# UMA TÉCNICA COM LÓGICA NEBULOSA PARA VERIFICAÇÃO DE PROTOCOLOS DE COMUNICAÇÃO


Orlando Bernardo Filho[1,2]
Aloysio de Castro Pinto Pedroza[2,3]
Jorge Lopes de Souza Leão[2]

(1) Departamento de Enga. de Sistemas e Computação/FEN-UERJ
E-mail: orlando@labcomp.eng.uerj.br
(2) Programa de Engenharia Elétrica/COPPE-UFRJ
Caixa Postal: 68504 CEP: 21945-700 Rio de Janeiro - RJ
TEL: (021) 260-5010 FAX: (021) 290-6626
E-mail: orlando@gta.ufrj.br/aloysio@gta.ufrj.br/ leao@gta.ufrj.br
(3) Departamento de Eletrônica/EE-UFRJ



**Resumo**

Este trabalho apresenta uma técnica para verificação formal de protocolos de comunicação descritos com o emprego de Modelos de Estados, usando Lógica Nebulosa. O um Sistema de Lógica Nebulosa estima a função de busca heurística no grafo de execução do protocolo considerado, objetivando encontrar um estado global meta que satisfaça uma dada propriedade de interesse. A técnica proposta é usada para verificar propriedades de um protocolo multiponto a multiponto.

**Abstract**

This paper shows a communication protocol formal verification aproach by use Fuzzy Logic. The protocols are specified by States Models. A Fuzzy Logic System estimates a function heuristic search in a protocol execution graph in order to seek an objective global state to prove an interest property. Such aproach is used to prove a multipoint-multipoint protocol.


## 1 INTRODUÇÃO

Diversos trabalhos sobre verificação de protocolos de comunicação, baseados em modelos de estados, já foram propostos, abordando as mais variadas técnicas. A maior parte deles consiste em algum tipo de análise, envolvendo uma busca no espaço de estados do modelo considerado. A exploração completa desse espaço de estados seria suficiente para resolver de forma categórica o problema da verificação. No entanto, é sabido que diversos protocolos possuem uma árvore de alcançabilidade muito grande, provocando o esgotamento de recursos computacionais (tempo e/ou memória), o que inviabiliza a sua aplicação pura e simples.

O esgotamento dos recursos computacionais conduziu à proposição de algumas técnicas de verificação [1,2,3] que consistem, geralmente, em variações da idéia básica do método original (exploração do espaço de estados), objetivando uma redução do grafo de estados, durante a análise de alcançabilidade. Tais técnicas apresentaram resultados interessantes, principalmente em especificações de protocolos de tamanho médio. Entretanto, para protocolos muito complexos, a dificuldade permanece, sendo assim, alguns pesquisadores resolveram propor um outro grupo de métodos baseados em simulações [4,5,6].

Os métodos de verificação, baseados em simulações, não enfrentam o problema do esgotamento de recursos computacionais, pois esses têm como meta colocar o protocolo em execução sob condições normais. No trabalho de WEST [6], por exemplo, é mencionada como solução para a verificação de protocolos reais, uma pesquisa (simulação) aleatória. Contudo, sem fazer o armazenamento de pelo menos algumas possibilidades de execução, não há como garantir a existência de uma determinada propriedade no protocolo, pois esse pode percorrer sempre um mesmo caminho.

Neste artigo expõe-se uma solução alternativa para o problema da verificação de protocolos reais, valendo-se de uma busca heurística no seu grafo de execução com o emprego de Lógica Nebulosa. A busca do protocolo sob análise será guiada pelo conhecimento genérico da categoria de que esse protocolo pertence. Essa análise se assemelha à simulação, todavia a execução do protocolo é dotada de informação, o que evita tanto as buscas repetidas por certos caminhos quanto a explosão do espaço de estados.

Na seção seguinte, indroduzem-se os principais conceitos de Lógica Nebulosa. Na terceira seção mostra-se a técnica de busca heurística em espaço de estados a ser usada pelo método de verificação. Apresenta-se, na quarta seção, o método de verificação propriamente dito; enquanto, na quinta seção, é exposta a aplicação do método através de um exemplo, onde são analisadas as propriedades de um protocolo multiponto-multiponto.

## 2 A LÓGICA NEBULOSA

A Lógica Nebulosa é uma aplicação da teoria dos conjuntos nebulosos. Na matemática clássica, define-se um conjunto como uma coleção de elementos distintos ou objetos que podem ser finitos ou não. Tal conjunto pode ser descrito de várias maneiras, entre as quais, destacam-se: a enumeração de cada um de seus elementos ($A$ = {1,2,3,4}), ou uma condição de pertinência ($A = \{x \mid x \leq 5\}$).

