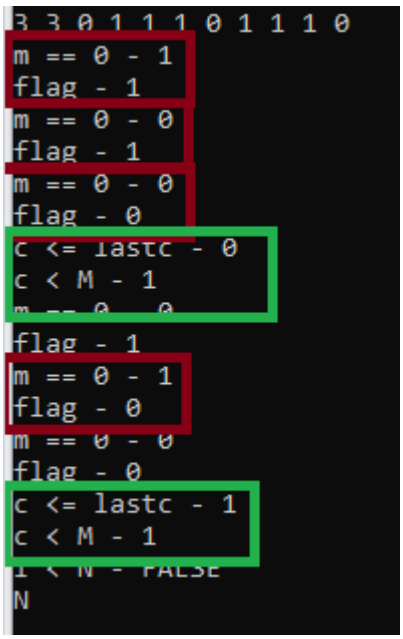
As cláusulas do for foram omitidas pois elas sempre apresentarão dois valores para N,M > 0

Cobertura Combinatorial

Na cobertura combinatorial, os requisitos são todos os valores possíveis para cada cláusula dentro de um predicado, então os requisitos são todas as linhas da tabela verdade do predicado.

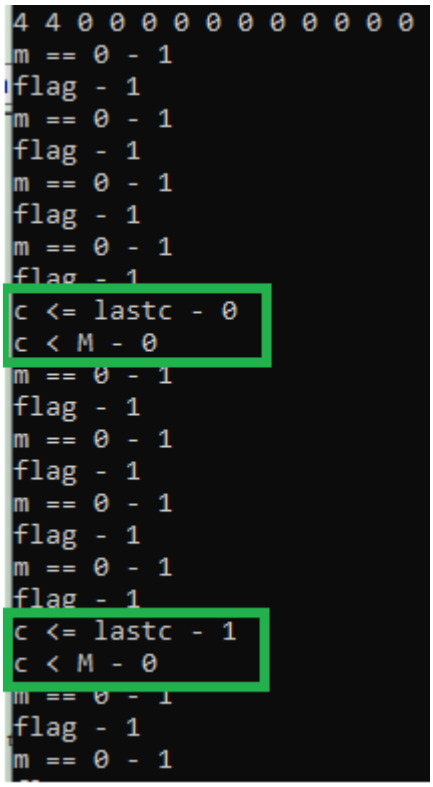
Aproveitando nosso primeiro caso de teste, vamos observar quais requisitos não são cumpridos, lembrando que o for sempre será coberto para valores de N e M > 0.

CT1 3 3 0 1 1 1 0 1 1 1 0 - Output N



Observamos que os únicos requisitos que não foram cobertos são:
1 - (c <= lastc && c < M) F, F
2 - (c <= lastc && c < M) T, F

CT2: 4 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 - Output N



Utilizamos esse último caso de teste para suprir os requisitos faltantes.

RACC

Para esse caso específico, o RACC acaba sendo equivalente aos outros critérios de cláusula ativa, já que existem poucas cláusulas dentro dos predicados, como o operador lógico entre as cláusulas é um AND, para tornar uma cláusula ativa, bastar tornar o valor da outra variável verdadeiro.

Os requisitos são:

1 - (m == 0 && flag) T, T
2 - (m == 0 && flag) T, F
3 - (m == 0 && flag) F, T
(1,3) Satisfaz m == 0
(1,2) Satisfaz flag
4 - (c <= lastc && c < M) T, T
5 - (c <= lastc && c < M) T, F
6 - (c <= lastc && c < M) F, T
(4,6) Satisfaz c <= lastc
(4,5) Satisfaz c < M

Como sabemos que RACC subsume o CoC, podemos reaproveitar os casos de testes do CoC com garantia de cobrir todos os requisitos do RACC.

CT1 3 3 0 1 1 1 0 1 1 1 0 - Output N
CT2: 4 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 - Output N