Teoria da Computação: Máquinas, Programas e suas Equivalências

Douglas Rodrigues Almeida[1], Daniel Dessbesell[1], Daniel Padilha[1], Thielyon Pinheiro[1], Laércio Castro[1],Vanderlei Cardoso[1],Álvaro Hauenstein[1], Alex Telocken[1]

[1]Curso de Ciência da Computação – Universidade de Cruz Alta (UNICRUZ)– Cruz Alta – RS – Brasil

{printsulinformatica@gmail.com, danielsp@ceee.com.br,daniel.dessbesell@bol.com.br, thielyon@gmail.com,laercio.ecco@gmail.com, oc.alvaro0113@gmail.com, douglasrodrigues.trabalho@gmail.com, Telockenalex@unicruz.edu.br}

Resumo: *Este trabalho faz um comparativo, mostrando a equivalência entre maquinas e programas, entre os modelos universais mais conhecidos como Máquina de Turing, Máquina de Post, Autômato de Duas Pilhas e Máquina de Registradores NORMA. Para todos estes modelos são definidos formalmente sua estrutura e seu modelo de execução. Respectivamente, são apresentadas conversões entre Máquinas de Turing e Máquinas de Post; entre Máquinas de Post e programas para Máquinas Norma; entre programas para Máquinas Normas e Autômatos de Duas Pilhas; entre Autômatos de Duas Pilhas e Máquinas de Turing. O fechamento do ciclo prova que todos os modelos são equivalentes entre si. Adicionalmente, todas as traduções são provadas corretas, ou seja, a tradução preserva a linguagem reconhecida pelas máquinas.*

Abstract: *This work makes a comparison between the universal models better known as Turing Machine, Post Machine, Two Stack Automaton and NORMA Register machine. For all these models are formally defined its structure and its execution model. In particular, conversions between Turing Machines and Post Machines are presented; Between Post Machines and programs for Standard Machines; between programs for Standard Machines and Two-Battery Automata; between two-stack automata and Turing machines. Closing the cycle proves that all models are equivalent to each other. In addition, all translations are proven correct, that is, the translation preserves the language recognized by the machines.*

1. Introdução

Neste Artigo veremos uma equivalência de forma cíclica, sendo a Máquina de Turing simulada por uma Máquina de Post, onde esta será simulada por uma Máquina NORMA, que por sua vez será simulada por um Autômato de Duas Pilhas e por fim este último será simulado por uma Máquina de Turing. Ou seja, a prova fará uso da transitividade da equivalência.

2. Equivalência de Máquinas

De acordo com Alves (2007), duas máquinas são consideradas equivalentes, quando a máquina 1 pode simular a máquina 2 e a 2 pode simular a máquina 1. Duas máquinas são ditas "equivalentes" se uma pode simular a outra e vice-versa. Sejam:

Tabela 1. Exemplo de máquinas equivalentes

M = (VM, X, Y, πXM , πYM, ΠOM, ΠTM)
e
N = (VN, X, Y, πXN , πYN, ΠON, ΠTN)

No caso mostrado na tabela 1, N "simula fortemente" M:

$\forall P, \exists Q \mid (P, M) = (Q, N)$

N "simula" M:

$\forall P, \exists (Q, c: XM \rightarrow XN, d: YN \rightarrow YM) \mid (P, M) = d \circ (Q, N) \circ c$

$(M \equiv N) \Leftrightarrow ((M \text{ simula } N) \wedge (N \text{ simula } M))$

Para a execução das simulações, serão utilizados as máquinas de Turing, Post, NORMA, e o autômato de duas pilhas, cujas definições formais, de acordo com Alves (2007) se encontram na tabela abaixo:

Tabela 2. Tabela de definições formais.

