F-RAM - Estudos Preliminares

Giovani Ângelo Silva da Nóbrega **Benjamín Callejas Bedregal**
Depto. de Informática e Matemática Aplicada, CCET,UFRN
(84)3215 3815, Natal, RN
E-mail: giovani.angelo@gmail.com, bedregal@dimap.ufrn.br,

Resumo Neste presente trabalho trazemos uma proposta para uma Máquina RAM *fuzzy*(F-RAM) incluindo uma defingão, uma sintaxe, sua semântica formal e seu poder computacional a partir do modelo computacional clássico de máquina RAM proposto por Bedregal e Santiago [5]. Propusemos uma Máquina de Turing Fuzzy Multi-Fita baseada no modelo de [2] afim de servir como base para um estudo sobre a computabilidade da F-RAM. Mostramos que para uma Máquina de Turing Fuzzy Multi-Fita computa os passos computacionais possíveis de uma F-RAM.

Palavras-chave F-RAM

1 Introdução

Os diversos modelos de computação clássicos computam elementos da matemática e da lógica clássica. Pensar em modelos que sejam capazes de computar elementos de outras lógicas é natural devido ao forte crescimento das aplicações de lógicas não clássicas na computação afim de resolver problemas que a computação clássica não é capaz de solucionar ou soluciona de forma insatisfatória. Uma dessas lógicas é a lógica fuzzy que desda sua mensão por L.A Zadeh [1] foi analisada a sua capacidade de computação de informações. Bedregal e Figueira [2] propuseram uma máquina de Turing *fuzzy* a partir do modelo de Wiedermann [3] e analisaram os seus poderes computacionais. Com base nessa máquina desenvolvemos alguns estudos a respeito de um modelo computacional para uma máquina RAM *fuzzy*, ou seja *F-RAM*, e uma proposta para uma semântica formal e uma sintaxe. Na seção 2 falamos de alguns fundamentos básicos da teoria fuzzy que será considerada em todo o artigo e na seção 3 introduziremos uma definição para a máquina RAM *fuzzy* (F-RAM), na seção 3.1 uma sintaxe formal e na seção 3.2 a semântica formal da máquina. Na seção 4 definimos uma Máquina de Turing Fuzzy Multi-Fita extendida da máquina proposta por Bedregal e Figueira [2]. Na seção 5 mostraremos a relação da Máquina de Turing Fuzzy Multi-Fita e a F-RAM considerando o poder de computabilidade das duas.

2 Elementos da Teoria Fuzzy

Seja $\mathbb{I}$ o intervalo fechado [0,1]. Um conjunto *fuzzy* $\mathcal{A}$ em um universo $\mathcal{U}_{\mathcal{A}}$ (um conjunto clássico) é caracterizado pelo grau de relacionamento dado pela função:

$$\mu_{\mathcal{A}} : \mathcal{U}_{\mathcal{A}} \to \mathbb{I}$$

Para cada $x \in \mathcal{U}_{\mathcal{A}}, \mu_{\mathcal{A}}$ fornece um grau de pertinência para o elemento x em um conjunto *fuzzy* $\mathcal{A}$. Chamamos de *suporte* o conjunto que:

$$\mathcal{S}(\mathcal{A}) = \{x \in \mathcal{U}_{\mathcal{A}} \mid \mu_{\mathcal{A}}(x) > 0\}$$

E denominamos o conjunto *crisp* como:

$$\mathcal{C}(\mathcal{A}) = \{x \in \mathcal{U}_{\mathcal{A}} \mid \mu_{\mathcal{A}}(x) = 1\}$$

2.1 Conectivos Fuzzy

Norma triângular ou t-norma, foi introduzida por Schweizer and Sklar[1] com a intensão de modelar os espaços métricos probabilísticos. Além disso, Alsina, Trillas e Valverde[2] mostraram que a nocão de t-norma é adequada ao modelo de conjução na lógica *fuzzy* ou eqüivalente a intersecção de conjuntos *fuzzy*.

