RECURSIVIDADE

Frank de Alcantara - 2022

RECORRÊNCIA E RECURSIVIDADE

Uma relação recorrente é uma equação que define uma sequência baseada em uma regra que determina o próximo termo como função de termos anteriores

Uma função recursiva é uma função que chama a si mesma.

Uma relação recorrente usa a recursão para criar uma sequência.

SEQUÊNCIA DEFINIDA POR RECORRÊNCIA

UMA SEQUÊNCIA INFINITA S SERÁ UMA LISTA DE ARTEFATOS INDEXADOS EM UMA DETERMINADA ORDEM. TAL QUE:

$$S_1, S_2, S_3, \ldots S_k$$

UMA SEQUÊNCIA PODE SER DEFINIDA POR RECORRÊNCIA SE CONHECEMOS ALGUNS DOS VALORES INICIAIS E A EQUAÇÃO DE RECORRÊNCIA.

RECORRÊNCIA (EXEMPLO 1)

EXEMPLO 1

Considere a sequência S definida por recorrência por

1. $S(1) = 2$
2. $S(n) = 2S(n - 1)$ para $n \geq 2$

Pela proposição 1, $S(1)$, o primeiro objeto em S, é 2. Depois, pela proposição 2, o segundo objeto em S é $S(2) = 2S(1) = 2(2) = 4$. Novamente pela proposição 2, $S(3) = 2S(2) = 2(4) = 8$. Continuando desse modo, vemos que a sequência S é

$$2, 4, 8, 16, 32, \ldots$$

Baseado em Gersting (2017, p. 134)

EXERCÍCIO

A SEQUÊNCIA DE FIBONACCI, F, É DADA POR:

$$F_1 = 1$$

$$F_2 = 1$$

$$F_n = F_{n-2} + F_{n-1} \; para \; n > 2$$

ESCREVA OS OITO PRIMEIROS ITENS DA SEQUÊNCIA DE FIBONACCI.

CONJUNTOS E RECORRÊNCIA

Conjuntos são primitivos matemáticos onde não existem repetições ou ordens explicitadas.

Conjuntos também podem ser definidos de forma recorrente.

RECORRÊNCIA (EXEMPLO 2)

EXEMPLO 5

Na Seção 1.1, notamos que certas cadeias de letras de proposição, conectivos lógicos e parênteses, tais como $(A \wedge B)' \vee C$, são consideradas legítimas, enquanto outras, como $\wedge\wedge A'' B$, não o são. A sintaxe para arrumar tais símbolos constitui a definição do conjunto de fórmulas proposicionais bem formuladas e é uma definição por recorrência.

1. Qualquer letra de proposição é uma fbf.
2. Se P e Q são fbfs, então $(P \wedge Q)$, $(P \vee Q)$, $(P \rightarrow Q)$, (P'), e $(P \leftrightarrow Q)$ também o são.[2]

Usando as prioridades para os conectivos lógicos estabelecidas na Seção 1.1, podemos omitir os parênteses quando isso não causar confusão. Assim, podemos escrever $(P \vee Q)$ como $P \vee Q$, ou (P') como P'; as novas expressões, tecnicamente, não são fbfs pela definição que acabamos de dar, mas representam, sem ambiguidades, fbfs.

Começando com letras de proposição e usando, repetidamente, a regra 2, podemos construir todas as fbfs proposicionais. Por exemplo, A, B e C são fbfs pela regra 1. Pela regra 2,

$$(A \wedge B) \text{ e } (C')$$

são, ambas, fbfs. Novamente pela regra 2,

$$((A \wedge B) \rightarrow (C'))$$

é uma fbf. Aplicando a regra 2 mais uma vez, obtemos a fbf

$$(((A \wedge B) \rightarrow (C))')$$

Eliminando alguns parênteses, podemos escrever essa fbf como

$$((A \wedge B) \rightarrow C')' \quad \bullet$$

RECORRÊNCIA (EXEMPLO 2)

O CONJUNTO DE TODOS OS *STRINGS* POSSÍVEIS DE UM ALFABETO Σ É INDICADO POR Σ^*. A DEFINIÇÃO RECORRENTE DE Σ^* É DADA POR:

1. O STRING VAZIO ε PERTENCE A Σ^*;
2. UM ELEMENTO $s \in \Sigma$ ENTÃO $s \in \Sigma^*$;
3. SE s E t SÃO *STRINGS* EM Σ^* ENTÃO A CONCATENAÇÃO $st \in \Sigma^*$.