Maquina de Turing
Uma Máquina de Turing é uma sétupla: *MT = (Q, ∑, T, Ȯ, q0, qa, qr)*
Definição formal da Máquina de Post
Uma Máquina de Post é uma sétupla: *MP = (L,E, ∑, T, Ȯ, qi, {qa, qr})*
Definição formal da Máquina de NORMA
Uma Máquina Norma é uma tupla : *MN = ({Rk}∞/k=0, {{ek}, {sk}, {zk}, {{adk}, {subk}}∞/k=0)*
Definição formal da Máquina Autômato de Duas Pilhas
Um Autômato de Duas Pilhas é uma sêxtupla: *MA = (Q, ∑, T, Ȯ, q0, F).*

2.1. Teorema de Equivalência Entre Turing e Post

Qualquer linguagem aceita por um Máquina de Turing pode ser aceita por uma Máquina de Post. A figura 1 mostra a simulação de uma máquina de Turing em uma máquina Post, conforme Alves (2007):

.

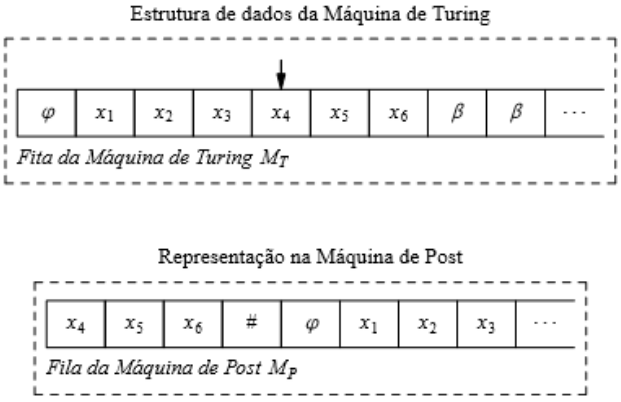

Figura 1. Simulação da Máquina de Turing.

Uma Máquina de Turing MT = (Q,Σ,Γ,δ,q0,qa,qr) é traduzido para uma Máquina de Post MP = (L,E,Σ,Γ,δ,qi,{qa,qr}) tal que:

1. L∪E = QT ×{Lt1,Lt2,Lt3,Lt4,Et1,Et2,Et3,Et4,Et5γ,Et6,Et7,Et8γ,escrever à frente, deslocar à direita} Onde para cada estado da MT é definido um conjunto de estados para a MP.

2. ΣP = ΣT

3. ΓP = ΓT ∪ {$} ∪ {#} 4. δP =St∈δTTt passa a ter uma transição incondicional para a tradução de q0, excluindo-se a escrita de # na fila.

6. qa = qa

7. qr = qr

Prova: Uma Máquina de Turing MT = (Q,Σ,Γ,δ,q0,qa,qr) e uma Máquina de Post MP = (L,E,Σ,Γ,δ,qi,{qa,qr}) construída de acordo com a tradução dada.

Se w ∈L(MT), então existe uma computação da forma (q0,φaw′$#)) ⊢∗ MT (qa,φγw′$#γ′)) onde w = aw′ e γ∈Γ∗

Utilizando a tradução acima, é possível mostrar que existe uma computação da forma (q0,w#) ⊢∗ MP (qa,γ#) tal que γ∈Γ∗. Ou seja, uma computação que aceita w. Esta demonstração será realizada por indução no comprimento da computação da palavra w.

2.2. Teorema de Equivalência Entre Post e Norma

Qualquer linguagem aceita por um Máquina de Post pode ser aceita por um programa para uma Máquina NORMA.

Simulando a máquina Post em uma máquina NORMA conforme Alves (2007):

Uma Máquina de Post MP = (Qp ,Σp, Γp , δp , {qi}, {qa, qr}) e uma Máquina NORMA

MN = ({Rk}∞/k=0,X, Y, {{ek}, {sk}, {zk}, {{adk}, {subk}}∞/k=0) tal que:

1. Qp é traduzido para um conjunto de instruções rotuladas tal que para todo q∈Qp implica em uma instrução rotulada,

2. Σn = ASCII(Σp),

3. Γn = ASCII(Γp),

4. δn = i :faça A := 2RE × 3RX(RI) vá para End_RA, para qualquer leitura e

i :faça A := 2RE × 3RX(RF) vá para End_RA, para qualquer escrita.

5. qi = {i0, i1, i2, i3, i4},

6. qa = a,

7. qr = r.