Seja uma operação binária $*$ em $\mathbb{I}$, $* : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$, é uma norma triângular ou *t-norma* se:

- $x * y = y * x, \forall\ x, y \in \mathbb{I}$
- $x * 1 = x, \forall\ x \in \mathbb{I}$
- $x * (y * z) = (x, y) * z, \forall\ x, y, z \in \mathbb{I}$
- Se $x \leq x'$ e $y \leq y'$ então $x * y \leq x' * y'$, $\forall\ x, x', y, y' \in \mathbb{I}$

Uma *t-norma* é dita uma *t-norma computável* se a aplicação
$* : \mathbb{I} \cap \mathbb{Q} \times \mathbb{I} \cap \mathbb{Q} \to \mathbb{I} \cap \mathbb{Q}$ é uma aplicação *computável.*

2.2 Funções Fuzzy

Dado dois conjuntos fuzzy $\mathcal{A}$ e $\mathcal{B}$ sobre um universo $\mathcal{U}_{\mathcal{A}}$ e $\mathcal{U}_{\mathcal{B}}$, respectivamente, uma *função parcial clássica* f, $f : \mathcal{U}_{\mathcal{A}} \to \mathcal{U}_{\mathcal{B}}$ é uma *função parcial fuzzy* de $\mathcal{A}$ em $\mathcal{B}$ se:

$$\forall\ x \in \mathcal{U}_{\mathcal{A}}, f(x) \uparrow\ \textit{ou}\ \mu_{\mathcal{B}}(f(x)) \leq \mu_{\mathcal{A}}(x)$$

Dubois e Prade em [4] coloca que a desigualdade é "$\geq$", mas para considerar propriedades já provadas sobre Máquina de Turing Fuzzy Multi-Fita.

3 A Máquina F-RAM

A idéia de maquína F-RAM consistem em uma extensão de um modelo de RAM clássica, proposto em [5], para uma máquina RAM em que possui para cada linha de comando um índice (ou muitas vezes chamada de rótulo, quando não é um símbolo que represente quantidade),possui também um grau de pertinência *fuzzy* que tem sua computação atribuida a uma máquina externa capaz de tal. Como já é conhecido pela literatura, a máquina RAM é Turing computável, com isso, temos o objetivo de definir uma máquina F-RAM que seja Turing computável pelo modelo de máquina de Turing proposto por Bedregal e Santiago [2]

Definição 1 *Seja* $\mathcal{I} = \{1, 2, 3, \ldots, n\}$ *o conjunto enumerável de índices e* $R = \{R_1, R_2, \ldots, R_i \ldots\}$ *um conjunto enumerável de nomes de registradores. O conjunto de instruções RAM é o menor conjunto cujos elementos são definidos como,* $\forall\ i, m \in \mathbb{N}$ *e* $j \in \mathcal{I}$.

1. $j\ INC\ R_i$ *- adiciona uma unidade ao conteúdo do registrador Ri*
2. $j\ DEC\ R_i$ *- subtrai uma unidade do conteúdo do registrador Ri. Caso o registrador contenha 0, o dado será inalterado.*
3. $j\ CLR\ R_i$ *- insere 0 no registrador Ri.*
4. $j\ R_i \leftarrow R_m$ *- insere o conteúdo do registrador Rm no conteúdo do registrador Ri.*
5. $j\ JMP\ k$ *- executa a próxima instrução de índice k. Caso k não for do programa, a máquina para.*
6. $j\ R_i\ JMP\ k$ *- executa a instrução JMP k, caso o conteúdo do registrador* R_i *seja 0 .*
7. $j\ CONTINUE$ *- faz nada.*

Definição 2 *Um programa F-RAM é uma seqüência finita de instruções RAM onde cada linha do programa tem um grau de caracterização definido.*

3.1 Sintaxe

A linguagem RAM possui classes básicas de palavras(”alfabeto”da linguagem):

Nome de variáveis : Um nome de variável para um programa F-RAM consiste no nome do registrador, a letra R acompanhada de uma seqüência de algarismos , por exemplo, R1, R10, R2.

