

Multiplicando números binários com Máquinas de Turing: Interdisciplinaridade no Ensino de Computação

**Maurílio Martins Campano Junior¹, Cinthyan Renata Sachs C. de Barbosa²,
Alan Salvany Felinto², Linnyer Beatrys Ruiz Aylon¹**

¹Grupo de Pesquisa Manna – Jogos. Programa de Pós-Graduação em
Ciência da Computação (PCC) - Departamento de Informática (DIN) –
Universidade Estadual de Maringá (UEM) – 87.020-900 – Maringá – PR – Brazil

²Grupo de Ensino de Desenvolvimento de Jogos e Ambientes Virtuais e
Programa de Pós-Graduação em Ciência da Computação – Departamento de
Computação (DC) – Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 – 86.057-970 – Londrina – PR – Brazil

maurilio.campanojr@gmail.com, {cinthyan,alan}@uel.br, lbruiz@uem.br

***Abstract.** This article describes a work that integrates the teaching practices of two subjects which are: Formal Languages and Automata and also Computer Architecture and Organization by developing a Turing Machine that performs multiplication of binary numbers in magnitude signal. The developed Turing Machine is intended to assist the teaching of Computational Arithmetic, as well as Turing Machine concepts.*

***Resumo.** Este artigo descreve um trabalho que unifica práticas de ensino de duas matérias que são Linguagens Formais e Autômatos e Arquitetura e Organização de Computadores desenvolvendo uma Máquina de Turing que realiza a multiplicação de números binários em sinal magnitude. A máquina de Turing desenvolvida visa auxiliar no ensino de Aritmética Computacional, bem como nos conceitos de Máquina de Turing.*

1. Introdução

O perfil dos estudantes atuais está diferente das gerações anteriores, pois hoje em dia, as gerações Y e Z possuem características que fazem com que os professores necessitem de metodologias diferentes às tradicionais aulas expositivas. Esse perfil de aluno, agitado, inquieto e impaciente, se por um lado pode ser uma dificuldade para o professor, por outro lado temos uma geração que consegue trabalhar com múltiplas tarefas e cada vez mais conectada tecnologicamente [Zucco 2010].

Na área da computação alguns métodos e novas metodologias vêm sendo propostos [Grissom 2013] e uma ideia de trabalho é relacionar conteúdos vistos em disciplinas diferentes em um mesmo trabalho, tendo assim a interdisciplinaridade, apoiada pela *Sociedade Brasileira de Computação* desde 1996 [SBC 2016]. Na interdisciplinaridade mais de uma disciplina se une em um projeto comum, com um planejamento que as relacione. Durante o processo essas áreas trocam conhecimentos e enriquece ainda mais as possibilidades. Como resultado, há um novo saber, menos

fragmentado e mais dinâmico. Essa visão dá significado à experiência escolar [Canal do Ensino 2018].

No âmbito universitário necessitamos construir, definir e avaliar práticas interdisciplinares que forneçam aos nossos alunos experiências de aprendizagem que qualifiquem sua formação profissional [Batista *et al.* 2020]. Uma das áreas que pode se beneficiar da interdisciplinaridade é a área da Teoria da Computação que é formada pela união de ramos da Matemática e da Computação e estuda modelos matemáticos e sua aplicabilidade na Computação [José Neto 2010]. Por se tratar de uma área teórica, os alunos sentem dificuldade em visualizar a importância do conhecimento dos conteúdos e tópicos relacionados com a matéria.

José Neto (2010) alerta sobre a necessidade de resgate das importantes relações entre disciplinas formativas e as de caráter mais pragmático existentes, manifestas nas afinidades entre os modelos abstratos estudados na teoria e as atividades, métodos e técnicas práticas de programação que neles devem sempre se apoiar.

O uso de simuladores para aplicação dos conceitos teóricos da matéria de Linguagens Formais e Autômatos (LFA) [Oliveira e Silva 2007], [Chakraborty, Saxena and Katti 2011], [D'Antoni *et al.* 2015], [Alur *et al.* 2013], [Pereira and Terra 2018], [Lacerda Neto e Terra 2016], [Paul 2015], [Rodger and Finley 2006] pode ser útil como auxílio no aprendizado.