Ao usar a descrição, a partir de uma condição de pertinência, quando um elemento ***x*** causar a avaliação dessa condição como verdadeira, este pertence ao conjunto; caso a avaliação seja falsa, conseqüentemente, não fará parte do conjunto. Para caracterizar o valor-verdade da condição de pertinência, pode-se empregar uma função que retorna **1**, se essa condição for verdadeira, e **0** se for falsa.

Na teoria dos conjuntos nebulosos [7,8,9], criada por ZADEH [10] em 1965, a função de pertinência não retornará apenas aos valores **0** ou **1**, mas qualquer outro valor deste intervalo **[0,1]**, o que revela a existência de vários graus de pertinência. Sendo assim, um conjunto nebuloso **Ã** possui o seguinte aspecto:

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\} \qquad (1)$$

O termo $\mu_{\tilde{A}}$ é a função de pertinência que mapeia o universo de discurso ***X*** ao espaço de pertinência ***M***. Quando ***M*** possui apenas os dois pontos **0** e **1**, **Ã** não é um conjunto nebuloso, sendo normalmente chamado de conjunto clássico.

Ao observar a definição de conjuntos nebulosos, percebe-se, que na verdade, esses podem ser vistos como um conjunto clássico de pares. O exemplo abaixo mostra o conjunto nebuloso **Ã** dos números inteiros próximos de **4**.

$$\tilde{A} = \{(1,0.3), (2,0.6), (3,0.8), (4,1.0), (5,0.8), (9,0.6), \ldots\} \qquad (2)$$

Como pode ser observado, **4** é o número mais próximo de **4**, logo ele recebe o índice **1.0** no conjunto nebuloso, enquanto os outros números que se afastam de **4** vão recebendo índices de menor valor. A função de pertinência que aparece explicitamente nos pares do conjunto do exemplo acima, pode ser fornecida analiticamente como é apresentado a seguir.

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid \mu_{\tilde{A}}(x) = (1 + (x-4)^2)^{-1}\} \qquad (3)$$

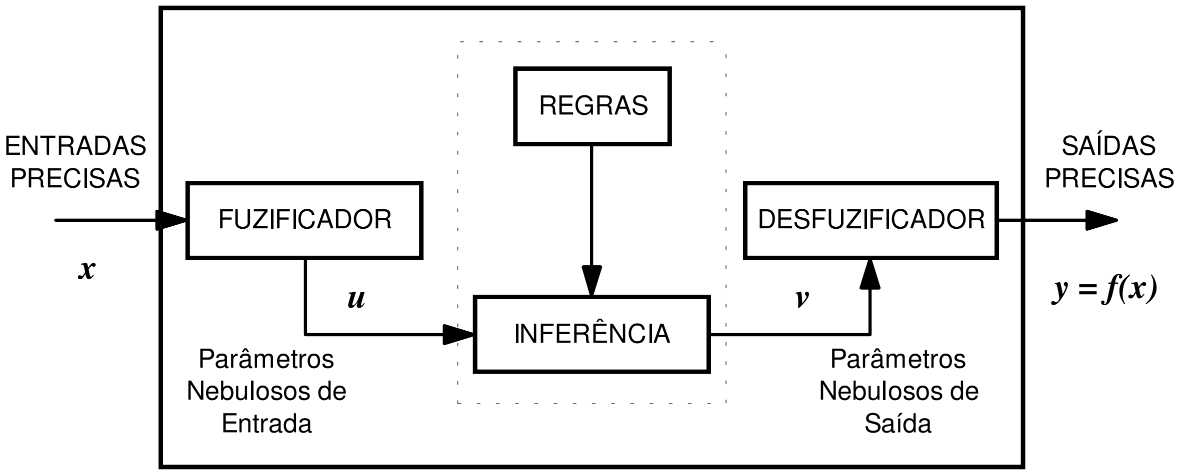
Dentro do estudo da Lógica Nebulosa e do raciocínio aproximado é bastante utilizada uma ferramenta conhecida como ***variável lingüística***, também chamada de variável de ordem mais alta. Essas variáveis não possuem números como valores, mas termos ou sentenças de uma linguagem natural ou artificial.