Prova: Seja uma MP = (QP ,Σp, Γp , δp , qi, {qa, qr}) e uma Máquina NORMA onde:MN =({Rk}∞/k=0,X, Y, {{ek}, {sk}, {zk}, {{adk}, {subk}}∞/k =0) para a qual é definido um programa PN = (X, Y, I,C, i0) construído de acordo com a tradução dada e sejaw∈L(MP).

Se w ∈L(MP), então existe uma computação da forma (qi,w) ⊢∗MP (qa). Pretende-se então mostrar que existe uma computação em PN da forma (0, ek(w))) ⊢∗MN (fim, "), ou seja, uma computação que aceita w.

A Figura 2 mostra a simulação de uma máquina NORMA para máquina autômato de duas pilhas conforme Alves (2007):

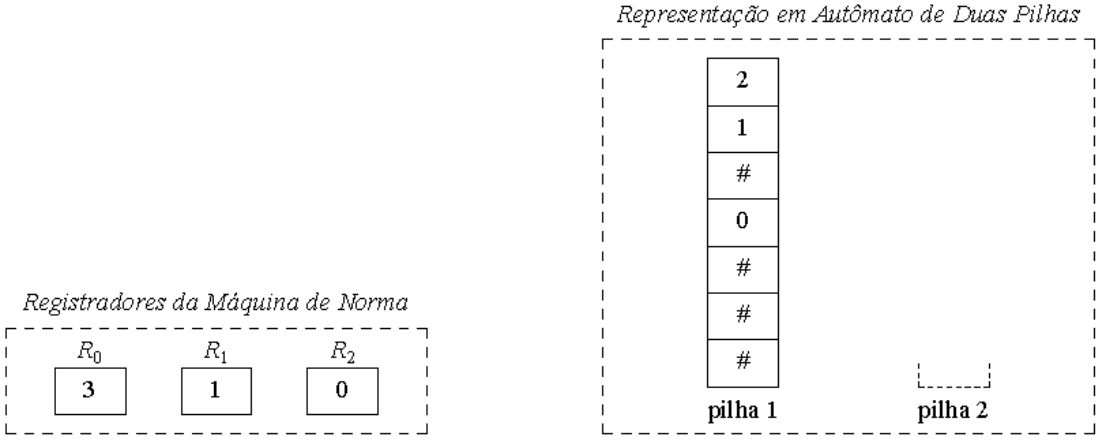

Figura 2. Representação dos registradores nas pilhas.

Como mostrado na Figura 2, o conteúdo de cada registrador é gravado na pilha após um símbolo auxiliar que indica o índice do registrador.

A máquina NORMA ({Rk}∞/k=0, {{ek}, {sk}, {zk}, {{adk}, {subk}}∞/k=0), é traduzido para um Autômato de Duas Pilhas MA = (QA, Σa, Γa ,δa, q0, F) da seguinte maneira:

1. QA = T(I) o conjunto de estados é formado pela tradução do conjunto de instruções da Máquina NORMA de acordo com a Figura 5,

2. ΣA = N ∪# onde N é o conjunto de índices dos registradores utilizados na Máquina NORMA MN,

3. ΓA = ΓA ∪ {$},

4. δA = St∈δATt definida pela Figura 4,

5. q0 = r0,

6. F é a tradução das instruções finais do programa da Máquina NORMA.

2.3. Teorema de Equivalência Entre Norma e Autômato de Duas Pilhas

Qualquer linguagem aceita por um Máquina NORMA pode ser aceita por um Autômato de Duas Pilhas.

Prova: Seja uma Máquina NORMA

MN = ({Rk}∞k=0,X, Y, {{ek}, {sk}, {zk}, {{adk}, {subk}}∞k=0) para a qual é definido um programa PN = (X, Y, I,C, i0) e um Autômato de Duas Pilhas MA = (QA, Σa, Γa , δa, q0, F) construída de acordo com a tradução dada e seja w ∈ L(PN).

Se w ∈L(PN), então existe uma computação da forma (r, ek(w)) ⊢∗MN (f, ε) tal que f é um rótulo final.

Existe então uma computação em MA da forma (q0,w, ε, ε)) ⊢∗MA (qa, ε, ε, y)), ou seja, uma computação que aceita w.