No presente trabalho são apresentados aspectos da evolução conceitual de termos relacionados à prática da interação entre disciplinas (multidisciplinaridade e interdisciplinaridade). O objetivo do trabalho é apresentar uma Máquina de Turing que realiza a multiplicação de números binários utilizando a ferramenta de simulação JFLAP (*Java Formal Languages and Automata Package*) (2018) em uma abordagem multidisciplinar e avaliar o referido artefato em sala de aula

O trabalho está organizado da seguinte forma: a Seção 2 aborda os trabalhos relacionados; a Seção 3 apresenta os fundamentos teóricos; a Seção 4 mostra o desenvolvimento do trabalho; a Seção 5 descreve os resultados obtidos e, por fim, a Seção 6 traz as conclusões do trabalho.

2. Trabalhos relacionados

A seguir serão mostrados em 2.1 os trabalhos que utilizam conceitos de multi, inter ou transdisciplinaridade (embora existam outras denominações e subdivisões desses termos) que se propõem a oferecer alternativas aos modos de pensar e fazer da ciência clássica, disponibilizando, para além do pensamento analítico-reducionista, formas de investigação científica que atendam às necessidades de compreensão de fatos e fenômenos em toda a sua complexidade. Os conceitos associados aos três termos aqui mencionados não são, contudo, únicos ou aceitos com tranquilidade pelos estudiosos. Eles foram assumindo significações diversas ao longo das últimas décadas, mantendo em comum a ideia de que representam movimentos que surgiram em resposta à fragmentação do conhecimento [Bicalho e Oliveira 2011]. Em 2.2 os trabalhos que envolvam simuladores e/ou ferramentas visuais também para o ensino de Computação serão apresentados e em especial o JFLAP.

2.1. Multidisciplinaridade na Computação

A *multidisciplinaridade* ocorre quando há mais de uma área de conhecimento em um determinado projeto ou propósito, mas cada uma dessas disciplinas mantém seus métodos e teorias em perspectiva. Serve para resolver problemas imediatos e não possui foco na articulação e nos ganhos colaborativos [Canal do Ensino 2018].

O trabalho de Massoud *et al.* (2018) tem como objetivo aplicar o ensino multidisciplinar em salas de aula de Computação, visando diferenciar o aprendizado para codificar e o para programar.

Von Wangenheim *et al.* (2017) propuseram um estudo de caso aplicado ao ensino de Computação, de maneira multidisciplinar para 105 crianças estudantes de quinta e sétima série utilizando-se de jogos educativos. Usaram a ferramenta Scratch obtendo resultados que mostram que a multidisciplinariedade é eficiente, efetiva e aumenta o interesse no aprendizado dos estudantes e conseqüentemente sua motivação.

Martins (2016) desenvolveu um trabalho multidisciplinar aplicado ao curso de Sistemas de Informação, no qual os estudantes participavam de um *Desafio de Desenvolvimento de Sistemas Web* (DDSW), permitindo aos participantes a resolução de problemas reais contextualizados inseridos em outras áreas do conhecimento.

Campano Junior *et al.* (2019) apresenta uma Máquina de Turing para realização de operações de soma em binário, na qual a partir de uma palavra em uma MT no formato “<numero_binário1#numero_binário2” é gerado o resultado da soma dos dois números na mesma fita, utilizando-se dos conceitos de Arquitetura de Computadores e Linguagens Formais e Autômatos aplicados em conjunto. De forma similar à Campano Junior (2019), este presente trabalho aumenta a capacidade de computação da máquina para a realização da multiplicação de dois números em binário de 8 bits.

2.2. Uso de Simuladores e Ferramentas Visuais no Ensino de Computação

Na área de Teoria da Computação, vários trabalhos utilizam-se de ferramentas e simuladores como apoio ao ensino [Oliveira e Silva 2007], [Chakraborty, Saxena and Katti 2011], [D'Antoni *et al.* 2015], [Alur *et al.* 2013], [Pereira and Terra 2018], [Lacerda Neto e Terra 2016], [Paul 2015], [Rodger and Finley 2006]. Neste presente trabalho é utilizada a ferramenta de simulação de JFLAP [Paul 2015], [Rodger and Finley 2006] e alguns desses trabalhos citados são descritos a seguir.

Oliveira e Silva (2007) utiliza a construção de simuladores para ensino de MT para o curso de Ciência da Computação usando testes e simulações práticas desses conceitos individualmente e em grupos. Apresentou aumento nas notas da maioria dos alunos que utilizaram do simulador, porém ressalta que mais testes devem ser aplicados com quantidade maior de alunos e melhorias na interface e eficiência do simulador.