Indice da linha : O índice das linhas de um programa F-RAM consiste em uma cadeia de algarismos.

Símbolo de programa : Existem palavras reservadas em um programa RAM: **inc**, **dec**, **jmp**, **clr**, **continue**. Ainda existem os símbolos ” ←”e “**;**” que significam uma atribuição a algum registrador e a indicação de o próximo símbolo será o grau do comando RAM do grau relacionado a linha do programa, repectivamente.

⟨programa RAM⟩ ::= ⟨comandos⟩

Os comandos são seqüências (que podem ser vazias) de comandos válidos:

⟨ comandos ⟩ ::= λ | ⟨ comando ⟩;⟨ comandos ⟩

⟨comando⟩::= ⟨índice⟩ ⟨atribuição⟩ **;** ⟨grau⟩ |
⟨índice⟩ **inc** ⟨registrador⟩ **;** ⟨grau⟩ |
⟨índice⟩ **dec** ⟨registrador⟩ **;** ⟨grau⟩ | ⟨índice⟩ **jmp** ⟨índice⟩ **;** ⟨grau⟩ |
⟨índice ⟩ ⟨registrador⟩ **jmp** ⟨índice⟩ **;** ⟨grau⟩ |
⟨índice⟩ **clr** ⟨registrador⟩ **;** ⟨grau⟩

⟨ atribuição ⟩ ::= ⟨ registrador ⟩ ← ⟨ outro registrador ⟩

⟨ grau ⟩ ::= $x : \forall\ x \in \mathbb{Q} \cap \mathbb{I}$

⟨ índice ⟩ ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0

3.2 Semântica Formal

A semâtica denotacional da máquina F-RAM é dada pela função parcial *fuzzy*

$$[\![\ \ ]\!] : \text{F-RAM} \rightarrow [\mathcal{U_A}^* \rightarrow \mathcal{U_B}^*]$$

e $\mathcal{A}$ e $\mathcal{B}$ são conjuntos fuzzy.

Onde $[\![\mathcal{U_A}^* \rightarrow \mathcal{U_B}^*]\!] = \{\ f : \mathcal{U_A}^k \rightarrow \mathcal{U_B}^m$ é parcial fuzzy e $k, m \in \mathbb{N}\ \}$.

$$[\mathcal{U_A}^* \rightarrow \mathcal{U_B}^*] = \bigcup_{k,m \in \mathbb{N}} [\mathcal{U_A}^k \rightarrow \mathcal{U_B}^m]$$

Seja FP um programa F-RAM, onde $[FP]$=$[C]$. Se $C = c_1; c_2; \ldots; c_p$ então $[C] = [c_p] \circ [c_{p-1}] \circ \ldots \circ [c_1]$, onde para cada $j = 1, \ldots, p$ tem-se que $[c_j] : \mathcal{U_A} \rightarrow \mathcal{U_B}$. Um programa $F - RAM$ em execução é referênciado por um par $\langle k, (x_0, x_1, \ldots x_n, \ldots)\rangle$ o valor $k \in \mathcal{I}$ representa índice corrente e $(x_0, x_1, \ldots x_n, \ldots)$ o estado dos dados da memória na linha em execução.