AS REGRAS 1 E 2 DEFINEM OS CASOS BASE DESTA DEFINIÇÃO ENQUANTO A REGRA TRÊS DEFINE O CASO INDUTIVO DE TAL FORMA QUE PARA QUALQUER $s \in \Sigma^*$ TEREMOS $s\varepsilon = \varepsilon s = s$

Baseado em Gersting (2017, p. 134)

RECORRÊNCIA (EXEMPLO 3)

EXEMPLO 9

Uma definição recorrente da operação de potenciação a^n de um número real não nulo a, em que n é um inteiro não negativo, é

1. $a_0 = 1$
2. $a_n = (a^n - 1)a$ para $n \geq 1$

EXEMPLO 10

Uma definição recorrente para a multiplicação de dois inteiros positivos m e n.

1. $m(1) = m$
2. $m(n) = m(n) - 1) + m$ para $n \geq 2$

Baseado em Gersting (2017, p. 134)

ALGORITMOS

Algoritmo Interativo

```
S(inteiro positivo n)
//função que calcula iterativamente o valor S(n)
//para a sequência S do Exemplo 1
Variáveis locais:
inteiro i      //índice do laço
ValorAtual     //valor atual da função S

   se n = 1 então
      retorne 2
   senão
      i = 2
      ValorAtual = 2
      enquanto i < = n faça
      ValorAtual = 2 * ValorAtual
      i = i + 1
   fim do enquanto

  //agora ValorAtual tem o valor S(n)
  retorne ValorAtual
fim do se
fim da função S
```

Retirado do exemplo 1

1. $S(1) = 2$
2. $S(n) = 2S(n-1)$ para $n \geq 2$

Algoritmo Recursivo

```
S(inteiro positivo n)
//função que calcula o valor S(n) de forma recorrente
//para a sequência S do Exemplo 1

   se n = 1 então
      retorne 2
   senão
      retorne 2 * S(n – 1)
   fim do se
fim da função S
```

DEFINIÇÃO DE ALGORITMOS RECURSIVOS

UMA FUNÇÃO RECURSIVA É UMA FUNÇÃO QUE CHAMA A SI MESMO:

1. UM CASO BASE, NÃO RECURSIVO;
2. UM CASO RECURSIVO.

SEMPRE, USE O CASO RECURSIVO COMO A ÚLTIMA DECLARAÇÃO DA FUNÇÃO (*TAIL RECURSION*).

RECURSÃO (EXEMPLO 3)

CONSIDERE:

$$fatorial\ (0) = 1$$

$$fatorial\ (n) = n \times fatorial(n-1)$$

TERÍAMOS:

$$fatorial\ (4) = 4 \times fatorial(3)$$

$$fatorial\ (4) = 4 \times (3 \times fatorial(2))$$

$$fatorial\ (4) = 4 \times (3 \times (2 \times fatorial(1)))$$

$$fatorial\ (4) = 4 \times (3 \times (2 \times (1 \times fatorial(0))))$$

$$fatorial\ (4) = 4 \times \Big(3 \times \Big(2 \times \big(1 \times (1)\big)\Big)\Big)$$

$$fatorial\ (4) = 24$$

USANDO O SITE *REPL.IT*, IMPLEMENTE UMA FUNÇÃO EM PYTHON CAPAZ DE CALCULAR O FATORIAL DE 20 DE FORMA RECURSIVA.

RECURSÃO EXERCÍCIO

ESCREVA, EM PYTHON, UMA FUNÇÃO RECURSIVA CAPAZ DE APRESENTAR OS 20 PRIMEIROS ELEMENTOS DE CADA UMA DAS SEQUÊNCIAS APRESENTADAS NA TABELA A SEGUIR. ALTERNATIVAMENTE, VOCÊ PODE FAZER O ALGORITMO EM PSEUDOCÓDIGO.

Relações de Recorrência	Valores iniciais	Sequência
$F_n = F_{n-1} + F_{n-2}$	$a_1 = a_2 = 1$	Fibonacci number
$F_n = F_{n-1} + F_{n-2}$	$a_1 = 1, a_2 = 3$	Lucas Number
$F_n = F_{n-2} + F_{n-3}$	$a_1 = a_2 = a_3 = 1$	Padovan sequence
$F_n = 2F_{n-1} + F_{n-2}$	$a_1 = 0, a_2 = 1$	Pell number

OBRIGADO

Frank de Alcantara

Exercícios disponíveis no Ambiente Virtual de Aprendizagem