Em Chakraborty, Saxena and Katti (2011), os simuladores de autômatos foram classificados em: baseados em linguagens e em visualização. Os *simuladores baseados em linguagens* são definidos a partir de símbolos e linguagens pré-definidas, enquanto os *baseados em visualização* utilizam de uma representação gráfica e visual dos autômatos para o simulador.

O simulador Automata Tutor v 2.0 [D'Antoni *et al.* 2015], [Alur *et al.* 2013] funciona como um simulador web para construção de autômatos finitos determinísticos, autômatos finitos não determinísticos e definição de expressões regulares. Ele permite a resolução de exercícios definidos por um tutor/professor e avalia cada resposta por meio

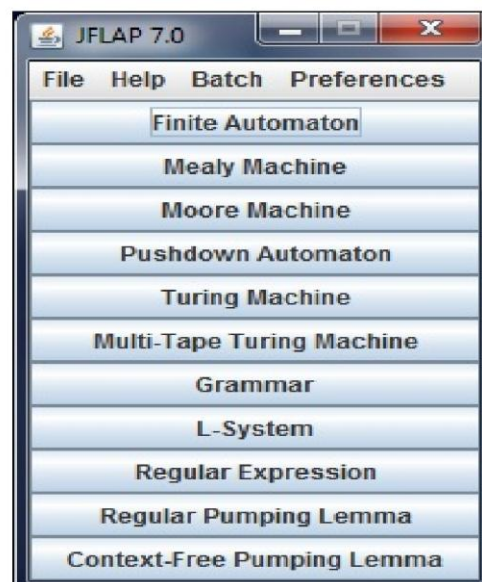
de uma nota de 0 a 10 fornecendo também um *feedback* para auxiliar na resolução correta dos exercícios. No simulador um professor pode criar um curso, definido por um nome e senha, e disponibilizar o acesso aos alunos para resolução de um banco de exercícios definidos pelo docente, permitindo que esse tenha acesso às notas dos alunos após a resolução.

O aplicativo LFApp [Pereira e Terra 2018], [Lacerda Neto e Terra 2016] construído para dispositivos móveis como celulares e *tablets* é projetado para trabalhar com Gramáticas, Autômatos Finitos, Autômatos com Pilha, Máquinas de Turing. Com relação às gramáticas, possibilita sua simplificação por meio das etapas de eliminações de produções vazias, produções unitárias e de símbolos inúteis e também representar gramática nas Formas Normais de Chomsky e Greibach. LFApp pode ser utilizado em sala de aula pelos alunos da disciplina de Linguagens Formais e Autômatos. A tela inicial desse aplicativo pode ser visualizada na Figura 1(a).



(a)

Figura 1. (a) Tela inicial do LFApp



(b)

(b) Tela inicial do JFLAP

3. Fundamentos Teóricos

A Seção 3.1 fala sobre a representação de números binários e a operação de multiplicação de dois números desses e 3.2 sobre a representação de MT.

3.1. Representação de números binários

A representação de números binários é o modo no qual o computador entende as informações e elas podem ser divididas em: binário puro, representação em sinal magnitude, representação em complemento de dois (essas para números inteiros) e a representação de números em ponto flutuante, no padrão da IEEE 754 [Tocci e Widmer 2011], [Idoeta e Capuano 2012].

Neste trabalho foi utilizada a representação de números inteiros em sinal magnitude. Esse método utiliza-se de um bit adicional para representar o sinal do número, sendo utilizado o número *zero* quando temos um número positivo e o *um* quando temos um número negativo e faz com que um número de n bits seja capaz de representar 2^{n-1} números diferentes para os valores positivos e negativos.

A representação em sinal magnitude tem como vantagem a facilidade da operação de negação de um número, bastando apenas trocar o bit de sinal, porém para a realização de adições e subtrações em sinal magnitude deve-se considerar o bit de sinal e a magnitude do número, além do fato de termos duas representações para o número zero [Stallings 2010]: +0 e -0.

As operações matemáticas de adição e subtração em sinal magnitude devem ser feitas analisando separadamente os bits de sinal e de magnitude. Já operações de multiplicação e divisão envolvem operações mais complexas como sucessivas adições, subtrações, deslocamentos lógicos e variáveis auxiliares para resolução.