A idéia de se utilizar variáveis lingüísticas com valores de palavras ou frases de linguagens naturais ou artificiais está relacionada ao fato desses valores não serem tão específicos como números. Nesse sentido, nas ocasiões em que o tratamento de um valor numérico preciso é irrelevante para o resultado final da operação, como em controle de sistemas não lineares, por exemplo, a manipulação dos valores lingüísticos é mais fácil, pois estes estão em menor quantidade e são mnemônicos.

Uma variável lingüística é definida por uma quíntupla $(x, T(x), U, G, \tilde{M})$ onde:

- ***x*** é o nome da variável;
- ***T(x)*** denota o conjunto de termos de ***x***, isto é, o conjunto de nomes dos ***valores lingüísticos*** de ***x*** com cada valor sendo um conjunto nebuloso;
- ***U*** é o universo de discurso dos conjuntos nebulosos que formam os termos de ***T(x)***;
- ***G*** é a regra sintática, que usualmente tem a forma de uma gramática, para gerar os nomes dos valores lingüísticos;
- $\tilde{M}(X)$ é a regra semântica que atribui um significado ao termo ***X*** do conjunto ***T(x)***, ou seja, $\tilde{M}(X)$ é um subconjunto nebuloso de ***U***.

Nos Sistemas de Lógica Nebulosa (SLN), representados pela figura 1, em geral, são fornecidas entradas precisas para um módulo ***fuzificador*** que, por sua vez, fornece parâmetros nebulosos para um motor de inferência, o qual processa a aplicação de uma regra do tipo **SE-ENTÃO**, constituída de proposições, envolvendo termos de variáveis lingüísticas. Após o processamento de uma regra, o valor nebuloso, obtido como resposta da inferência, é ***desfuzificado***, obtendo-se, dessa forma, a saída ***crisp*** do sistema.




**<u>Figura 1</u>:** Sistema de Lógica Nebulosa.

Um parâmetro precisas pode ser ***fuzificado*** seguindo um método simples, trapezoidal, triangular ou gaussiano; já a ***desfuzificação*** pode ser ser feita através do valor máximo, média de valores máximos ou por meio do cálculo do centróide. No que diz respeito à inferência, essa pode seguir a regra de ***modus ponens generalizado*** ou composição de relações nebulosas.

Relações nebulosas são subconjuntos nebulosos do produto cartesiano $\boldsymbol{X} \times \boldsymbol{Y}$. Essas relações possuem um papel importante dentro da teoria de conjuntos nebulosos, sendo aplicadas na resolução de diversos problemas. A definição das relações nebulosas é dada a seguir:

Sejam $\boldsymbol{X},\boldsymbol{Y} \subseteq \boldsymbol{R}$ os conjuntos dos universos de discursos, então

$$\tilde{R}=\{((x,y),\mu_{\tilde{R}}(x,y))|(x,y)\in X\times Y\} \tag{4}$$

é chamada de uma relação nebulosa em $\boldsymbol{X} \times \boldsymbol{Y}$.

Uma das operações mais úteis definidas sobre as relações nebulosas é a chamada composição **max-min** que combina relações nebulosas de produtos de espaços diferentes.

Sejam $\tilde{R}_1(x,y),(x,y)\in X\times Y$ e $\tilde{R}_2(y,z),(y,z)\in Y\times Z$ duas relações nebulosas. A composição max-min entre $\tilde{R}_1$ e $\tilde{R}_2$ é dada pelo seguinte conjunto nebuloso:

$$\tilde{R}_1 \circ \tilde{R}_2 = \{[(x,z), \max_y \{min\{\mu_{\tilde{R}_1}(x,y), \mu_{\tilde{R}_2}(y,z)\}\}] | x \in X, y \in Y, z \in Z\} \tag{5}$$

O termo $\max_y \{min\{\mu_{\tilde{R}_1}(x,y), \mu_{\tilde{R}_2}(y,z)\}\}$ é a função de pertinência da composição de relações nebulosas.

Uma idéia básica na Lógica Nebulosa é a de que uma proposição **P** em uma linguagem natural ou artifical pode ser vista como uma coleção de restrições elásticas [11], restringindo os valores de uma coleção de variáveis $\boldsymbol{x} = (\boldsymbol{x_1}, \boldsymbol{x_2}, ...., \boldsymbol{x_n})$ definidas nos universos $\mathbf{U^n}$. Geralmente as variáveis, bem como, as restrições aparecem em **P** de maneira implícita. Visto nessa perspectiva, a representação do conhecimento de **P** consiste, na essência, no processo de tornar explícitas as variáveis e restrições. Isso é obtido, representando **P** através de sua forma canônica $\mathbf{P} \rightarrow \boldsymbol{x}$ É $\mathbf{A}$, onde $\mathbf{A}$ é um predicado nebuloso.