A figura abaixo demonstra a simulação da máquina autômato de duas pilhas em uma máquina de Turing conforme Alves (2007):

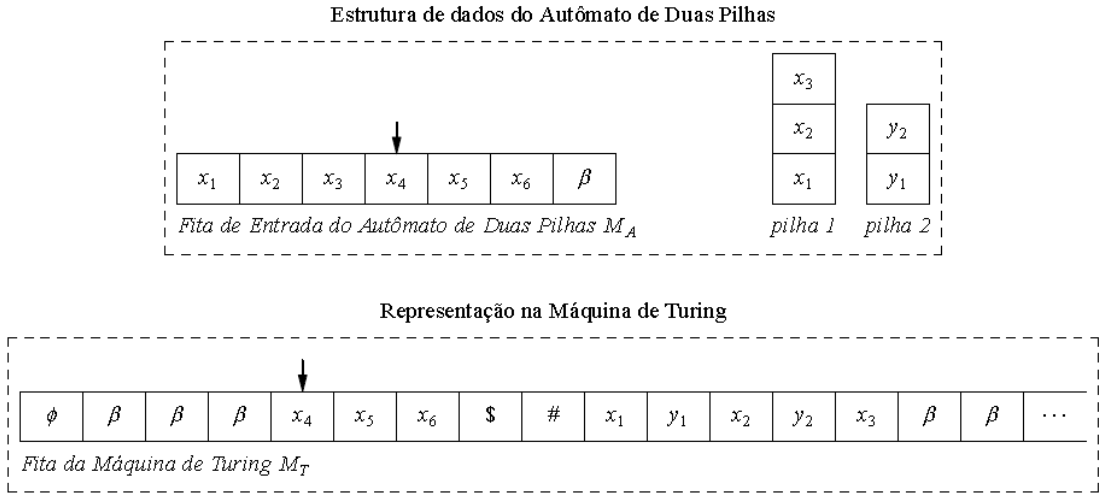

Figura 3. Representação da fita de entrada e das pilhas Autômato de Duas Pilhas em uma fita da Máquina de Turing.

O conjunto de estados gerados na tradução de Autômato de Duas Pilhas para Máquina de Turing é gerado de acordo com o conteúdo das transições do estado original.

Um Autômato de Duas Pilhas MA = (QA, Σa, Γa,δa, q0, F) é traduzido para uma Máquina de Turing MT = (QT,ΣT ,ΓT , δT , q0, qa, qr) tal que:

1. QT = QA ∪ {qt1, qt2, . . . , qt13} ∪ {qa, qr}

2. ΣT = ΣA,

3. ΓT = ΓA ∪ {$,#, ϕ},

4. δT = Ut∈δATt

5. q0T = q0A.

6. qa será o estado destino das transições que serão incluídas a partir dos estados de aceitação do Autômato de Duas Pilhas((qA, $), (qa, $,←))*se* qA∈F

7. qr será o estado destino das transições que serão incluídas a partir dos estados de rejeição do Autômato de Duas Pilhas((qA, $), (qr, $,←))*se* qA /∈F

2.4. Teorema de Equivalência Entre Autômato de Duas Pilhas e Máquina de Turing

Qualquer linguagem aceita por um Autômato de Duas Pilhas pode ser aceita por uma Máquina de Turing.

Prova: Seja um Autômato de Duas Pilhas MA = (QA, ∑, ΓA, δA, q0, FA) e uma Máquina de Turing MT = (QT, ∑, ΓT , δT , q0, qa, qr) construída de acordo com a tradução dada na Definição 3.4 e seja w ∈ L(MA).

Se w ∈L(MA), então existe uma computação da forma (q0, w, ε, ε) ⊢∗MA (f, ε, ε, y) tal que f ∈ FA e y∈Γ∗. Existindo assim uma computação em MT da forma (q0, ϕw$#)) ⊢∗MT (qa, ϕββ . . . β$#y′)) onde o número de brancos entre ϕ e $ é igual ao comprimento de w, |w|. Ou seja, uma computação que aceita w.

Completando assim o ciclo de equivalência entre as máquinas indicadas no início deste artigo.