A implementação da multiplicação binária é realizada por meio de um algoritmo, que executa um ciclo n vezes, sendo esse valor de n dependente da quantidade de bits dos números que se deseja multiplicar. Nesse ciclo são realizadas adições, deslocamentos lógicos até que o valor de n , que é decrementado a cada ciclo, se torne igual à zero. Um fluxograma para a multiplicação de números binários sem sinal pode ser visualizado na Figura 2 (a) [Stallings 2010].

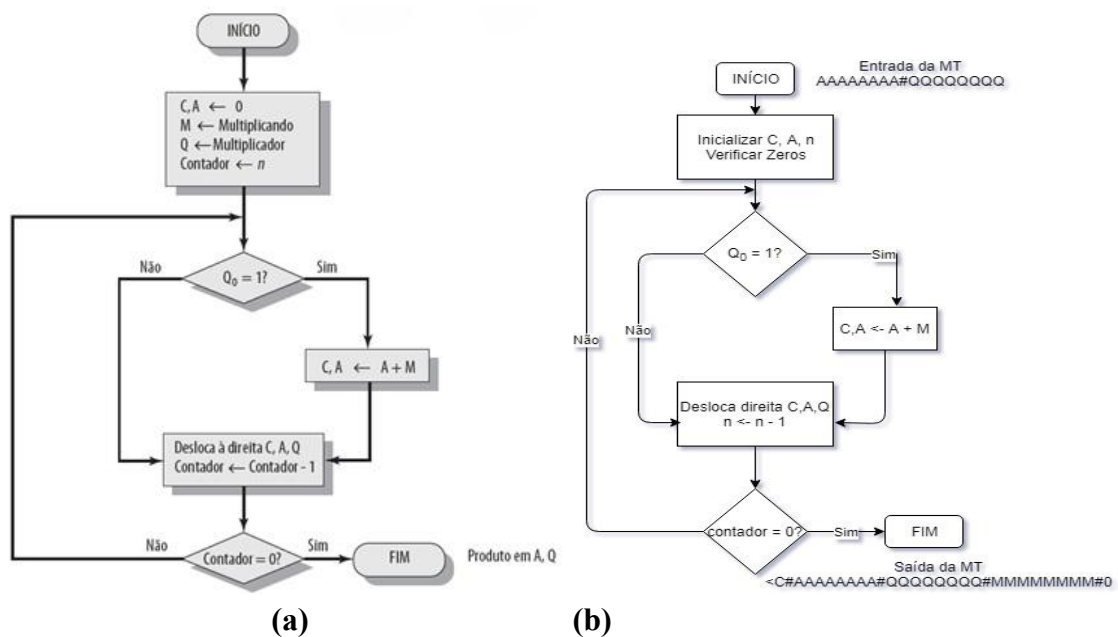


Figura 2. (a) Multiplicação números binários em sinal magnitude (b) Idem utilizando MT

3.2 Máquinas de Turing

Uma Máquina de Turing (MT) é um modelo matemático para formalização de um algoritmo [Vieira 2006], a qual é formada por uma fita infinita à direita e à esquerda. Essa fita é dividida em células, onde é armazenado um símbolo e pode ser lida ou escrita por vez, sendo a célula indicada pelo cabeçote da fita. Esse cabeçote pode se mover para a direita ou esquerda após ler uma célula.

A MT ainda é representada por um conjunto de estados, estado inicial e conjunto de estados finais, símbolos do alfabeto da fita e auxiliar, para identificar uma célula em branco e um que pode ser utilizado para delimitar um início de fita do lado esquerdo, deixando assim somente o lado direito da fita infinito, facilitando o manuseio e resolução de problemas mais simples [Vieira 2006]. O modelo mais utilizado como formalização de algoritmo é provavelmente essa máquina proposta em 1936 por Alan Turing [Diverio e Menezes 2000].

Apesar desses dispositivos serem mais genéricos que os autômatos finitos, sua aparência básica é similar. Contudo, as MTs não são simplesmente mais uma classe de autômatos para serem substituídos mais tarde por um tipo ainda mais poderoso. Existem MTs com muitas fitas ou máquinas com os mais sofisticados dispositivos de memória que podem ser lidos ou gravados em um modelo de acesso aleatório remanescente dos computadores reais; significativamente, esses dispositivos revelam-se como sendo não mais poderosos em termos de poder computacional do que as MTs básicas [Lewis e Papadimitriou 2000].