1. $c_j = j\ CLR\ Ri\ ;\ \mu(j)$ para algum registrador $Ri \in R$

   $[j\ CLR\ Ri\ \mu(j)]\langle j, (x_0, x_1, \ldots, x_n, \ldots)\rangle =$

   $\langle j + 1, (\mu(j) * x_0, x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots)\rangle$

2. $c_j = j\ DEC\ Ri\ ;\ \mu(j)$ para algum registrador $Ri \in R$
$[j\ DEC\ Ri\ \mu(j)]\langle k, (x_0, x_1, \ldots, x_n, \ldots)\rangle =$
$\langle j+1, (\mu(j) * x_0, x_1, \ldots, x_{i-1}, x_i - 1, x_{i+1}, \ldots)\rangle$

3. $c_j = j\ INC\ Ri\ ;\ \mu(j)$ para algum registrador $Ri \in R$
$[j\ INC\ R_i\ \mu(j)]\langle j, (x_0, x_1, \ldots, x_n, \ldots)\rangle =$
$\langle j+1, (\mu(j) * x_0, x_1, \ldots, x_{i-1}, x_i + 1, x_{i+1}, \ldots)\rangle$

4. Se $c_j = j\ Ri \leftarrow Rm\ ;\ \mu(j)$ para dois registradores Ri e $Rm \in R$ quaisquer:
$[j\ Ri \leftarrow Rm\ ;\ \mu(j)]\langle j, (x_0, x_1, \ldots, x_n, \ldots)\rangle =$
$\langle j+1, (\mu(j) * x_0, x_1, \ldots, x_{i-1}, x_i \leftarrow x_m, x_{i+1}, \ldots)\rangle$

5. Se $c_j = j\ JMP\ q\ ; \mu(j)$ para algum $q \in \mathcal{I}$
$[j\ JMP\ q\ ; \mu(j)]\langle k, (x_0, x_1, \ldots x_n, \ldots)\rangle =$
$\langle q, (\mu(j) * x_0, x_1, \ldots, x_n, \ldots)\rangle$

6. Se $c_j = j\ Rm\ JMP\ q\ ; \mu(j)$ para algum $q \in \mathcal{I}$ e $Rm \in R$
$[j\ Rm\ JMP\ q\ ; \mu(j)]\langle j, (x_0, x_1, \ldots x_n, \ldots)\rangle$

$$= \begin{cases} \langle q, (\mu(j) * x_0, x_1, \ldots x_n, \ldots)\rangle \\ \text{, se } (x_0, x_1, \ldots, x_{m-1}, 0, x_{m+1}, \ldots) \\ \langle j+1, (\mu(j) * x_0, x_1, \ldots x_n, \ldots)\rangle \\ \text{,caso contrário} \end{cases}$$

4 Máquina de Turing Fuzzy Multi-Fita

A Máquina de Turing Fuzzy Multi-Fita consiste em uma extenção da Máquina de Turing Fuzzy proposta por Bedregal e Santiago [2].

Definição 3 *Uma Máquina de Turing Multi-Fita Não Determinística pode ser descrita como uma 7-upla $\mathcal{T} = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, F\rangle$. Onde Q é o conjunto finito de estados, Σ é o alfabeto de entrada, Γ é o alfabeto das fitas, $q_0 \in Q$ é o estado inicial da máquina, $\Box \in \Gamma$ é valor nulo, $F \subset Q$ é o conjunto de estado finais da máquina e*
$\delta : Q \times \Gamma^k \to Q \times (\Gamma \times \{R, L\})^k$

Para mapear os dados da fita utilizaremos as funções strings $head(s,i)$ (Dado uma string s e o índice de uma fita i, a função retorna o valor mais a esquerda da fita), $head^R(s,i)$ (Dado uma string s e o índice de uma fita i, a função retorna o valor mais a direita da fita) $tail(s,i)$ (Dado uma string s e o índice de uma fita i, a função retorna o valor s se esse valor for o mais a esquerda da fita) e $tail^R(s,i)$(Dado uma string s e o índice de uma fita i, a função retorna o valor s se esse valor for o mais a direita da fita). O valor i pode ser qualquer um valor que represente o índice ou o valor do cabeçote de alguma fita. Uma descrição de um instânte nesse modelo de máquina pode ser visto como uma tripla (u,q,v) onde u e v são strings contidas nas fitas e q é o estado corrente. Um movimento válido de (u,q,v) para (u',p,v') por um processo da máquina, descrevemos como $(u,q,v) \vdash_{\mathcal{T}} (u',p,v')$ quando ocorre.

