

PROBLEMA DA PARADA E EXEMPLOS

O Problema da Parada é um problema de decisão. Existem problemas para os quais a máquina não deve parar, como o cálculo de todos dígitos de π . Pode-se dizer que o que define o Problema da Parada é a capacidade de olhar para a máquina e determinar se algum dia ela vai parar.

Quando o loop termina, o invariante nos fornece uma propriedade útil que ajuda a mostrar que o algoritmo é correto. Ou seja, serve para provar, por indução, que o processo iterativo está correto.

Exemplo 1)

Invariantes devem ser *estáticos, parados no tempo*. O invariante correto para o processo iterativo for da figura abaixo: no início de cada iteração, x é um elemento máximo de $v[0..j-1]$. O início de cada iteração fica *imediatamente antes* do ponto em que o código decide se o processo iterativo deve parar ou continuar.

```
public int max(int[] v) {  
    int n = v.length;  
    int j, x = v[0];  
    for (j = 1; j < n; j++)  
        if (x < v[j]) x = v[j];  
    return x;  
}
```

Exemplo 2)

A figura abaixo mostra um loop que armazena na variável *s* a soma dos elementos do vetor *b*[0..*n*—1]. O loop está comentado de forma que a asserção *inv* descreve os valores de *s* e *i* antes e depois de cada iteração.

A asserção *inv* é chamada de invariante do loop porque ela é invariavelmente verdadeira antes e depois de cada iteração. A função teto *ft* é uma garantia de que o loop irá terminar.

```
1  /* pre:
2    * b[0..n—1], onde n ≥ 0;
3    * (n = 0 => b[0..n—1], vetor vazio)
4    */
5    i = 0; s = 0;
6    /* inv:
7      * i = 0;
8      * s é a soma de b[0..n—1];
9      * n iterações a fazer
10   */
11   while( i != n )
12   {
13     /* inv:
14       * 0 ≤ i < n;
15       * s é a soma de b[0..i—1];
16       * n—i iterações a fazer
17     */
18     s += b[i++];
19   }
20   /* inv = pos:
21     * i = n;
22     * s é a soma de b[0..n—1];
23     * 0 iterações a fazer
24   */
```

Referências

Kernighan, B.W.; Ritchie, D.M., The C Programming Language, Prentice-Hall, Second Edition, 1988.

Observações Sobre Invariantes. [Acesso em 20 de Julho].

<http://www.ime.usp.br/~pf/estruturas-de-dados/aulas/invariants.html>

Projeto e Análise de Algoritmos. Universidade Federal de Santa Maria. [Acesso em 20 de Julho]. <http://www-usr.inf.ufsm.br/~juvizzotto/elc120-2016b/aula1.pdf>