As MTs podem ser divididas em máquinas reconhecedoras e as máquinas transdutoras. As reconhecedoras são máquinas que recebem como entrada uma palavra e geram como saída resposta Sim ou Não, indicando se a palavra faz parte ou não da linguagem reconhecida pela máquina. Já as máquinas transdutoras recebem como entrada uma palavra e geram como saída uma palavra na própria fita.

Um exemplo de MT Transdutora pode ser visualizado na Figura 3(a), que tem uma MT que recebe como entrada um número binário e gera como saída o seu complemento (transformar os zeros em uns e uns em zero). Uma MT Reconhecedora pode ser visualizada na Figura 3(b). A MT reconhece qualquer palavra para a linguagem $L = \{(\text{)}^n / n > 0\}$. Ou seja, uma linguagem formada por parênteses balanceados, onde para cada parêntese aberto deve ter um para fechar. Assim, se inserirmos uma palavra com parênteses balanceados a MT finalizará em um estado final.

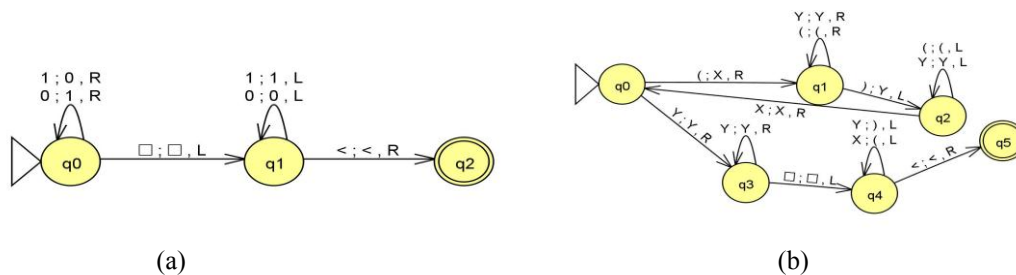


Figura 3. (a) Exemplo de MT Transdutora que gera o complemento binário (b) Exemplo de MT Reconhecedora para a linguagem $L = \{(\text{)}^n / n > 0\}$

4. Desenvolvimento

A implementação deste trabalho foi desenvolvida utilizando a ferramenta JFLAP (2018) com o objetivo de realizar a multiplicação binária sem sinal. O algoritmo realiza a multiplicação de dois números em binários, desconsiderando o bit de sinal. O fluxograma que representa a multiplicação de binários pode ser visto na Figura 2(a), enquanto o para a multiplicação por Máquina de Turing pode ser visto na Figura 2(b).

A multiplicação por MT é realizada com números de no máximo 1 byte ou 8 bits nos quais os números são inseridos no formato Q#M, sendo que Q e M tem 8 bits cada e o marcador “#” representa o fim do primeiro número e o começo do segundo número.

Como primeira etapa do algoritmo construído de multiplicar por MT são inicializados os valores de C (*carry*) e A (*auxiliary*) do fluxograma da Figura 2(a) que são variáveis extras para auxiliar na multiplicação dos números. Essas variáveis são inseridas à esquerda do primeiro número, sendo separadas pelo marcador de # e definido como símbolo marcador de início da fita o “<”. A segunda etapa visa definir o valor de n , que é a quantidade de bits dos números e nesse caso o valor 8. Realizadas essas etapas iniciais, temos que a representação temporária na fita da MT será “<C#A#Q#M#8BBB...” sendo M e Q os números de 8 bits que estão sendo multiplicados, A e C números inicializados com zeros, sendo A com 8 bits e C com 1.

Após a definição dos elementos iniciais, o algoritmo verifica se existem zeros nos números a serem multiplicados, pois caso ocorra temos que o resultado da multiplicação já será zero. Caso ambos os números não sejam zeros, a multiplicação deve ser realizada normalmente. A multiplicação dos números ocorre em um ciclo de n vezes, sendo esse dependente da quantidade de bits dos números que estão sendo multiplicados, no caso deste trabalho temos um $n = 8$.

Para a realização da multiplicação, a cada ciclo do algoritmo, o bit menos significativo do termo Q é analisado. Se esse bit tiver o valor zero é realizado um deslocamento lógico à direita entre os valores de C, A e Q e caso o valor do bit seja igual a um é realizado uma soma entre A e M, e armazenado o resultado em A, sendo utilizado o termo C para um possível *overflow*. Ao final de cada ciclo, o valor de n é decrementado (n é representado em decimal) e quando esse tiver o seu valor igual a zero a multiplicação encerrou-se, sendo o resultado armazenado nos termos de A e Q.

