

Turing to Norma, uma ferramenta para auxiliar o estudo da equivalência entre Máquina de Turing e Máquina Norma

Adriano dos Reis Carvalho¹, Aline Marques Del Valle², Emerson Dias do Prado³ e João Cláudio Dias Araújo⁴

¹Instituto Federal do Sul de Minas – Campus Muzambinho, Muzambinho, MG, arcarvalhov@gmail.com ²Instituto Federal do Sul de Minas – Campus Muzambinho, Muzambinho, MG, aline.valle@eafmuz.gov.br ³Instituto Federal do Sul de Minas – Campus Muzambinho, Muzambinho, MG, edprado6@hotmail.com ⁴Instituto Federal do Sul de Minas – Campus Muzambinho, Muzambinho, MG, jklaus.sd@gmail.com

Introdução

Hoje em dia é possível encontrar diversas ferramentas desenvolvidas para auxiliar o aprendizado dos conceitos da Teoria da computação e dos diversos modelos de dispositivos de computação. Grande parte dessas ferramentas se enquadra na categoria de simuladores de máquinas universais. Também é possível encontrar, com certa dificuldade, algumas ferramentas capazes de fazer a conversão de programas escritos para uma determinada máquina universal em programas para outro tipo de máquina universal. Um exemplo de ferramenta capaz de realizar tal conversão é a ferramenta SIMCO, que aborda conversões apenas entre algumas máquinas universais (ALVES, 2007).

Devido à escassez de ferramentas conversoras de máquinas universais, e visando abordar casos não tratados pela ferramenta encontrada, foi desenvolvida uma ferramenta capaz de converter uma Máquina de Turing em um conjunto de instruções rotuladas de uma Máquina Norma, e a partir de uma sequência de instruções rotuladas de uma Máquina Norma, gerar uma Máquina de Turing, para auxiliar professores e estudantes no estudo dos fundamentos da Teoria da Computação e na prova da equivalência entre máquinas universais.

Material e métodos

Proposta por Alan Mathison Turing (1912 - 1954) em 1936 a Máquina de Turing foi o primeiro modelo de uma máquina universal proposto e é “o modelo computacional mais conhecido e comumente utilizado para formalização de algoritmos (ALVES, 2007).” Segundo Sipser (2011) uma Máquina de Turing é semelhante a um autômato finito, possuindo uma memória ilimitada e irrestrita e pode realizar tudo que um computador real pode fazer. Sua estrutura compreende:

- Fita: dispositivo tanto de entrada quanto de saída, além de servir como memória de trabalho;
- Unidade de leitura e gravação, ou unidade de controle: componente responsável pela leitura e gravação na fita;
- Função de transição ou programa: Define as ações da unidade de controle de acordo com o estado onde ela se encontra e o símbolo lido.

O seu modelo formal é uma 8-tupla, $M = \{\Sigma, Q, \Pi, q_0, F, V, \beta, \odot\}$, onde:

- Σ = alfabeto de símbolos de entrada;
- Q = Conjunto finito de estados possíveis da máquina;
- Π = Programa;
- q_0 = Estado inicial da máquina, $q_0 \in Q$;
- F = Conjunto de estados finais da máquina, $F \subseteq Q$;
- V = Alfabeto auxiliar;
- β = Símbolo branco;
- \odot = Símbolo inicial;

Podemos encontrar variações de Máquinas de Turing, mas nenhuma variação aumenta seu poder computacional.

A Máquina Norma foi proposta em 1976 por Richard Bird e trata-se de uma máquina de registradores. Seu nome é um acrônimo de Number Theoretic Register Machine (Máquina Teórica de Números de Registradores). A arquitetura dos computadores da atualidade lembra a sua forma de definição.

Nela pode-se utilizar um número infinito de registradores, quantos houver necessidade, como memória e sobre cada um dos registradores temos três instruções:

- adição do valor 1 ($A := A + 1$);
- subtração do valor 1 ($A := A - 1$);
- testa se o valor armazenado é zero ($A = 0$).