# 3 TÉCNICA DE BUSCA HEURÍSTICA

As técnicas de busca Heurística podem ser usadas na verificação de protocolos de comunicação, a fim de otimizar a expansão do seu grafo de execução através da aplicação de critérios que reduz os caminhos para a obtenção de objetivos que provem uma dada propriedade de interesse do protocolo.

A seguir, é apresentado em pseudo-código, o algoritmo **A*** [12] de busca heurística utilizada pela técnica de verificação descrita na próxima seção. O Algoritmo **A*** foi escolhido como base para a técnica de verificação por fornecer o melhor caminho para o nó objetivo. Isso é importante na medida em que os nós que compõem os grafos dos protocolos de comunicação, normalmente, possuem uma grande quantidade de informação, necessitando, portanto, de muito espaço em memória/disco para armazenamento o que pode suscitar um maior tempo no processamento da verificação.

```
ALGORITMO A*

{ seja f' = g + h' a estimativa da função custo f  que um nó x possui                }
{ para atingir o nó meta                                                              }
{ g é o custo do nó raiz ao nó x e h' é o custo que se espera de x ao nó meta    }

VARIÁVEIS:
ESTRUTURA NÓ_SUC =        pSuc              : PONTEIRO PARA NÓ;
                          próximo           : PONTEIRO PARA NÓ_SUC;
FIM {ESTRUTURA NÓ_SUC}

ESTRUTURA NÓ =      dados            : tipo_DADOS_do_nó;
                    g                : INTEIRO;
                    h'               : INTEIRO;
                    ptrSucessores    : PONTEIRO PARA NÓ_SUC;
                    ptrPai           : PONTEIRO PARA NÓ;
                    próximo          : PONTEIRO PARA NÓ;
FIM {ESTRUTURA NÓ_SUC}

AcheiObjetivo                        : BOOLEANO;
VELHO, MELHOR_NÓ, NÓ_RAIZ            : PONTEIRO PARA NÓ;
primABERTO, primFECHADO              : PONTEIRO PARA NÓ;
SUCESSORES, SucAux                   : PONTEIRO PARA NÓ_SUC;

INICIO
Ler (NÓ_RAIZ);
AcheiObjetivo <- FALSO;
primFECHADO <- nil;
primABERTO <- NÓ_RAIZ;
NÓ_RAIZ^.g <- 0;
NÓ_RAIZ^.próximo <- nil;
NÓ_RAIZ^.ptrSucessores <- nil;
REPETIR
MELHOR_NÓ <- primABERTO;
primABERTO <- primABERTO^.próximo;
MELHOR_NÓ^.próximo <- primFECHADO;
primFECHADO <- MELHOR_NÓ;
AcheiObjetivo <- É_Ojetivo (MELHOR_NÓ);
SE NÃO AcheiObjetivo
ENTÃO
SUCESSORES <- GerarSuc (MELHOR_NÓ);
SucAux <- SUCESSORES;
```

```
ENQUANTO SucAux # nil FAÇA
TRATA_SUCESSORES (SUCESSORES);
SucAux <- SucAux^.próximo;
FIMENQUANTO
FIMSE
ATÉ AcheiObjetivo OU (primABERTO = nil);
FIM {A*}

SUBALGORITMO TRATA_SUCESSORES (Suc : PONTEIRO PARA NÓ);

VARIÁVEIS:
Estou_em_aberto, Estou_em_fechado : BOOLEANO;

INÍCIO
Estou_em_aberto <- FALSO;
Estou_em_fechado <- FALSO;
Suc^.ptrPai <- MELHOR_NÓ;
Suc^.g <- MELHOR_NÓ^.g + 1;
VELHO <- PEGAR_NÓ_IGUAL_NA_FILA (ABERTO,Suc);
SE VELHO # nil
ENTÃO
Estou_em_aberto <- VERDADEIRO;
Inserir_nos_sucessores (MELHOR_NÓ,Suc);
SE Suc^.g < VELHO^.g
ENTÃO
VELHO^.g <- Suc^.g;
VELHO^.ptrPai <- MELHOR_NÓ;
FIMSE
FIMSE
SE NÃO Estou_em_aberto
ENTÃO
VELHO <- PEGAR_NÓ_IGUAL_NA_FILA (FECHADO,Suc);
SE VELHO # nil
ENTÃO
Estou_em_fechado <- VERDADEIRO;
Inserir_nos_sucessores (MELHOR_NÓ,Suc);
SE Suc^.g < VELHO^.g
ENTÃO
VELHO^.g <- Suc^.g;
VELHO^.ptrPai <- MELHOR_NÓ;
Atualizar_custo_dos_sucessores (VELHO);
FIMSE
FIMSE
FIMSE
SE NÃO Estou_em_aberto E NÃO Estou_em_fechado
ENTÃO
Inserir_nos_sucessores (MELHOR_NÓ,Suc);
Colocar_ordenado_em_aberto (Suc);
FIMSE
FIM {TRATA_SUCESSORES}
```