A MT resultante da implementação do algoritmo no JFLAP resultou em 84 estados, sendo um estado inicial e um final. O alfabeto de entrada é formado pelos números binários 0 e 1 e também pelo delimitador “#”. Para alfabeto auxiliar foram utilizados os números decimais de 0 a 8 para delimitar a quantidade de ciclos do algoritmo, o marcador de início da fita “<” e o marcador de símbolos em branco da fita “□”. A quantidade de transições foi superior a 200, sendo sua representação gráfica de difícil análise e compreensão, devido a essa quantidade de estados e transições.

A máquina descrita realiza apenas operações de multiplicação entre dois números de 8 bits, assim se queremos multiplicar 4 vezes 6 devemos representar os números em binário sempre com 8bits, ou seja 00000100 vezes 00000110. Além disso, casos da multiplicação que poderiam gerar um *overflow* também são tratados, como por exemplo, a multiplicação de 11111111 vezes 11111111 que gera como resultado o valor correto 1111111000000001 ($255 * 255 = 65025$).

5. Resultados obtidos

No próprio simulador JFLAP, todas as multiplicações foram efetuadas corretamente, sendo testados os seguintes casos: um dos números é zero, dois números iguais, onde acontece o *overflow* da soma de A e M, entre outros. Um exemplo dos testes pode ser visualizado na Figura 4, na qual a coluna Input indica como foram inseridos os números na MT, a coluna Output indica a saída resultante da MT, no formato <C#A#Q#M#0, sendo que o resultado da multiplicação está em A e Q, e a coluna Result indica o

resultado da Máquina de Turing, se foi aceito (*accept*) ou rejeitada (*reject*) a entrada. Na Figura 4 pode-se observar que: - na 1ª linha temos a multiplicação de 5, 00000101 pelo número zero, 00000000, que resulta no número zero; - na 2ª linha, por sua vez, temos a multiplicação do número 7, 00000111 multiplicado pelo número 6, 00000110 e apresentando o resultado 42, o número binário 00101010;- na 3ª linha temos a multiplicação de 170, 10101010, por 120, 01111000, resultando em 20400, 010011110110000; - na 4ª linha, a multiplicação dos dois maiores números possíveis de representação com 8 bits, 255 vezes 255, resultando em 65025, na representação em binário temos 11111111 vezes 11111111 com um resultado igual a 111111000000001.

Input	Output	Result
00000101#00000000	0#00000000#00000000#00000000#8	Accept
00000111#00000110	0#00000000#00101010#00000110#0	Accept
10101010#01111000	0#01001111#10110000#01111000#0	Accept
11111111#11111111	0#11111110#00000001#11111111#0	Accept

Figura 4. Resultados da multiplicação de números em sinal magnitude utilizando Máquina de Turing no JFLAP

O exercício de MT desenvolvido neste trabalho foi utilizado como exercício durante o ensino da matéria de Linguagens Formais, Autômatos e Computabilidade para o segundo ano do curso de Ciência da Computação da Universidade Estadual do Paraná (UNESPAR) na cidade de Apucarana, enquanto a aplicação da multiplicação de números binários foi utilizada na matéria de Arquitetura e Organização de Computadores (AOC) para a mesma turma e curso.

Em um primeiro momento, aos alunos foram apresentados os conceitos de MT e multiplicação de números binários pelo professor das disciplinas. Esse quem elaborou a máquina no simulador e passou os conceitos da multiplicação na matéria de AOC e da Máquina de Turing na matéria envolvendo LFA para os discentes.

Na sequência, para a matéria de AOC os alunos desenvolveram pseudocódigo para a realização de multiplicação de números em binários e a partir desse, uma representação genérica de estados foi criada visando representar os estados que a Máquina de Turing teria.

Os 28 alunos envolvidos realizaram uma avaliação da interdisciplinaridade entre as duas matérias. Eles tiveram uma aula para testar e validar a máquina no simulador ajudando a procurar erros/problemas com a Máquina de Turing, no entanto, nenhum problema foi encontrado e responderam um questionário com questões objetivas na escala Likert [Alexandre *et al.*, 2003]. Entre as questões que formavam o questionário aplicado aos alunos podemos destacar: qual a importância da inter e multidisciplinaridade no ensino das matérias teóricas de Ciência da Computação, a qual obteve uma média 4,8; já para a questão, a inter e multidisciplinaridade ajudam no entendimento de conceitos teóricos, a média obtida foi de 4,1. Na Figura 5 encontram-se as avaliações gerais desses alunos em relação à multiplicação de números binários utilizando uma Máquina de Turing.