3. Equivalência De Programas Em Uma Máquina

Sejam P e Q dois programas arbitrários, não necessariamente do mesmo tipo e uma máquina M qualquer. Então o par (P, Q) está na Relação Equivalência de Programas na Maquina M, denotado por:

$$P \equiv_M Q$$

se, e somente se as correspondentes funções computadas na máquina M são iguais, ou seja:

$$(P, M) = (Q, M)$$

P e Q, nesse caso, são ditos "programas equivalentes na máquina M", ou "programas M - equivalentes".

4. Equivalência Forte De Programas

Segundo Diverio &Menezes, (1999), é importante que se considere a Relação Equivalência Forte de Programas por várias razões, como por exemplo:

- permite identificar diferentes programas cujas funções computadas coincidem, para qualquer máquina;

- as funções computadas por programas equivalentes fortemente têm a propriedade de que as mesmas operações são efetuadas na mesma ordem, independentemente do significado dos mesmos (por que não diferentes testes ou operações ou, ainda, diferente ordem?);

- fornece subsídios para analisar a complexidade estrutural de programas.

Dois programas P e Q, de quaisquer tipos, são ditos "fortemente equivalentes", denotado:

$$P \equiv Q$$

Se, e somente se:

$$\forall M, (P, M) = (Q, M0)$$

Ou seja, P e Q são fortemente equivalentes se, e somente se, as respectivas funções computadas coincidem para qualquer máquina M que se possa considerar. Essa relação induz a uma partição do conjunto de todos os programas em classes de equivalências. Ela permite analisar, de forma comparativa, as propriedades exibidas pelos programas, como é o caso da sua complexidade estrutural.

5. Definição De Máquina De Traços

Segundo Diverio & Menezes, (1999), uma máquina de traços não executa as operações propriamente ditas, apenas produz um histórico ou rastro da ocorrência delas, chamado de traço. O traço propriamente dito é uma palavra em um alfabeto de identificadores de operações. De acordo com Diverio &Menezes, (1999), uma máquina de traços é representada pela definição mostrada na tabela 3:

Tabela 3. Definição da Máquina de Traços.

M=(Op*,Op*,Op*,idOP*,idOP*,∏F,∏T

6. Considerações Finais ou Conclusão

Um programa de qualquer tipo não pode ser modificado dinamicamente, durante uma computação, e o mesmo, para ser fortemente equivalente a outro, não pode conter ou usar facilidades adicionais como memória auxiliar ou operações ou testes extras. Para que um programa monolítico possa simular uma recursão sem um número finito e predefinido de quantas vezes a recursão pode ocorrer, seriam necessárias infinitas opções de ocorrências das diversas operações ou testes envolvidos na recursão em questão. Infinitas opções implicam um programa infinito, o que contradiz a definição de programa monolítico, o qual é constituído por um conjunto finito de instruções rotuladas.

Neste artigo foi demonstrado que diversas maquinas equivalentes devem ser capazes de simular uma a outra, mostrando uma equivalência de forma cíclica, onde a Máquina de Turing foi simulada por uma Máquina de Post, que foi simulada por uma Máquina NORMA, que foi simulada por um Autômato de duas pilhas, que foi simulado por uma Máquina de Turing. O fechamento do ciclo provou que todos os modelos são equivalentes entre si. Adicionalmente, todas as traduções são provadas corretas, ou seja, a tradução preserva a linguagem reconhecida pelas máquinas.

Referências Bibliográficas

ALVES, Debora Pandolfi, Equivalência de Máquinas Universais: Demonstração, Análise e Simulação, julho de 2007 Universidade Do Vale Do Rio Dos Sinos.

PEREIRA, Andréa, Aula 5 Teoria da Computabilidade, Curso Ciência da Computação, 2006, Universidade de Cruz Alta UNICRUZ.

RAMOS, Marcus Vinícius Midena Ramos, maio de 2012, Universidade Federal do Vale do São Francisco UNIVASF

DIVERIO, Tiarajú A. MENEZES, Paulo B., Máquinas Universais e Computabilidade. 1999, Teoria da Computação - 3.Ed. - UFRGS Bookman Editora