$$\exists (q, head(s,i), p, b, R) \in \delta \text{ tal que } u' = u \circ b,$$

$v' = tail(v,i)$ ou

$\exists (q, head(s,i), p, b, L) \in \delta$ tal que $u = u \circ head(v', i)$ e

$v' = head^R(v,i) \circ b \circ tail(v,i)$

Definição 4 *A Máquina de Turing Fuzzy Multi-Fita é uma tripla $\mathcal{F} = \langle \mathcal{T}, *, \mu\rangle$ onde $\mathcal{T} = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, F\rangle$ é uma Máquina de Turing Não Determinística Multi-Fita, * é uma t-norma computável e μ é uma aplicação que define o grau de pertinência de cada movimento, ou seja: $\mu : \delta \to \mathbb{I}$*

Uma descrição instânea de uma Máquina de Turing Fuzzy Multi-Fita é um par (uqv, d) onde uqv é ID clássico da Máquin de Turing Multi-Fita e d é o grau de relacionamento acumulado sobre esse movimento. Um *movimento válido* de um ID (uqv, d) para $(u'p'v', d')$ é denotado por $(uqv, d) \vdash_{\mathcal{F}} (u'p'v', d')$ ocorre quando $(uqv) \vdash_{\mathcal{T}} (u'p'v')$

$$d' = \begin{cases} d * \mu(q, head(v,i), p, head^R(u',i), R) \text{ se} \\ tail(u',i) = u \\ d * \mu(q, head(v,i), p, head^R(tail(v,i),i), L) \text{ se} \\ tail(u,i) = u' \end{cases}$$

5 F-RAM e a Máquina de Turing Fuzzy Multi-Fita

Para mapear um programa F-RAM, propomos a existência de uma máquina externa que seja capaz de ler o programa a partir das funções abaixo.

H A função que mapeia uma linha do programa F-RAM, e retornando o valor do registrador que está em uso.

$$H : [comando] \to \mathbb{N}$$

in A função que mapeia um programa F-RAM a partir dos índices de cada linha do programa retornando os comandos da linha referente.

$$in : \mathbb{N} \to [comando]$$

∂ a função que recebe como entrada o valor do índice da fita (i) que será utilizada, o estado da máquina, e o símbolo na fita na posição corrente. Retornando o índice da fita que será utilizada, o novo estado da máquina o valor computádo e o passo que será executado em seguida na fita.

$$\partial : \mathbb{N} \times \mathbb{N} \times \mathbb{Q} \times \Gamma \longrightarrow \mathbb{N} \times \mathbb{N} \times \mathbb{Q} \times \Gamma \times W$$
$$\partial(k, i, q_k, u) \longmapsto (k', i', {q'}_k, u, w)$$

Onde $W = \{D, E, S\}$. e w é um elemento da lista W que representa os possÃveis passos que a máquina pode executar sobre a fita: D ,vai para a direita, E ,vai para a esquerda e S, vai para o início da fita.

Consideramos que a máquina Fuzzy Turing Multi-Fita computa o alfabeto unário. E que o grau fuzzy será computado externamente por uma outra máquina capaz de tal.

Teorema 1 *Uma máquina de Turing Fuzzy Mult-Fita computa uma Máquina F-RAM.*

Prova: Para cada comando possível em um programa F-RAM mostraremos que existe um conjunto de operações equivalentes na Máquina de Turing Multi-Fita

1. $[k\ CLR\ Ri\ ;\ \ \mu_k] =$

$$\begin{cases} \partial(k, i, q_0, 1) = (k, i, q_0, \Box, D)\ ;\ 1 \\ \partial(k, H(in(k)), q_0, \Box) = \\ (k+1, H(in(k+1)), q_0, \Box, S)\ ;\ \mu(k) \end{cases}$$