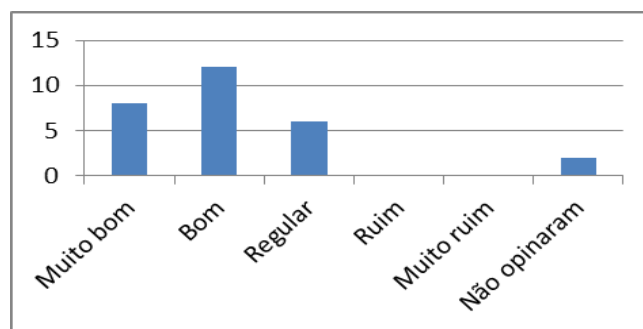


Figura 5. Resultados das avaliações discentes sobre multiplicação de números binários

6. Conclusões

Uma Máquina de Turing representa o poder máximo que pode ser atingido por um computador, a chamada *Hipótese de Church*. Isso significa que em uma MT pode ser realizada qualquer operação de um computador, porém a complexidade de uma operação na Máquina de Turing é consideravelmente grande.

Devido à complexidade de operações com MT, o ensino desse tópico se torna difícil tanto para alunos quanto para docentes. Um dos fatores que pode contribuir com o melhor entendimento desses conceitos é a interdisciplinaridade, aliando termos e técnicas de outras matérias para o ensino das Máquinas de Turing. Essa pode ser vista aqui como a união do ensino da operação de multiplicação de números binários na representação de sinal magnitude com o funcionamento de uma MT Transdutora, além de potencializar reforço dos conceitos de aprendizado de ambas as matérias.

Como trabalhos futuros pretende-se aumentar a capacidade de bits da multiplicação, permitindo números com qualquer quantidade de bits. Almeja-se também inserir outras operações matemáticas, desenvolvendo calculadora na MT, de modo que a entrada será números e a operação desejada e saída o resultado da operação em binário. Além disso, a ideia do trabalho é elaborar agora um pré e pós-testes, antes e depois dos alunos utilizarem a implementação da Máquina de Turing, permitindo assim medir a percepção de aprendizagem dos conteúdos com uso dos conceitos multi disciplinares.

Agradecimentos

O presente trabalho foi realizado com apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) – Bolsista do Brasil (311685/2017-0) e da Fundação Araucária (17.633.124-0).

Referências

- Alexandre, J. W. C., Andrade, D. F., Vasconcelos, A. P., Araujo, A. M. S. e Batista, M. J. (2003). Análise do número de categorias da escala de Likert aplicada à gestão pela qualidade total através da teoria da resposta ao item. *Anais do XXIII Encontro Nacional de Engenharia de Produção*. Ouro Preto, ABEPRO, p.1-8.
- Alur, R., D'Antoni, L., Gulwani, S., Kini, D. and Viswanathan, M. (2013). Automated grading of DFA constructions. *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. Beijing, China, ACM, p.1976-1982.
- Alves, F., Almeida, D., Silva, L. B., Gomes, A. B., Ferreira, R. S. e Nacif, J. A. M. (2015). Ensinando arquiteturas vetoriais utilizando um simulador de instruções