Estas instruções, aparentemente simples, são capazes de solucionar todos os problemas resolvidos pela Máquina de Turing, apresentando, portanto, o mesmo poder computacional.

Segundo Diverio e Menezes (2011), a prova de que a Máquina de Turing é equivalente a Máquina Norma consiste em duas etapas:

a) “Turing \leq Máquina Norma: a estrutura da máquina de Turing é simulada na Norma usando um arranjo unidimensional” (DIVERIO; MENEZES, 2011, p.159)

b) Norma \leq Turing: consiste na demonstração de que a Máquina de Turing é capaz de simular o uso dos registradores X e Y da Norma. Essa simulação ocorre da seguinte maneira:

b1) O conteúdo de cada registrador (valor natural) é implementado de forma unária em Turing (ou seja, a repetição de um mesmo símbolo ao longo da fita, tantas vezes quanto for o valor do número natural);

b.2) O registrador **X** ocupa as células pares da fita, e **Y**, as ímpares maiores que 1 (DIVERIO; MENEZES, 2011, p. 159).

Suponha uma máquina de Turing $M = (\Sigma, Q, \Pi, q_0, F, V, \beta, \odot)$. A simulação de M por um programa P em uma máquina Norma depende das seguintes definições:

a) *Fita*. A fita é simulada como um arranjo unidimensional no registrador X. Cada célula da fita corresponde a uma posição no arranjo. Os símbolos de cada célula da fita são codificados como um número natural. Onde para um alfabeto $\Sigma = \{a_1, a_2, a_3, \dots, a_n\}$, o símbolo a_i é codificado como o número natural i , o símbolo β é codificado como 0 e \odot como $n + 1$.

b) *Estados*. Para o conjunto de estados $Q = \{q_0, q_1, q_2, \dots, q_n\}$ em M, o programa P em Norma possui um conjunto de instruções rotuladas equivalente, onde o valor do rótulo referente a um estado qualquer é calculado pela equação $2^{(u+1)}$, onde u representa o estado. Assim, para o conjunto de estados $Q = \{q_0, q_1, q_2\}$ em M, temos os rótulos $\{2, 4, 8\}$ em P.

c) *Estado corrente*. Para simular o estado corrente de M em uma máquina Norma é utilizado o registrador Q, que durante a simulação assume valores de n números naturais, onde n é o número de estados de M.

d) *Cabeça da fita*. A posição corrente da cabeça de leitura da fita é simulada em Norma com a utilização de um registrador C. Ele armazena a posição corrente do arranjo armazenado no registrador X.

e) *Função programa*. Compreende a simulação de uma transição de M representada pela expressão:

$\Pi = (q_u, a_r) = (q_v, a_s, m)$. Onde:

q_u = estado corrente;

a_r = símbolo lido;

q_v = próximo estado a ser visitado;

a_s = símbolo a ser gravado na fita;

m = sentido do movimento (E = esquerda; D = direita);

Para simular o sentido do movimento, “m”, a máquina Norma mantém o registrador C. Para simular o movimento para direita C é incrementado, movimento para a esquerda C é decrementado.

Como a transição depende do estado corrente, “ q_u ”, e do símbolo lido, “ a_r ”, numa instrução rotulada (2^{u+1}) especifica um desvio incondicional para uma instrução rotulada pelo par (2^{u+1} , r).

Algumas instruções em Norma conterão as macros End_A ou End_Q, que simulam o endereçamento indireto. O valor do registrador A é expresso por a . O valor do registrador Q é expresso por 2^{v+1} , que representa o estado q_v .

A expressão $3^Q \cdot 5^{x(C)}$ corresponde ao valor de A. Assim é possível distinguir as transições possíveis em um função do estado corrente e símbolo lido.

f) *Codificação*. Codificação do conteúdo inicial da fita em um arranjo unidimensional no registrador X.

g) *Decodificação*. Decodificação do conteúdo final da fita armazenado no registrador Y.

