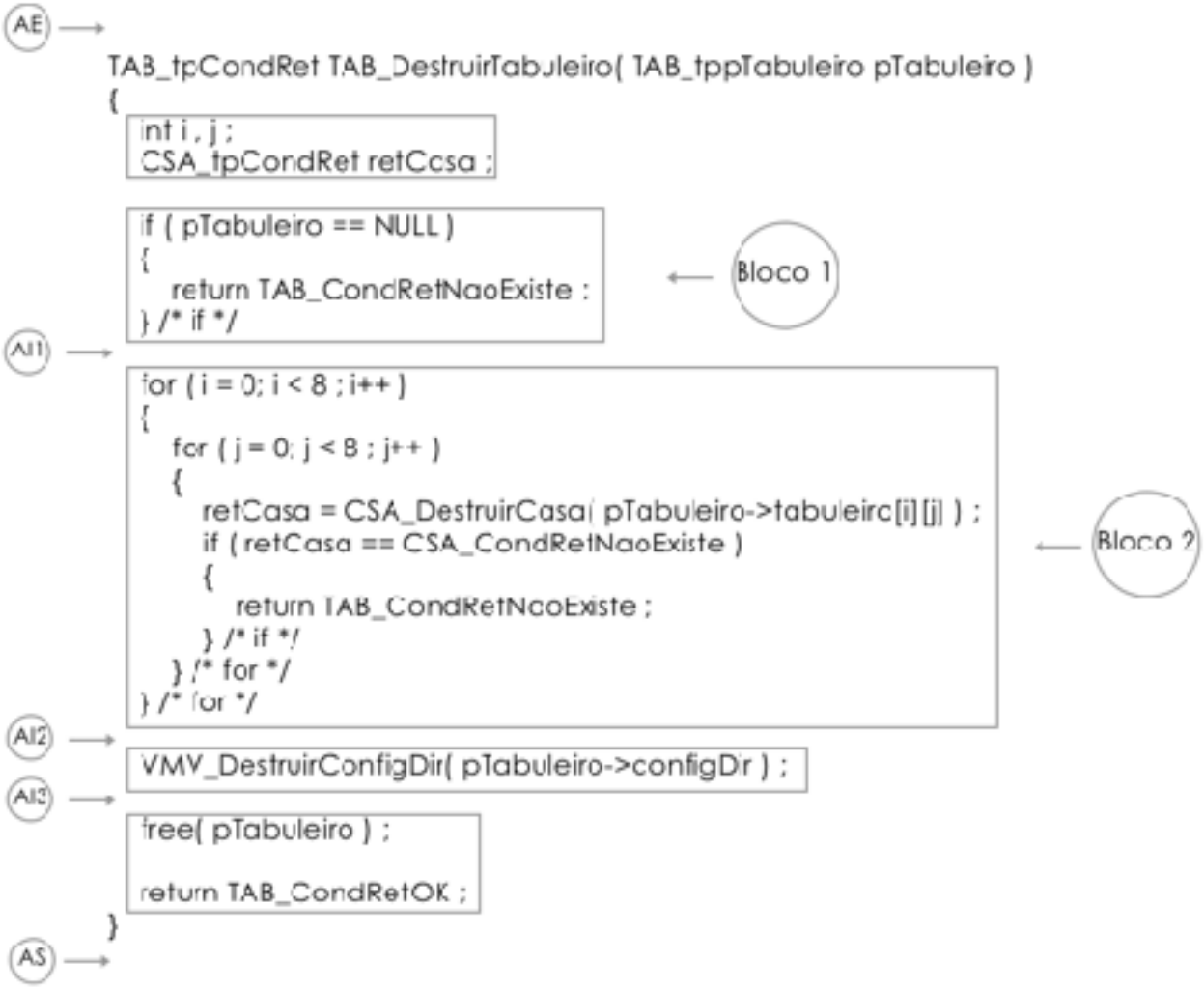
Rio de Janeiro, 13 de novembro de 2016
Pontifícia Universidade Católica do Rio de Janeiro
INF1301 - Programação Modular
Grupo: Ian Albuquerque Raymundo da Silva ;
Lucas Ferraço de Freitas;
Victor Augusto Lima Lins de Souza .

Argumentação de Corretude da Função TAB_DestruirTabuleiro

- Função:



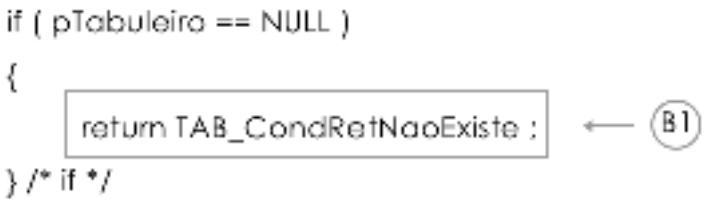
Argumentação:

- AE:
  - o tabuleiro pode não existir e, caso exista, pode estar vazio;
  - valem as assertivas estruturais do tabuleiro, caso o tabuleiro exista.
- AS:
  - o tabuleiro foi destruído e seu espaço na memória desalocado ou o tabuleiro ou alguma de suas casas não existe e a função retorna TAB_CondRetNaoExiste;
  - não valem mais as assertivas estruturais do tabuleiro.
- AI1:
  - o tabuleiro existe e pode estar vazio ou a função retorna TAB_CondRetNaoExiste, caso o tabuleiro seja nulo (não exista);

- AI2:
  - o tabuleiro não possui casas;

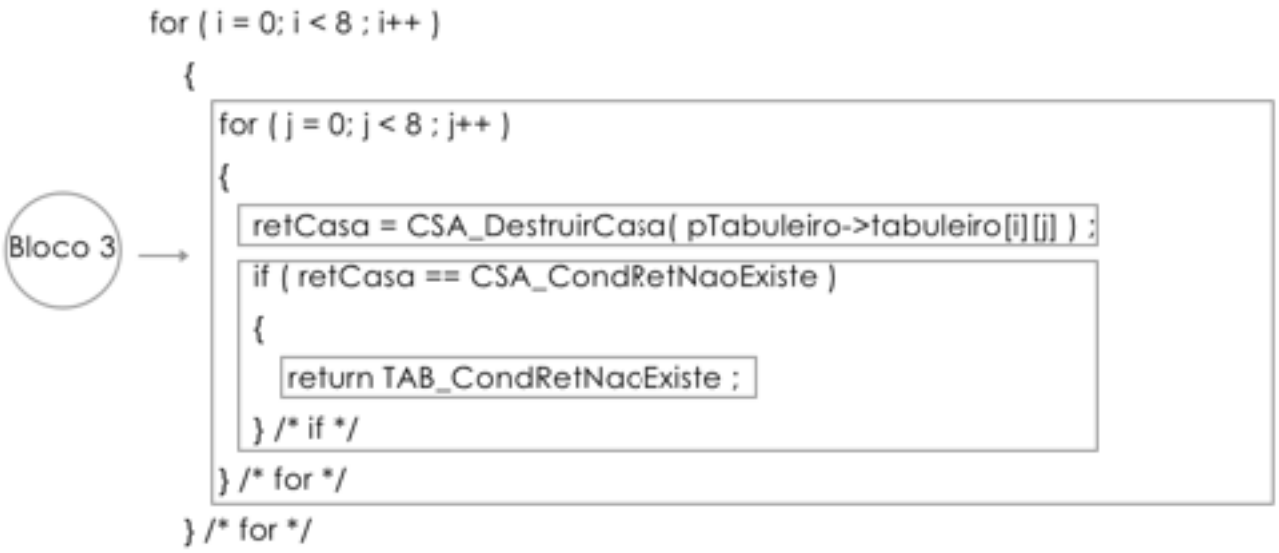
- AI3:
  - o configDir do tabuleiro foi destruído e seu espaço na memória desalocado (observação: não é necessário testar a existência do configDir, pois sua existência está atrelada ao do tabuleiro, e a existência desse já foi testada na função);

- Bloco 1:



  - AE: AE da função;
  - AS: AI1;
  - AE && (C == T) ⊕ B1 => AS: Pela assertiva de entrada, o tabuleiro pode não existir. Como a condição é verdadeira, então o tabuleiro é nulo. Assim, a função retorna TAB_CondRetNaoExiste indicando a situação do tabuleiro e a assertiva de saída do bloco é válida;
  - AE && (C == F) => AS: Pela assertiva de entrada, o tabuleiro pode existir. Como a condição é falsa, então o tabuleiro é existe. Assim, o tabuleiro existe e a assertiva de saída do bloco é válida;

- Bloco 2:



- AE: o tabuleiro existe e pode estar vazio;
- AS: é igual a AI2;
- AINV:
  a) i aponta para uma linha de casas da matriz que representa o tabuleiro;
  b) existem 2 conjuntos: linhas de casas já destruídas e linhas de casas a serem destruídas;

- AE => AINV: Pela AE, o tabuleiro existe, logo todas as suas linhas de casas pertencem ao conjunto que podem ser destruídas. O conjunto de linhas de casas destruídas está vazio.

- AE && (C == F) => AS: Pela AE, o tabuleiro pode estar vazio, fazendo a condição falsa, pois não existem linhas no tabuleiro, e também a AS, pois o tabuleiro não tem linhas e consequentemente não tem casas;

- AE && (C == T) ⊕ Bloco3 => AINV: Pela AE, o tabuleiro existe e pode não estar vazio, logo, i aponta para a primeira linha do tabuleiro. Como a condição é verdadeira, toda esta linha de casas foi posta no conjunto de linhas de casas destruídas pelo Bloco 3 e i agora aponta para a próxima linha, assim, a AINV é válida;

- AINV && (C == T) ⊕ Bloco3 => AINV: Para a AINV ser válida, o Bloco 3 deve garantir que uma linha de casas seja transferida para o conjunto de linhas de casas destruídas e i aponte para a próxima possível linha do conjunto de linhas de casas a serem destruídas;

- AINV && (C == F) => AS: No último ciclo, todas as possíveis linhas de casas do tabuleiro foram destruídas e, como a condição é falsa, i aponta para uma linha que não existe no tabuleiro. Logo, a AS é válida;