- MIPS. In *International Journal of Computer Architecture Education*, v.4, n.1, pages 9-12, SBC.
- Batista, M. S. S., Oliveira, E. M. S., Silva Bisneta, M. L. S. Bisneta e Tavares, A. G. (2020). Scratch dê vida a sua imaginação: relato de experiência. *Anais do V Congresso Internacional de Educação e Tecnologia*. São Carlos, UFSCAR. p.1-9.
- Bicalho, L. M. e Oliveira, M. (2011). Aspectos conceituais da multidisciplinaridade para interdisciplinaridade e a pesquisa em Ciência da Informação. In *Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação*, v.16, n.32, pages 1-26, BRAPCI.
- Campano Junior, M. M., Faria, C. R., Barbosa, C. R. S. C. e Felinto, A. S. (2019). Um merge entre Máquina de Turing e operações matemáticas em binário no ensino de Linguagens Formais e Autômatos. *Anais do XXIV Congresso Internacional de Informática Educativa. Nuevas Ideas en Informática Educativa*, Volumen 15, Arequipa, Perú, Sánchez, J. (Ed.). p.78-83.
- Canal do Ensino (2018). “Multidisciplinaridade, interdisciplinaridade e transdisciplinaridade: diferenças e convergências”, <https://canaldoensino.com.br/blog/multidisciplinaridade-interdisciplinaridade-e-transdisciplinaridade-diferencas-e-convergencias>, Janeiro.
- Chakraborty, P., Saxena, P. C. and Katti, C. P. (2011). Fifty years of automata simulation: a review. In *ACM Inroads*, v.2, n.4, pages 59-70.
- D'Antoni, L., Weavery, M., Weinert, A. and Alur, R. (2015). Automata Tutor and what we learned from building an online teaching tool. In *Bulletin of the European Association for Theoretical Computer Science*. n.117, pages 1-15. EATCS.
- Diverio, T. A. e Menezes, P. B. (2000). *Teoria da Computação: máquinas universais e computabilidade*. Porto Alegre: Instituto de Informática da UFRGS e Sagra Luzzatto.
- Grissom, S. (2013). Introduction to special issue on alternatives to lecture in the Computer Science Classroom. In *ACM Transactions on Computing Education*. v.13, n.3, pages 1-5. ACM Journals.
- Idoeta, I. V. e Capuano, F. G. (2012). *Elementos de eletrônica digital*, São Paulo: Érica.
- JFLAP (2018). JFlap Home Page. <https://www.jflap.org/>, July.
- José Neto, J. (2010). A Teoria da Computação e o profissional de Informática. In *Revista de Computação e Tecnologia da PUC-SP*. v.1, n.1, pages 4-21. Departamento de Computação/FCET/PUC-SP.
- Lacerda Neto, J. F. e Terra, R. (2016). LFApp: um aplicativo móvel para o ensino de Linguagens Formais e Autômatos. *Anais do XXIV Workshop sobre Educação em Computação*, Porto Alegre, SBC. p.2196-2205.
- Lewis, H. R. e Papadimitriou, C. H. (2000). *Elementos de Teoria da Computação*. Porto Alegre: Bookman.
- Martins, E. R. (2016). Experiência multidisciplinar no curso de Sistemas de Informação (SI) do Instituto Federal de Goiás. In *Revista Técnica*, v.1, pages 51-66.

- Massoud, L., Hallman, S., Plaisent M. and Bernard, P. (2018). Applying multidisciplinary teaching techniques to the Computer Programming/Coding Classroom. In *Journal of Information Technology and Economic Development*, Beverly Hills, v.9, n.1, pages 38-47.
- Oliveira, G. P. e Silva, M. A. S. (2007). Construção de simuladores gráficos para Teoria da Computação: uma proposta para ensino do conceito de Máquinas de Turing. *Anais do IV Simpósio de Excelência em Gestão e Tecnologia*. Resende, AEDB.
- Paul, J. (2015). Using JFLAP to engage students and improve learning of computer science theory: tutorial presentation. In *Journal of Computing Sciences in Colleges*, v.31, n.2, pages 145-148.
- Pereira, C. H. and Terra, R. (2018). A mobile app for teaching formal languages and automata. In *Computer Application in Engineering Education*. v.26, pages 1742–1752.
- Rodger, S. H. and Finley, T. W. (2006). *JFLAP: An Interactive Formal Languages and Automata*, Edited by Jones & Bartlett Publishers, Sudbury, MA.
- SBC-Sociedade Brasileira de Computação (2016)“Computação e Interdisciplinaridade”, *Revista da SBC*. https://www.sbc.org.br/images/flippingbook/computacaobrasil/computa_31/Comp_Brasil_02_2016.pdf, Julho.
- Stallings, W. (2010). *Arquitetura e organização de computadores*, São Paulo: Pearson.
- Tocci, R. J. e Widmer, N. S. (2011). *Sistemas Digitais: Princípios e Aplicações*. Pearson.
- Vieira, N. J. (2006). *Introdução aos Fundamentos da Computação*. São Paulo: Pioneira Thomson Learning.
- Von Wangenheim, C. G., Alves, N. C., Rodrigues, P. E. and Hauck, J. C. (2017). Teaching computing in a multidisciplinary way in social studies classes in school - a case study. In *International Journal of Computer Science Education in Schools*, v.1, pages 3-16.
- Zucco, A. (2010). *Estilos de mobilização profissional de docentes de curso de graduação em Administração*. São Caetano do Sul: Universidade Municipal de São Caetano do Sul. Dissertação de Mestrado.