Argumentação de Corretude

Leonardo Rocha

Bárbara Castro

Rudá Valle

Função com suas Assertivas:

AE

TAB\_tpCondRet TAB\_CriarTabuleiro(TAB\_tpTabuleiro ptabuleiro)

{

int i, j;

if (ptabuleiro == NULL)

{

return TAB\_CondRetFaltouMemoria;

}

else

{

i = 0;

AI1

while ( i < ptabuleiro->linha )

AINV1

{

j = 0;

AI3

while ( j < ptabuleiro->coluna )

AINV2

{

ptabuleiro->casa[i][j] = TAB\_IniciarCasa();

AI5

j++;

}

AI4

i++;

}

AI2

return TAB\_CondRetOK;

}

}

AS

Detalhamento das Assertivas:

AE

- Tabuleiro passado como parâmetro está devidamente alocado ou é NULL

- Linha passada como parâmetro é um número inteiro.

- Coluna passada como parâmetro é uma letra maiúscula, ou algum caractere após as letras maiúsculas na tabela ASCII.

AS

- O tabuleiro é devidamente gerado e suas casas inicializadas, ou, em caso de erro, é retornado um valor relativo ao erro encontrado.

* Argumentação de Seleção:

1 – AE && (C==T) + B1 =>AS: (De B2)

* Pela AE, ptabuleiro pode ser NULL, como (C==T), ptabuleiro é NULL, indicando falta de memória, em B1 é retornado o erro correspondente a falta de memória, vale a AS.

2 – AE && (C==F) + B2 =>AS: (De B2)

* - Pela AE, e como (C==F) ptabuleiro aponta para um tabuleiro já criado em memória dinâmica. Nesse caso o segundo bloco (B2.2) é executado, inicializando as casas do tabuleiro, sendo assim, vale a AS.
* Argumentação de Sequência:

AI1

- i aponta para a primeira linha de casas do tabuleiro, ou é maior do que ptabuleiro->linha, caso o tabuleiro não tenha linhas.

AI2

- i é maior do que ptabuleiro->linha, pois já foram percorridas todas as linhas do tabuleiro.

* Argumentação de Repetição:

AINV1

- i aponta para a primeira linha não percorrida .

1 – AE => AINV

* Pela AE, i aponta para a primeira linha do tabuleiro, como nenhuma linha do tabuleiro foi percorrida, vale AINV.

2 – AE && (C==F) => AS

* Pela AE, o tabuleiro pode não ter linhas, com isso (C==F) pois i=0 que não é menor do que 0, com isso vale AS, pois se o tabuleiro não tem linhas, ele não tem casas, então pode se considerar todas as suas casas como inicializadas.

3 – AE && (C==T) + B => AINV

* Pela AE, i aponta para a primeira linha do tabuleiro. Como (C==T), esta linha pertence ao tabuleiro, sendo assim, B será executado, inicializando as casas dessa linha e avançando para a próxima. Vale AINV.

4 – AINV && (C==T) + B => AINV

* Para que AINV continue valendo, B deve garantir que todas as casas da linha i foram inicializadas e i deve avançar para a linha seguinte a percorrer.

5 – AINV && (C==F) => AS

* No último ciclo, i é maior do que ptabuleiro->linha. Nesse caso, vale AS.

6 – Término

* Como a cada ciclo todas as casas de uma linha são inicializadas, e o tabuleiro possui um número finito de linhas, a repetição termina em um número finito de passos.
* Argumentação de Sequência:

AI3

- j aponta para a primeira coluna de casas do tabuleiro, ou é maior do que ptabuleiro->coluna, caso o tabuleiro não tenha colunas.

AI4

- j é maior do que ptabuleiro->coluna, pois já foram percorridas todas as colunas do tabuleiro, na linha i.

* Argumentação de Repetição:

AINV2

- j aponta para a primeira coluna não percorrida da linha i

1 – AE => AINV

* Pela AE, i aponta para a primeira coluna da linha i, como nenhuma coluna da linha i foi percorrida, vale AINV

2 – AE && (C==F) => AS

* Pela AE, o tabuleiro pode não ter colunas, com isso (C==F) pois j=0 que não é menor do que 0, com isso vale AS, pois se o tabuleiro não tem colunas, ele não tem casas, então pode se considerar todas as suas casas como inicializadas.

3 – AE && (C==T) + B => AINV

* Pela AE, j aponta para a primeira coluna da linha i. Como (C==T), esta coluna pertence ao tabuleiro, sendo assim, B será executado, inicializando as casas dessa coluna, na linha i e avançando para a coluna seguinte. Vale AINV

4 – AINV && (C==T) + B => AINV

* Para que AINV continue valendo, B deve garantir que a casa da coluna j da linha i seja inicializada e j deve avançar para a coluna seguinte a percorrer.

5 – AINV && (C==F) => AS

* No último ciclo, j é maior do que ptabuleiro->coluna. Nesse caso, vale AS

6 – Término

* Como a cada ciclo a casa da coluna j da linha i é inicializada, e o tabuleiro possui um número finito de colunas, a repetição termina em um número finito de passos
* Argumentação de Sequência:

AI5

- Pela AS da função TAB\_IniciarCasa a casa da coluna j, da linha i foi inicializada.