# 4 UMA TÉCNICA DE VERIFICAÇÃO

A técnica para verificação de Modelos de Estados aqui apresentada, consiste em uma busca heurística no grafo de execução do protocolo sob estudo, objetivando encontrar um nó que acarretaria a correção de uma dada propriedade de interesse. Tal busca é feita segundo o algortimo **A***, modificado para operar com variáveis lingüísticas da Lógica Nebulosa.

No algoritmo **A***, mostrado na seção anterior, o passo (ou subalgoritmo) da linha 83 **Colocar_ordenado_em_aberto (Suc)** tem por objetivo armazenar os nós recentemente criados na fila de nós abertos. A fila de ABERTOS, no algoritmo **A*** original, mantém os nós ordenados de maneira crescente, segundo os valores de **f' = g + h',** de tal forma que o primeiro nó da fila é aquele que possui o menor custo para o objetivo. Na técnica ora proposta, a fila de ABERTOS será ordenada por intermédio do processamento de regras de um Sistema de Lógica Nebulosa (SLN).

O bom desempenho do algoritmo **A*** está calcado nas estimativas para o termo **h'** da função custo **f'**. A estimativa de **h'** é feita a partir do conhecimento específico do problema em análise. No caso em questão, deve-se adotar o conhecimento genérico sobre protocolos de comunicação.

A maior dificuldade presente na utilização do algoritmo **A***, é estabelecer a estimativa do valor de **h'** em cada nó, devido à eventual complexidade de tradução do conhecimento específico relacionado ao problema tratado; e, à sistematização (padronização) imprecisa desse conhecimento para determinados sistemas.

O emprego da Lógica Nebulosa facilita a descrição do conhecimento dos protocolos de comunicação a ser utilizado pelo algoritmo **A***, pois trata-se de um arcabouço matemático próprio para lidar com raciocínio aproximado.

A técnica de verificação será aplicada em um determinado protocolo de comunicação, doravante Sistema sob Estudo (**SE**), o qual deverá ser especificado por uma Técnica de Descrição Formal que se baseie em um modelo de estados e transições.

A especificação de **SE** deverá ser comparada à especificação de um outro sistema, aqui denominado Sistema Padrão (**SP**), considerado como um "bom" paradigma para **SE**. Pode-se entender **SP** como uma base de conhecimento genérica para auxiliar o projeto dos protocolos de comunicação.

A técnica prevê a utilização de algumas versões de **SP**, ou seja, vários paradigmas de diversas categorias de protocolos, a fim de que o usuário de um verificador, calcado nessa mesma técnica, possa definir qual dos referidos paradigmas mais se assemelha ao seu **SE**.

Portanto, é a partir do estabelecimento das semelhanças entre os objetos do **SE** e do **SP**, feito pelo usuário de forma nebulosa, que começa o emprego da técnica de verificação. Em outras palavras, tem-se:

- **E** = $\{e_1, e_2, ...., e_{n1}\}$ é o conjunto (***crisp***) dos estados parciais de **SE**;
- $\mathcal{E}$ = $\{\epsilon_1, \epsilon_2, ...., \epsilon_{n2}\}$ é o conjunto (***crisp***) dos estados parciais de **SP;**

