

AULA 8 - Argumentação de Corretude

Iniciais dos alunos: *AVC, JPP, PC*

Disciplina: *Programação Modular (INF1301)* – Professor: *Flavio Bevilacqua*
:

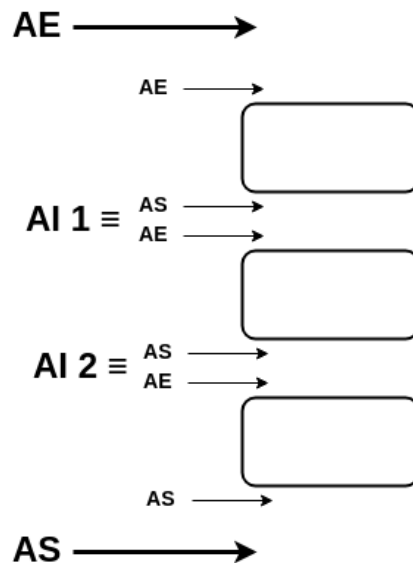
Objetivo: Garantir através de uma argumentação baseada nas assertivas definidas para um código, que o bloco/programa está correto. Um programa correto gera saídas válidas para qualquer entrada. Quanto mais próximo do que o cliente quer, mais alta a qualidade do que foi feito.

Tipos de Argumentação:

- Sequência
- Seleção
- Repetição

O descritor de estado é aquilo que indica evolução em um loop.

Argumentação de sequência:



Algoritmo 1: Excluir nó corrente intermediário de lista duplamente encadeada

AE \rightarrow **INÍCIO**

AJUSTA PONTEIRO NÓ ESQUERDO

AI 1 \rightarrow

AJUSTA PONTEIRO NÓ DIREITO

AI 2 \rightarrow

EXCLUI NÓ CORRENTE

AI 3 \rightarrow

REDIRECIONA PONTEIRO CORRENTE

FIMAS \rightarrow

AE:

- Valem as assertivas...
- Ponteiro corrente aponta para o nó a ser excluído.

AS:

- Valem as assertivas...
- Nó corrente foi excluído.
- Ponteiro corrente foi reposicionado para o primeiro nó da lista.

AI 1:

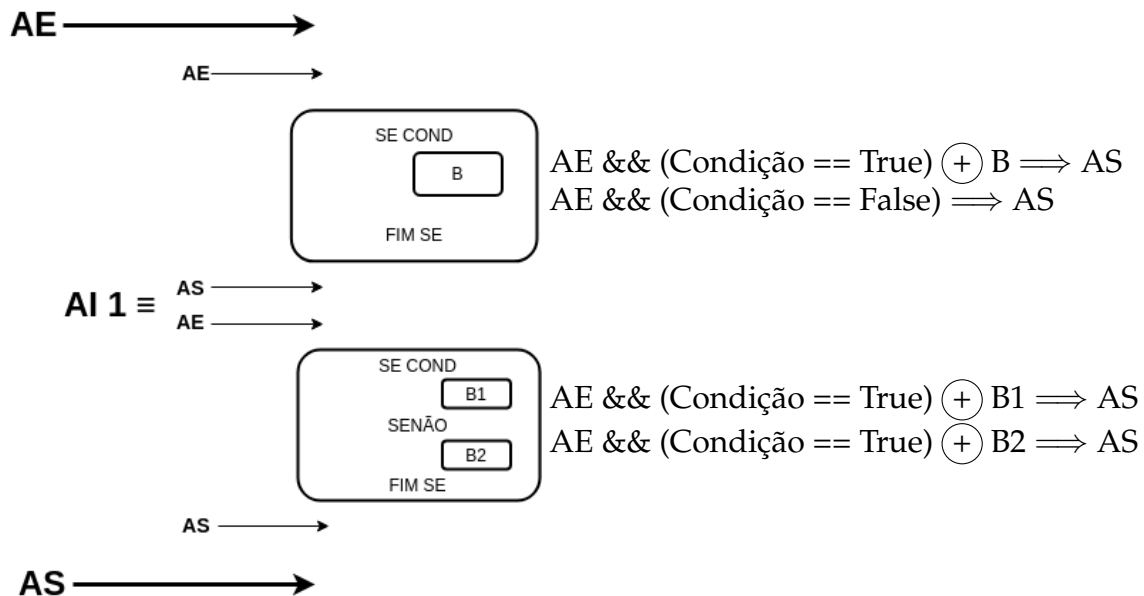
- Ponteiro do nó anterior ao corrente aponta para o próximo.
- Ponteiro para o nó anterior é nulo.

AI 2:

- Ponteiro do nó posterior ao corrente aponta para o anterior.
- Ponteiro para o nó posterior é nulo.

AI 3:

- Ponteiro do nó corrente é nulo.

Argumentação de seleção:**Algoritmo 2: Condição de ordenação**

```

AE →
  INÍCIO
  |   SE LISTA VAZIA ENTÃO "NÃO É POSSÍVEL ORDENAR"
  |   SENÃO ORDENA LISTA ALFABETICAMENTE
  FIM
AS →

```

AE:

- Valem as assertivas...
- Lista pode estar vazia ou não.

AS:

- Valem as assertivas...
- Lista está ordenada ou mensagem foi apresentada.

① $AE \ \&\& \ (Condição == True) \oplus B1 \implies AS$

Pela AE, lista pode estar vazia. Como $(Condição == True)$, lista está vazia. Neste caso, executa B1 que apresenta a mensagem "não é possível ordenar", valendo a AS.

$$\textcircled{2} \text{ AE } \&\& (\text{Condição} == \text{False}) \textcircled{+} \text{ B2} \implies \text{AS}$$

Pela AE, lista pode não estar vazia. Como (Condição == False), lista possui pelo menos um elemento. Neste caso, executa B2 que ordena a lista. Vale a AS pois a lista termina ordenada.

Argumentação de repetição:

Tudo gira em torno do estado e do descritor de estado.

AE: Assertiva de entrada.

AS: Assertiva de saída.

AINV: Assertiva invariante. Válida a cada ciclo da repetição e relacionada ao descritor de estado.

$$\textcircled{1} \text{ AE} \implies \text{AINV}$$

$$\textcircled{2} \text{ AE } \&\& (\text{Condição} == \text{False}) \implies \text{AS}$$

(Condição == False) significa que não entrou dentro do bloco ou não concluiu o primeiro ciclo.

$$\textcircled{3} \text{ AE } \&\& (\text{Condição} == \text{True}) \textcircled{+} \text{ B} \implies \text{AINV}$$

Primeiro ciclo.

$$\textcircled{4} \text{ AINV } \&\& (\text{Condição} == \text{True}) \textcircled{+} \text{ B} \implies \text{AINV}$$

Demais ciclos

$$\textcircled{5} \text{ AINV } \&\& (\text{Condição} == \text{False}) \textcircled{+} \text{ B} \implies \text{AS}$$

Último ciclo.

$$\textcircled{6} \text{ Término}$$