Considerando a equivalência entre programas recursivos e monolíticos, onde para todo programa monolítico existe um recursivo equivalente, a prova de máquina Norma \leq máquina de Turing se satisfaz considerando apenas programas monolíticos.

Para isso suponha um programa monolítico P em uma máquina Norma de dois registradores (X, Y). A simulação de P por uma Máquina de Turing $M = (\Sigma, Q, \Gamma, q_0, F, V, \beta, \odot)$, onde o alfabeto Σ é um conjunto unário $\{1\}$, pode ser definida como:

a) *Registrador X*. Armazena em unário nas células pares da fita de M. Logo, um determinado número natural n armazenado em X, n células pares da fita armazenam o símbolo 1.

b) *Registrador Y*. O registrador Y é armazenado na fita em unário, mas nas células ímpares, excetuando-se a primeira célula que armazena o símbolo \odot , marco do início da fita.

c) *Rótulos de instruções*. Os rótulos inicial e final correspondem ao estado inicial e final, respectivamente. Quanto aos demais rótulos, um rótulo r corresponde a um estado q_r de M.

d) *Programa*. A simulação do programa P por M se dá através da simulação das rotinas básicas de um programa em Norma:

- **Adição**: essa operação em uma Máquina Norma compreende o incremento do valor de um registrador, e na simulação consiste em adicionar o símbolo 1 na primeira célula par vazia para incrementar o valor do registrador X e para incrementar o valor de Y adiciona-se o símbolo 1 na primeira célula ímpar vazia, excetuando-se a primeira célula da fita.
- **Subtração**: Consistem em gravar o símbolo β sobrescrevendo o último símbolo 1 armazenado em células pares para decrementar o valor de X ou

sobrescrever o último símbolo 1 armazenado em células ímpares para decrementar o valor de Y.

- Teste: Verifica se a primeira célula par armazena o símbolo β para verificar se o registrador X é igual a zero ou se a segunda célula ímpar armazena o símbolo β , para verificar se Y é igual a zero.

e) *Codificação*. Consiste em codificar o conteúdo inicial do registrador X em unário armazenando o valor nas células pares da fita.

f) *Decodificação*. Compreende a decodificação dos valores armazenados nas células ímpares da fita repassando-o para o registrador Y.

Resultados e Discussão

Turing to Norma enquadra-se na categoria de conversores de máquinas universais, pois suas funcionalidades são:

- Conversão de um conjunto de estados de uma Máquina de Turing para um programa em Máquina Norma.
- Conversão de um programa em Máquina Norma em um conjunto de estados de uma Máquina de Turing.
- Exportar o programa em Máquina Norma.