- Término: Como a cada ciclo uma linha de casas passa do conjunto de linhas de casas a serem destruídas para o de linhas de casas destruídas e a quantidade de linhas (elementos desse conjunto) é finita, a repetição termina após um número finito de passos;

- Bloco 3:

```
for ( j = 0; j < 8 ; j++ )
{
   retCasa = CSA_DestruirCasa( pTabuleiro->tabuleiro[i][j] ) ;
   if ( retCasa == CSA_CondRetNaoExiste )
   {
      return TAB_CondRetNaoExiste ;
   } /* if */
} /* for */
```

AE: o tabuleiro existe e i aponta para uma linha válida do mesmo, tanto a linha quanto todo o tabuleiro podem estar vazios;
- AS: a linha apontada por i do tabuleiro não possui casas;
- AINV:
  a) j aponta para uma casa da linha de casas da matriz que representa o tabuleiro apontada por i;
  b) existem 2 conjuntos: casas já destruídas e casas a serem destruídas;

- AE => AINV: Pela AE, o tabuleiro existe e i aponta para uma linha válida, logo todas as as casas dessa linha pertencem ao conjunto que podem ser destruídas. O conjunto de casas destruídas está vazio.

- AE && (C == F) => AS: Pela AE, o tabuleiro pode estar vazio, fazendo a condição falsa, pois não existem casas na linha apontada do tabuleiro, e também a AS, pois tal linha não tem casas;

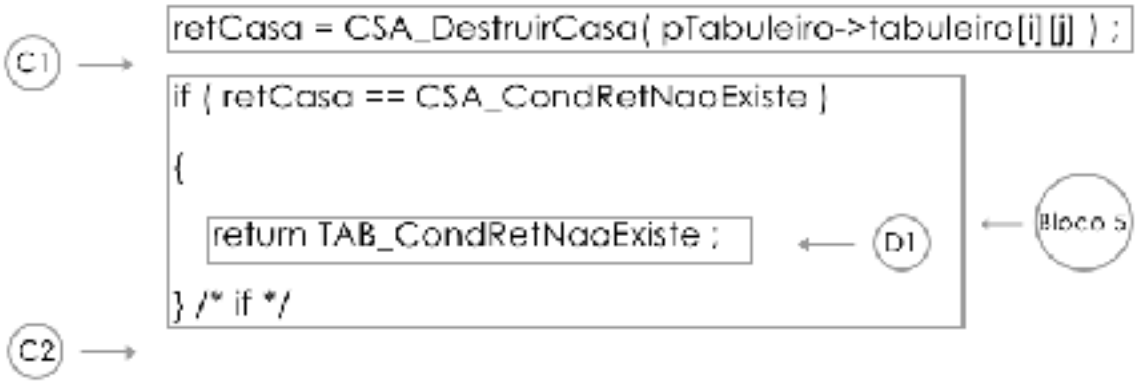
- AE && (C == T) ⊕ Bloco4 => AINV: Pela AE, o tabuleiro existe e pode não estar vazio, logo, i aponta para a primeira linha do tabuleiro e, consequentemente, j aponta para a primeira casa desta linha. Como a condição é verdadeira, esta casa foi posta no conjunto de casas destruídas pelo Bloco 4 e j agora aponta para a próxima casa dessa linha, assim, a AINV é válida;

- AINV && (C == T) ⊕ Bloco4 => AINV: Para a AINV ser válida, o Bloco 4 deve garantir que uma casa da linha apontada seja transferida para o conjunto de casas destruídas e j aponte para a próxima possível casa do conjunto de casas a serem destruídas;

- AINV && (C == F) => AS: No último ciclo, todas as possíveis casas da linha apontada do tabuleiro foram destruídas e j aponta para uma casa que não existe no tabuleiro. Logo, a AS é válida;

- Término: Como a cada ciclo uma casa de uma linha do tabuleiro passa do conjunto de casas a serem destruídas para o de casas destruídas e a quantidade de casas (elementos desse conjunto) é finita, a repetição termina após um número finito de passos;

- Bloco 4:



  - C1: retCasa guarda o resultado da operação de destruir a casa do tabuleiro na linha apontada por i e na coluna apontada por j (sucesso ou a casa não existe);
  - C2: a casa do tabuleiro na linha apontada por i e na coluna apontada por j foi destruída ou a função retorna TAB_CondRetNaoExiste, caso tal casa não exista;

- Bloco 5:
  - AE: C1;
  - AS: C2;
  - AE && (C == T) ⊕ D1 => AS: Pela assertiva de entrada, retCasa pode indicar que a casa não existe. Como a condição é verdadeira, então a casa é nula, não existe. Assim, a função retorna TAB_CondRetNaoExiste indicando tal situação e a assertiva de saída do bloco é válida;
  - AE && (C == F) => AS: Pela assertiva de entrada, retCasa pode indicar o sucesso na destruição da casa. Como a condição é falsa, então a casa foi destruída com sucesso. Assim, a casa referenciada foi destruída e a assertiva de saída do bloco é válida.