Algoritmos e Estruturas de Dados II
Resumo sobre Somatórios

Matheus Kraisfeld B. de Lima

[1]Instituto de Ciências Exatas e Informática
Pontifícia Universidade Católica de Minas gerais
Caixa Postal 1.686 – 30535-901 – Belo Horizonte – MG – Brasil

{matheuskraisfeld@gmail.com.br}

1. Informações Gerais

Segundo [Graham 1994], somatórios são utilizados para representar soma de termos de uma recorrência, quando os mesmos sao diferentes um do outro. O somatório é representado pela letra grega Σ (Sigma). Temos então $\sum_{i=1}^{n} x_i$, onde i representa o índice inicial ou limite inferior, e n representa o índice superior ou limite superior. Por exemplo:

$$\sum_{i=1}^{10} i = 1 + 2 + 3 + ... + 9 + 10 = 55$$

2. Outra forma de representação

A representação na forma anteriormente exposta pode ser substituída por

$$\sum_{1 \leq i \leq n} x_i$$

3. Propriedades Gerais dos Somatórios

3.1. Lei distributiva:

Tem-se ca+cb = c(a+b). Generalizando obtém-se (com c uma qualquer expressão que não dependa de i):

$$\sum_{i=k}^{n} ca_i = c \sum_{i=k}^{n} a_i$$

e

$$\sum_{i \in k} ca_i = c \sum_{i \in k} a_i$$

3.2. Leis associativas:

3.2.1.

Sendo K_1 e K_2 dois conjuntos finitos de inteiros, temos:

$$\sum_{i \in K_1} a_i + \sum_{i \in K_2} a_i = \sum_{i \in K_1 \cap K_2} a_i + \sum_{i \in K_1 \cup K_2} a_i$$

Casos particulares:

$$\sum_{i \in K_1 \cup K_2} a_i = \sum_{i \in K_1} a_i + \sum_{i \in K_2} a_i, \; se K_1 \cup K_2 = \oslash$$

$$\sum_{i=K}^{n} a_i = \sum_{i=K}^{j} a_i + \sum_{i=j+1}^{n} b_i \; (com K \leq j \leq n)$$

3.2.2.

$$\sum_{i=k}^{n}(a_i+b_i) = a_1+b_1+\ddots+a_n+b_n = a_1+\ddots+a_n+b_1+\ddots+b_n = \sum_{i=k}^{n}a_i+\sum_{i=k}^{n}b_i$$

Mais geralmente:

$$\sum_{i\in k}(a_i+b_i) = \sum_{i\in k}a_i+\sum_{i\in k}b_i$$

3.3. Lei comutativa:

Sendo ω uma qualquer permutação do conjunto dos inteiros, temos:

$$\sum_{i\in K}a_i = \sum_{\omega(i)\in K}a_{\omega(i)}$$

Casos particulares de interesse (com K, n $\in \mathcal{Z}$ e K $\leq$ n):
· Considerando a transformação ω(i) = i + c (com c uma constante inteira), obtém-se:

$$\sum_{i=k}^{n}a_i = \sum_{k\leq i\leq n} = a_i = \sum_{k\leq\omega(i)\leq n}a_{\omega(i)} = \sum_{k\leq i+c\leq n}a_{i+c} = \sum_{k-c\leq i\leq n-c}a_{i+c} = \sum_{i=k-c}^{n-c}a_{i+c}$$

i. e.

$$\sum_{i=k}^{n}a_i = \sum_{i=k-c}^{n-c}a_{i+c}(=\sum_{i=\omega^{-1}(k)}^{\omega^{-1}(n)}a_{\omega(i)})$$

· E considerando a transformação ω(i) = c - i (com c uma constante inteira), obtém-se:

$$\sum_{i=k}^{n}a_i = \sum_{i=c-n}^{c-k}a_{c-i}(=\sum_{i=\omega^{1}(n)}^{\omega^{-1}(k)}a_{\omega(i)})$$

A lei distributiva permite mover constantes para dentro e fora de um somatório. As leis associativas permitem dividir um somatório em dois ou vice-versa, e a lei comutativa permite que podemos reordenar os termos para somar em qualquer ordem. Entende-se por ”resolver um somatório” encontrar uma expressão equivalente que possa ser facilmente resolvida em ambiente computacional, sem necessidade de recursividade ou ciclos. Essas formas são chamadas de closed forms (toda forma que não possui ”...”). As propriedades permitem trabalhar os somatórios para alcançar uma closed form.

4. Referências

References

Graham, R. L. (1994). Concrete mathematics: a foundation for computer science.