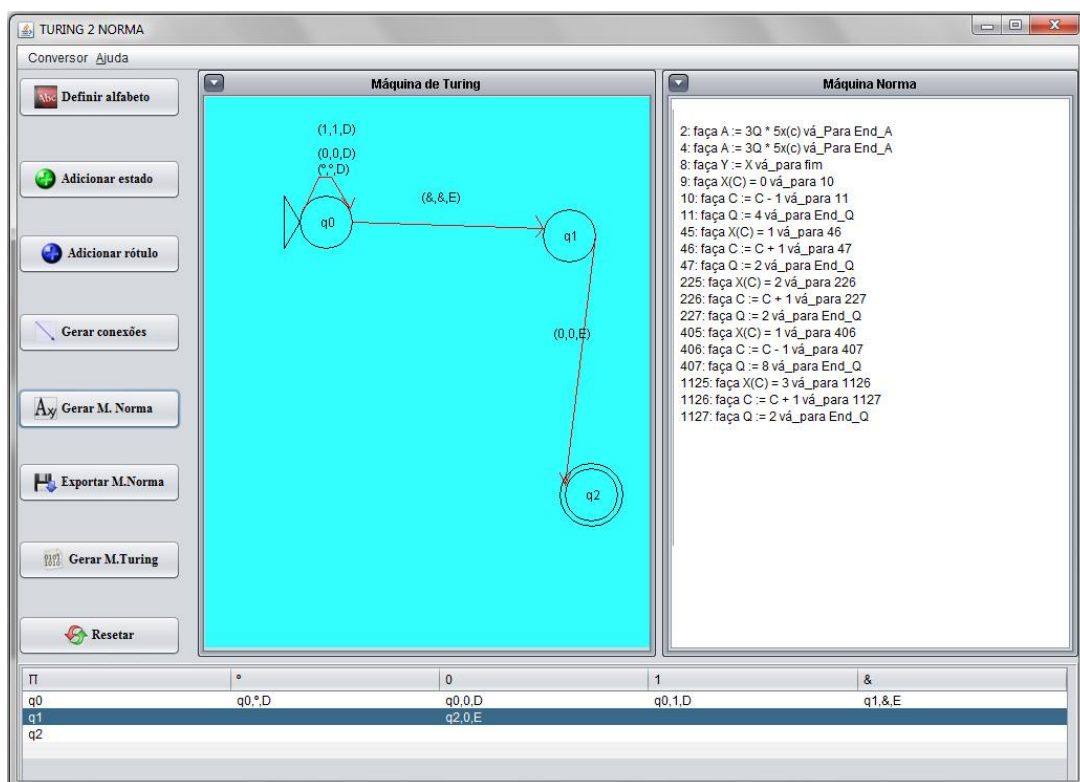


Figura 1. Tela principal do programa Turing to Norma.

A Figura 1 apresenta a tela principal do conversor Turing to Norma. Seu funcionamento é simples. Para encontrar a máquina Norma equivalente a uma Máquina de Turing genérica, o usuário deve fornecer o alfabeto, o conjunto de estados e preencher a tabela de transições da Máquina de Turing. Em seguida, clicar no botão “gerar conexões”, para o programa carregar as transições, e por fim, clicar no botão “gerar M. Norma”. Uma vez seguidos esses passos será mostrado no painel a direita do painel de visualização da Máquina de Turing, as instruções da Máquina Norma equivalente. Caso deseje, o usuário poderá salvar um arquivo de texto simples (.txt) com os rótulos e instruções do programa equivalente em Máquina Norma.

Para converter um conjunto de instruções de um programa em Máquina Norma para uma Máquina de Turing genérica é preciso incluir as instruções através do botão “adicionar rótulo”. Ao clicar nesse botão irá aparecer um quadro onde o usuário irá fornecer o rótulo, a instrução a ser executada e o próximo rótulo. No caso de instruções de teste deve ser informado o rótulo, qual o teste, o rótulo se verdadeiro e o rótulo se falso. Em seguida, clicar no botão “Gerar M. Turing”. Será gerada a tabela de transição da Máquina de Turing equivalente.

Conclusões

Neste trabalho foi apresentado um software capaz de, a partir de uma Máquina de Turing, desenvolver um programa em Máquina Norma, e um programa em Máquina Norma gerar a tabela de transição de uma Máquina de Turing equivalente.

Como trabalho futuro há o aperfeiçoamento da representação gráfica da Máquina de Turing de modo que também seja possível visualiza-la quando ela for produto da conversão de um programa em Máquina Norma. Ou ainda incluir um simulador desses dispositivos de computação.

Referências Bibliográficas

ALVES, D. P. **Equivalência de Máquinas Universais: Demonstração, Análise e Simulação**. São Leopoldo, 2007. Disponível em :<<http://debora.wait4.org/tc.pdf>> acesso em 20 de maio de 2012.

DIVERIO, T. A.; MENEZES, P. B. **Teoria da computação: Máquinas universais e computabilidade**. 3. ed. Porto Alegre: Bookman, 2011. p 157-179.

SIPSER, M. **Introdução à Teoria da Computação**. 2 ed. São Paulo: Cengage Learning, 2011. 143p.