Teoria da Computação

Programas Recursivos

Cristiano Lehrer, M.Sc.

ybadoo

Introdução

- O terceiro tipo de estrutura de controle é encontrado, com diversas variações, na maioria das linguagens de alto nível que admite a definição de sub-rotinas recursivas.
- Recursão é uma forma indutiva de definir programas.
- Sub-rotinas permitem a estruturação hierárquica de programas, possibilitando níveis diferenciados de abstração.

Definição (1/2)

- Uma Expressão de Sub-rotinas é indutivamente definida como:
  - A operação vazia constitui uma expressão de sub-rotinas;
  - Cada identificador de operação constitui uma expressão de sub-rotinas;
  - Cada identificador de sub-rotina constitui uma expressão de sub-rotinas;
  - Composição sequencial:
    - Se `D1` e `D2` são expressões de sub-rotinas, então a composição seqüencial denotada por `D1; D2` resulta em uma expressão de sub-rotinas cujo efeito é a execução de `D1` e, após, a execução de `D2`.
  - Composição condicional:
    - Se `D1` e `D2` são expressões de sub-rotinas e `T` é um identificador de teste, então a composição condicional denotada por `se T então D1 senão D2` resulta em uma expressão de sub-rotinas cujo efeito é a execução de `D1` se `T` é verdadeiro ou `D2` se `T` é falso.

Definição (2/2)

- Um Programa Recursivo P tem a seguinte forma:
  - P é E_0 onde R_1 def E_1, R_2 def E_2, ..., R_n def E_n onde (suponha $k \in \{1, 2, ..., n\}$):
    - E_0 Expressão Inicial a qual é uma expressão de sub-rotinas;
    - E_k Expressão de Define R_k, ou seja, a expressão que define a sub-rotina identificada por R_k.

Exemplo - Enunciado

Desenvolver um programa recursivo, sobre uma máquina genérica, que calcule o fatorial de n utilizando a fórmula

```
f = 1 * 2 * 3 * 4 * ... * n
```

- O valor de n será fornecido pelo usuário, devendo ser um valor inteiro e positivo.
- Por exemplo, caso o valor fornecido pelo usuário para n seja `5`, o programa deverá apresentar como resposta o valor `120`, ou seja, `1 * 2 * 3 * 4 * 5`.
- Caso o usuário forneça um valor inválido para n, o programa deverá apresentar uma mensagem de erro.

Exemplo - Código

```
função fatorial(numero)
  se (numero > 1) então
    retornar numero * fatorial(numero – 1);
  senão
    retornar 1;
fim função;

programa
  ler(numero);
  se (numero > 0) então
    escrever(fatorial(numero));
  senão
    escrever(erro);
  fim se;
fim programa;
```

Exemplo - Execução

```
função fatorial(numero)
  se (numero > 0) então
    retornar numero * fatorial(numero – 1);
  senão
    retornar 1;
fim função;

programa
  ler(numero);
  se (numero > 0) então
    escrever(fatorial(numero));
  senão
    escrever(erro);
  fim se;
fim programa;
```

```
numero = 5

fatorial (5) = 5 * fatorial(4)

fatorial (4) = 4 * fatorial(3)

fatorial (3) = 3 * fatorial(3)

fatorial (2) = 2 * fatorial(1)

fatorial (1) = 1 * fatorial(0)

fatorial (0) = 1
```

Exemplo - Código em C

```c
#include <stdio.h>

long int fatorial (int n) {
  if (n > 1)  {
    return n * fatorial(n - 1);
  }
  else  {
    return 1;
  }
}
```

```c
int main() {
  int n;
  printf("Forneca o valor de n: ");
  scanf("%d", &n);
  if (n >= 0)  {
    printf("\nO fatorial de %d eh %ld\n", n, fatorial(n));
  }
  else  {
    printf("\n'%d' nao eh um valor valido para n\n", n);
  }
  return 0;
}
```


ybadoo