Trabalho Final GCC-108 - Teoria da Computação

Prof.: Douglas H. S. Abreu

Nome: Felipe de Oliveira Fernandes e Jonathas Luis de Sousa

Turma: 10A

Link do repositório [GitHub (https://github.com/FelipeOFernandes/Trabalho_Final_GCC108)](https://github.com/FelipeOFernandes/Trabalho_Final_GCC108)

- O trabalho deve ser feito em grupos de no máximo 2 componentes
- Trabalhos entregues após a data limite não serão aceitos
- Data limite de entrega: 29 de Abril de 2022 : 23h59m
- Enviar o trabalho para o campus virtual, do seguinte modo: Notebook exportado em PDF contendo o código e também o link do repositório GitHub para acesso aos arquivos. A Documentação deve estar no readme
- O trabalho deve ser desenvolvido no modelo Notebook utilizando a linguagem Python

---

Introdução

Este trabalho propõe a utilização de operações da aritmética computacional por meio de uma Máquina de Turing. A máquina que foi desenvolvida recebe como entrada dois números em binário e gera como saída o resultado da adição desses números.

Números binários e adição em números binários

Os números binários são utilizados para representar dados em um meio digital, como por exemplo, a representação no meio analógico com presença ou ausência de carga elétrica e no meio digital por meio de zeros e uns. Essa representação com dois símbolos utiliza-se da mesma técnica do telégrafo, que transmitia mensagens por código Morse, sendo os símbolos curto e longo análogos ao zero e um (1).

Utilizando-se a notação binária é possível representar uma faixa de valores diferentes de acordo com a quantidade de bits. Por exemplo, com dois bits pode-se representar quatro valores distintos, sendo eles 00, 01, 10 e 11. Ou seja, com n bits, podemos representar 2n valores distintos.

Para a notação de números inteiros usando a base binária de zeros e uns, podemos representar os números utilizando as seguintes representações: de binário puro, de binários em sinal magnitude e a representação em complemento de 2 (1).

Tomando como base a representação de números inteiros na base binária pura, que também é a representação utilizada neste trabalho, pode-se observar na Tabela 1, que com quatro bits temos as seguintes possibilidades para números inteiros.

Binário	Decimal	Binário	Decimal
0000	0	1000	8
0001	1	1001	9
0010	2	1010	10
0011	3	1011	11
0100	4	1100	12
0101	5	1101	13
0110	6	1110	14
0111	7	1111	15

Tabela 1. Comparação da base binário de 4 bits com base decimal

As operações matemáticas de adição e subtração feitas na base binária seguem as mesmas regras da base decimal, contando com a diferença que temos apenas dois dígitos. Para a adição de dois números, temos quatro possibilidades de valores, sendo elas: 0 + 0, 0 + 1, 1 + 0, e 1 + 1. As três primeiras têm os mesmos resultados de uma operação em decimal, já para a operação de 1 + 1 temos como resultado zero, gerando um “vai um” para a coluna da esquerda (1).

Máquina de Turing

Turing descreve um computador digital como sendo formado por: uma unidade de armazenamento, uma unidade de execução e uma unidade de controle. A unidade de armazenamento é formada por uma fita, dividida em células, com um cabeçote apontando para a célula atual, a qual pode ser lida/escrita de acordo com a unidade de execução. Por sua vez, a unidade de execução tem como objetivo fazer a leitura do caractere representado na célula atual, analisar o que deve ser feito e alterar quando necessário. Já a unidade de controle faz as movimentações do cabeçote de acordo com o que a unidade de execução deseja, movendo o cabeçote para esquerda ou direita (2).

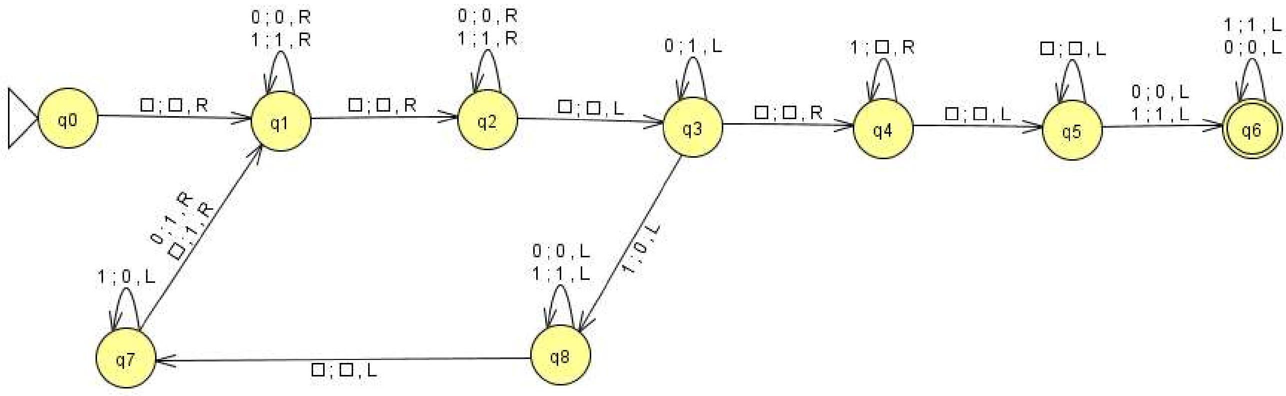
Exercício 1)