2. $[k\ DEC\ Ri\ ;\ \ \mu(k)] =$

$$\begin{cases} \partial(k, i, q_0, 1) = (k, H(in(k)), q_0, 1, D)\ ;\ 1 \\ \partial(k, i, q_0, \Box) = (k, i, q_0, \Box, E)\ ;\ 1 \\ \partial(k, i, q_1, 1) = \\ (k+1, H(in(k+1)), q_0, \Box, S)\ ;\ \mu(k) \end{cases}$$

3. $[k\ INC\ Ri\ ;\ \ \mu(k)] =$

$$\begin{cases} \partial(k, i, q_0, 1) = (k, i, q_0, 1, D);\ 1 \\ \partial(k, i, q_0, \Box) = \\ (k+1, H(in(k+1)), q_0, \Box, S)\ ;\ \mu(k) \end{cases}$$

4. $[k\ Ri \leftarrow Rm\ ;\ \mu(k)] =$

$$\begin{cases} \partial(k, i, q_0, 1) = (k, i, q_0, \Box, D)\ ;\ 1 \\ \partial(k, i, q_0, \Box) = (k, m, q_0, 1, S)\ ;\ 1 \\ \partial(k, m, q_0, 1) = (k, i, q_1, x, D)\ ;\ 1 \\ \partial(k, i, q_1, 1) = (k, i, q_1, 1, D)\ ;\ 1 \\ \partial(k, i, q_1, \Box) = (k, m, q_0, 1, S)\ ;\ 1 \\ \partial(k, m, q_0, x) = (k, m, q_0, x, D)\ ;\ 1 \\ \partial(k, m, q_0, \Box) = (k, m, q_1, \Box, S)\ ;\ 1 \\ \partial(k, m, q_1, x) = (k, m, q_1, 1, D)\ ;\ 1 \\ \partial(k, m, q_1, \Box) = \\ (k+1, H(in(k+1)), q_0, \Box, S); \mu(k) \end{cases}$$

5. $[k\ JMP\ l\ ;\ \ \mu_k]$=

$$\begin{cases} \partial(k, 1, q_0, \Box) = (l, H(in(l)), q_0, \Box, S); \mu(k) \\ \partial(k, 1, q_0, 1) = (l, H(in(l)), q_0, \Box, S); \mu(k) \end{cases}$$

6. $[k\ Ri\ JMP\ \ l\ ;\ \ \mu_k] =$

$$\begin{cases} \partial(k, i, q_0, \Box) = (l, H(in(l)), q_0, \Box, S);\ \mu(k) \\ \partial(k, i, q_0, 1) = \\ (k+1, H(in(k+1)), q_0, \Box, S);\ \mu(k) \end{cases}$$

Com a composição dos diversos comandos formamos um programa F-RAM e que cada passo dado por esse programa existe uma rotina da Máquina de Turing Fuzzy Multi-Fita que é uma seqüência dos passos descritos acima. Podemos concluir que para todo programa F-RAM existe uma seqüência de passos da Máquina de Turing Fuzzy Multi-Fita que realiza a mesma operação que o programa F-RAM.

■

Referências

[1] L.A. Zadeh,Fuzzy Sets, *Informations and Control 8*, (1965) 338-353.

[2] B. R. C. Bedregal, S. Figueira,“ Classical Computability and Fuzzy Turing Machines”, J.R. Correa, A. Hevia, and M. Kiwi (Eds.): LATIN 2006, LNCS 3887, pp. 154-165, Springer-Verlag Berlin Heidelberg 2006

[3] N.J. Higham,Caracterizing the super-Turing computing power and efficiency of classical fuzzy Turing machines, “Theorical Computer Sciences 317”,pp 61-69, 2004.

[4] “Fuzzy Sets and Systems: Theory and Applications”, pp 95-96, ACADEMIC PRESS, 1980.

[5] R. N.H. Santiago, B. R. C. Bedregal, Computabilidade e Limites da Computação,“Notas de Matemática Aplicada”, SBMAC, São Carlos, SP, 2004.