Conversões de Bases

Paulo Ricardo Lisboa de Almeida

2021

Conteúdo da Aula

- Conversão de bases para inteiros e fracionários

1 Conversão de uma base β para base 10

Teste Você Mesmo

Nós estamos acostumados com a **base 10**, onde os algarismos válidos vão de 0 a 9 (0,1,2, ..., 9). Por quê? Poderíamos escolher, por exemplo, 8 algarismos (0,1,2,...,7), não?

Tente encontrar pelo menos uma base numérica que utilizamos no nosso dia a dia que é diferente da base 10.

O conjunto de algarismos válidos é dado de acordo com a base que estamos trabalhando. Para a base 10 (do nosso dia a dia), os algarismos são 0,1,2,...,9. E para as bases 8? 5? 2? De maneira geral, dada uma base $\beta > 1$ qualquer, onde β é um inteiro, quais são os algarismos válidos para essa base? Tente definir isso, mesmo que informalmente, antes de continuar com a aula.

De maneira geral, dada uma base $\beta > 1$ qualquer, onde β é um inteiro, os algarismos válidos para essa base são $0, 1, \ldots, \beta - 1$. Por exemplo, para a base 10, $\beta = 10$, e os algarismos válidos são $(0, \ldots, \beta - 1) = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$. Para a base 4, os algarismos são $0, 1, 2, 3$.

1.1 Base β para Decimal – Inteiros

Considere os números 347_{10} e 10111_2, onde os números em **subscrito indicam as bases dos números**. A base 2 tem um nome especial, sendo a base **binária**. Os números podem ser escritos de maneira polinomial:

$347_{10} = 3 \times 10^2 + 4 \times 10^1 + 7 \times 10^0$

$10110_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 22_{10}$

Um número em uma base β, representado por $(a_j, a_{j-1}, ..., a_2, a_1, a_0)_\beta$, $0 \leq a_k \leq (\beta - 1)$, $k = 1, ..., j$, pode ser escrito na forma polinomial:

$a_j\beta^j + a_{j-1}\beta^{j-1} + ... + a_2\beta^2 + a_1\beta^1 + a_0\beta^0$

Através do polinômio, podemos converter um número de uma base qualquer para a base decimal. Exemplo:

$10111_2 = 1\times2^4+0\times2^3+1\times2^2+1\times2^1+1\times2^0 = 1\times16+0\times8+1\times4+1\times2+1\times1 = 23_{10}$

1.2 Base β para Decimal – Fracionários

Considere agora um número que contém uma parte fracionária, como $243,51_{10}$. Converter esse número em sua forma polinomial segue o mesmo raciocínio anterior, sendo que os números após a vírgula recebem expoentes negativos, começando pelo -1. Ao resolver o polinômio, mais uma vez, temos o número convertido para a base 10.

Exemplos:

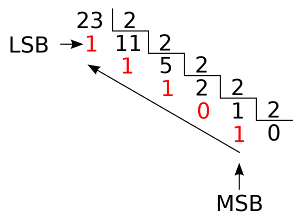
$243,51_{10} = 2\times10^2 + 4\times10^1 + 3\times10^0 + 5\times10^{-1} + 1\times10^{-2} = 243,51_{10}$

$0.001_2 = 0\times2^0 + 0\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} = 0 + 0 + 0 + 0.125 = 0.125_{10}$

2 Conversão da base 10 para uma base β

2.1 Conversão de inteiros

Para a conversão de decimal para binário, realizamos **sucessivas divisões inteiras por 2**, tomando sempre o **resto das divisões como o valor do número na base binária**. Tome como exemplo a conversão do número 23_{10} para binário.



LSB (Least Significant Bit): O bit menos significativo

MSB (Most Significant Bit): O bit mais significativo

Logo, $23_{10} = 10111_2$

De modo geral, ao converter um número da base 10 para uma base β, realizamos sucessivas divisões inteiras, onde o divisor é igual a β e os restos das divisões são os valores convertidos.

2.2 Decimal Fracionário para outras bases

Para converter um número r_{10}, entre 0 e 1 do sistema decimal para um número binário r_2 representado por $(0, d_1d_2...d_j)$ utiliza-se o Algoritmo 1.

Note que o algoritmo só está multiplicando o valor por 2 a cada iteração, e removendo a parte inteira. O algoritmo termina quando o resultado é zero.

Para estender o algoritmo para uma base β qualquer, basta realizar as multiplicações por β ao invés de 2.

Exemplo:

Converter 0.125_{10} para binário

Algorithm 1 Conversão Fracionária $N_{10} \rightarrow N_2$

$k = 1, F = r_{10}$
Faça:
    $F = 2 \times F$
    $d_k = parteInteira(F)$
    $F = F - d_k$
    $k = k + 1$
Enquanto $(F > 0)$

k	F	$d_1 d_2 d_3 d_4 \dots$
1	0.125	0
	0.25	
2	0.25	
	0.5	00
3	0.5	
	1.0	001
4	0.0	**001**

Logo, $0.125_{10} = (0.d_1 d_2 d_3)_2 = 0.001_2$

Teste Você Mesmo

Transforme 0.1_{10} para binário. O que acontece?

Isso explica erro numérico gerado pelo seu computador ao somar 0.1 múltiplas vezes, exemplificado na seção "Teste você mesmo" da aula inaugural da disciplina (o computador utiliza a base binária para realizar operações).

Você precisa estar ciente dos problemas gerados por conversões de base quando, por exemplo, utiliza um computador para resolver determinado problema.

Para converter um valor da base 10, que possui uma parte inteira i e uma parte fracionária f, converta a parte inteira i utilizando divisões sucessivas, e a parte fracionária f utilizando o Algoritmo 1. O resultado final vai ser a junção das duas conversões.

Exemplo: $4,125_{10}$ convertido para a base 2:

Convertendo a parte inteira (4) utilizando sucessões sucessivas, temos 100_2.

Convertendo a parte fracionária (0.125) utilizando o Algoritmo 1, temos 0.001_2.

Logo, $4,125_{10} = 100.001_2$.

Dica

Para transformar um número de uma base $\alpha \neq 10$ para uma base $\beta \neq 10$, primeiro converta para a base 10. A base 10 serve como intermediária, para que você possa converter para β. Sem isso você precisaria, por exemplo, resolver polinômios utilizando a base β, o que pode ser trabalhoso e complicado para nós humanos.

Exercícios

1. Converta os seguintes números para decimal (base 10):

a) 1_2
b) 1000_2
c) 1101101_2
d) 10_8
e) 736_8
f) $0,101_2$
g) $111,012$
h) $123,128$

2. Converta os seguintes números da base decimal para as bases especificadas

a) 251_{10} para base 2
b) 128_{10} para base 2
c) 143_{10} para base 8

3) Converta os valores para binário

a)$0,375_{10}$

b)$0,25_{10}$

c)$47,1217_{10}$

d)$255,59375_{10}$

4) Desafio: Escreva um programa em uma linguagem de sua preferência para converter valores inteiros positivos da base 10 para uma base especificada pelo usuário. Considere apenas bases entre 2 e 9 para simplificar.

3 Licença

Referências

PATTERSON, D.; HENNESSY, J. *Organização e Projeto de Computadores, 5ª Edição: Interface Hardware / Software*. [S.l.]: Elsevier Brasil, 2017.

RUGGIERO, M.; LOPES, V. da R. *Cálculo numérico: aspectos teóricos e computacionais*. [S.l.]: Makron Books do Brasil, 1996.

SANCHES, I.; FURLAN, D. C. C. *Métodos Numéricos*. [S.l.]: UFPR, 2007.