Descreva com suas palavras uma estratégia para o desenvolvimento de uma maquina de Turing que compute a soma de 2 numeros binário.

R: A estratégia é tirar uma unidade do segundo numero e adicionar ao primeiro a cada ciclo. Tendo a representação binaria de dois números na fita, com esses números separados por □, move-se a cabeça de leitura para a direita até o final da fita (ultimo digito do segundo número) e em seguida retornando para a esquerda diminuindo uma unidade do segundo número e ao chegar ao primeiro numero somar uma unidade. Esse processo se repete até a representação binaria for segundo número ser formada apenas por zeros;

Exercício 2)

Faça o esboço por meio de desenho da máquina de Turing proposta.



Exercício 3)

Defina a MT como uma quíntupla M=(Q,Σ,Γ,δ,q_0):

```
Q = conjunto de estados (padrão q[0-9]+)

Σ = alfabeto de entrada

Γ = alfabeto da fita

δ = função de transição no formato (q_i,x)→(q_j,y,D); assim, estando no estado q_i, lendo x, vai para o estado q_j, escreve y e movimenta na direção de D. D será L para esquerda ou R para direita.

q_0 = estado inicial
```

R:
M=(Q,Σ,Γ,δ,q0)
Q={q0, q1, q2, q3, q4, q5, q6, q7, q8}
Σ={0,1,B}
Γ={1,,0,B}

δ(q_0,B)=(q_1,B,R),

δ(q_1,B)=(q_2,B,R),
δ(q_1,1)=(q_1,1,R),
δ(q_1,0)=(q_1,0,R),
δ(q_2,1)=(q_2,1,R),
δ(q_2,0)=(q_2,0,R),
δ(q_2,B)=(q_3,B,L),
δ(q_3,B)=(q_4,B,R),
δ(q_3,0)=(q_3,1,L),
δ(q_3,1)=(q_8,0,L),
δ(q_4,1)=(q_4,B,R),
δ(q_4,B)=(q_5,B,L),
δ(q_5,B)=(q_5,B,L),

δ(q_5,1)=(q_6,1,L),
δ(q_5,0)=(q_6,0,L),
δ(q_6,0)=(q_6,0,L),
δ(q_6,1)=(q_6,1,L),
δ(q_7,B)=(q_1,1,R),
δ(q_7,0)=(q_1,1,R),
δ(q_7,1)=(q_7,0,L),
δ(q_8,1)=(q_8,1,L),
δ(q_8,0)=(q_8,0,L),
δ(q_8,B)=(q_7,B,L)
q_0 = q_0

Exercício 4)

Faça a conversão de M em R(M)
R:
0001011101101110110011011101110111011001101011010110011011011011001110101110101100111011011
1110111110111011001111011011110101001111010111111111011010011111010111110111011001111101110111111
11010011111101011111110101001111110110111111011010011111101101111110110100111111010111111101
1111110110110101100111111110101111111101101001111111101011111111010100111111110110111111110110
11101000

Exercício 5)

Desenvolva uma função MTU que receba R(M) acrescido de uma entrada w, onde w é um arquivo csv que contem dois números binário. A saída da função MTU deve ser a computação de M para uma entrada w.

Exercício 6)

***A)* Explique a Tese de Chuch-Turing de forma sucinta**
R: A Tese de Chuch-Turing defende que se qualquer função só pode ser computada por algum tipo de algoritmo se, e somente se, ela for computada por uma MT, ou seja, os algoritmos são o que as MT são capazes de fazer. Além disso, subentende-se que se não há MT para resolver um problema, então o problema simplesmente não tem solução computacional!

***B)* Dada uma máquina de Turing arbitrária M e uma string de entrada w, a computação de M com entrada w irá parar em menos de 100 transições? Descreva uma máquina de Turing que resolva esse problema de decisão.**
R: O problema da parada para máquinas de Turing é indecidível, é necessário conhecer a MT, pois apenas a parada de uma máquina M (fixa) para quaquer w é decidível.

C) Motre a solução para cada um dos seguintes sitemas de correspondência de Post:

a) (a, aa), (bb, b), (a, bb)
R: aabbbb

a	a	bb	bb
aa	bb	b	b

b) (a, ab), (ba, aba), (b, aba), (bba, b)
R: abbaba

a	bba	ba		b
ab	b	aba		aba

c) (abb, ab), (aba, ba), (aab,abab)
R: Indecidivel, deve começar com (abb, ab), unico que começam iguais, em seguida utiliza-se (aba, ba), mas não há mais combinações.

d) (ab,aba), (baa, aa), (aba, baa)
R: Indecidivel, deve começar com (ab, aba), unico que começam iguais, em seguida utiliza-se (aba, ba), mas não há mais combinações.

e) (a, aaa), (aab, b), (abaaa, ab)
R: aaab

a	aab		abaaa
aaa	b		ab

f) (ab, bb), (aa, ba), (ab, abb), (bb, bab)
R: Indecidivel, deve começar com (ab, abb), unico que começam iguais, em seguida utiliza-se (bb, bab), mas não há mais combinações.

D)

a) Prove que a função é primitiva recursiva

$$f(x_1, \ldots, x_n,) = \overset{u(x_1, \ldots, x_n)}{\mu z} [p(x_1, \ldots, x_n, z)]$$

sempre que *p* e *u* são recursivas primitivas

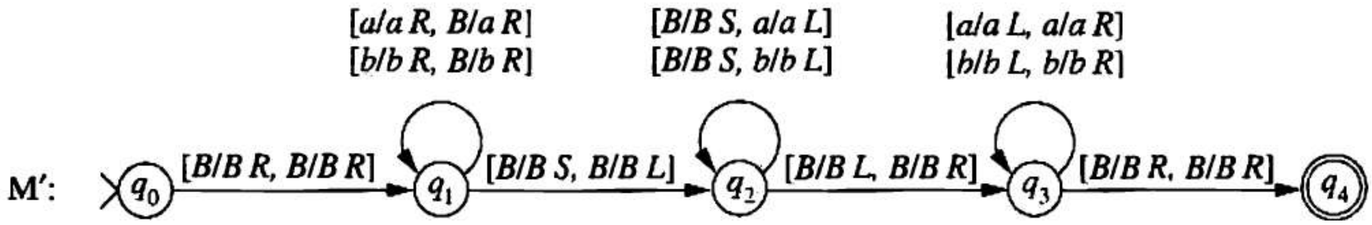
b) defina o valor "passo a passo" de *gn(4,1,0,2,1)* =
R:
gn(4,1,0,2,1) = pn(0)^4 * pn(1)^1 * pn(2)^0 * pn(3)^2 * pn(4)^1
gn(4,1,0,2,1) = 2^4 * 3^1 * 5^0 * 7^2 * 11^1
gn(4,1,0,2,1) = 16 * 3 * 1 * 49 * 11
gn(4,1,0,2,1) = 25872

E)

a) Dado *f(x) = 3x² + 4x + 6* e *g(x) = 5x²*

Prove que *g(x)* ∈ *O(f)* e *f(x)* ∈ *O(g)*
R:
f(x) = 3x² + 4x + 6
g(x) = 5x²
O termo de maior ordem de f(x) é 3x², logo ele existe em O(n²) por existir um termo x'n² > 3x².
O termo de maior ordem de g(x) é 5x², logo ele existe em O(n²) por existir um termo x'n² > 5x².
Como f(x) e g(x) existem em 0(n²),g(x) existe em O(f) e f(x) existe em O(g)

b) Qual é a complexidade e o "*big O"* de M'?



R: tcM(n) = 3 (n + 1) + 1 O(n)

Referências

(1) *Ronald. J. Tocci, Neal. S. Widmer e Gregory L. Moss. 2011. Sistemas Digitais: Princípios e Aplicações. (11ª ed.). Pearson.*

(2) *Alan Turing. 1937. Computability and λ-definability. Journal of Symbolic Logic, 2, 4: 153–163.*

(3) *Sudkamp, T. A. 2006. Languages and machines: an introduction to the theory of computer science. 3rd Edition*