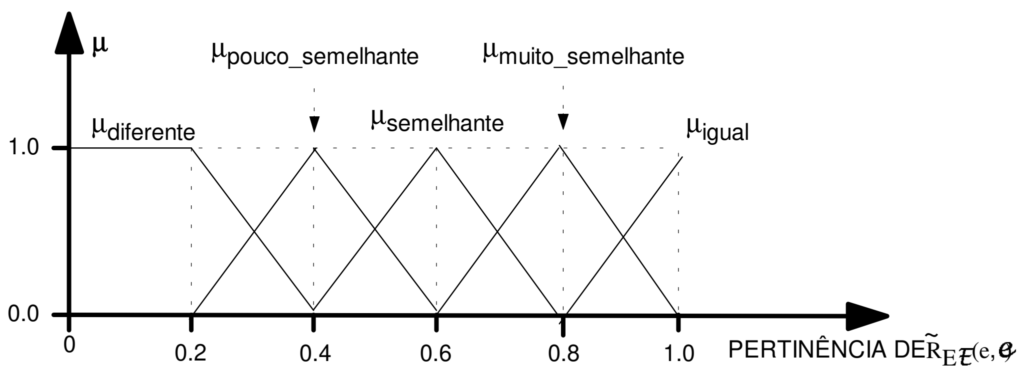
A partir de então, define-se a relação nebulosa: $\tilde{R}_{E\mathcal{E}}(e,\epsilon)$ que caracteriza a medida de semelhança entre os estados parciais do **SE** e do **SP**. Além dessa relação e da especificação do **SE**, é dado também um nó objetivo $\eta_o$ a ser alcançado pelo algoritmo **A*** nebuloso (**Ã***) a partir do nó raiz $\eta_R$ do grafo de execução de **SE**.

Uma vez que **SE** não possui, a princípio, a mesma quantidade de estados que **SP**, isto é, os sub-índices ***n1*** e ***n2*** são diferentes entre si, então para se obter um estado parcial $\epsilon$ em **SP** que corresponda a um estado parcial **e** em **SE**, deve ser feita a seguinte composição de relações nebulosas:

$$\tilde{\mathcal{O}}(\epsilon) = \tilde{O}(\mathbf{e}) \circ \tilde{R}_{E\mathcal{E}}(e,\epsilon) \qquad (6)$$

onde **Õ(e)** é um dos estados do **SE**, $\mathbf{e_i}$, ***fuzificado*** pelo método simples; e, $\tilde{\mathcal{O}}(\epsilon)$ é o estado parcial nebuloso em **SP,** equivalente a **Õ(e)** em **SE**, que, após ser ***desfuzificado*** pelo método do máximo, fornece o objeto $\epsilon_k$ pertencente ao conjunto $\mathcal{E}$.

A equação (6) deve ser aplicada para todos os estados parciais do **SE** a fim de se obter os seus equivalentes em **SP**. Todavia essa equivalência, em geral, não será perfeita e, portanto, deverá ser medida para compor as regras de inferência de um SLN. A medida da equivalência dar-se-á pela variável lingüística ***semelhança***, cujo universo de discurso **U** é composto pelos valores da função de pertinência da relação $\tilde{R}_{E\mathcal{E}}(e,\epsilon)$. Na figura 2, pode-se observar a definição dessa variável lingüística.




<u>**Figura 2**</u>**:** Funções de pertinência dos termos da variável lingüística ***semelhança***.

A medida da equivalência é fundamentalmente importante, pois a mesma pode ficar ainda mais prejudicada devido ao agrupamento dos estados parciais para compor os estados globais. Suponha que o **SE** e o **SP** tenham duas entidades comunicantes e que um dado estado global de **SE** seja o par ($\mathbf{e_1},\mathbf{e_2}$). Após aplicar-se a fórmula (6) para cada um dos estados parciais $\mathbf{e_1}$ e $\mathbf{e_2}$, chegou-se a conclusão de que os seus equivalentes, deveriam ser $\mathbf{e_3}$ e $\mathbf{e_5}$ no **SP**. No entanto, o par ($\mathbf{e_3}$ , $\mathbf{e_5}$) não constitui um estado global em **SP** o que impediria a obtenção de um custo para o nó objetivo em **SP**. Logo, deve-se descartar os valores previamente ***desfuzificados*** $\mathbf{e_3}$ ou $\mathbf{e_5}$ , e pegar como equivalentes do **SE**, os próximos estados parciais de **SP** com valores de pertinência imediatamente abaixo do máximo e assim por diante até se obter um par de estados globais em **SP** que seja um estado global e ainda seja o mais próximo de ($\mathbf{e_1},\mathbf{e_2}$).

O subalgoritmo da linha 83 **Colocar_ordenado_em_aberto (Suc)** do algoritmo **Ã*** aloca os nós na fila de ABERTOS a partir de um SLN que manipula termos de uma outra variável lingüística, chamada ***posição*** [11], cuja definição aparece esquematicamente na figura 3. Nessa figura, observa-se o universo de discurso da variável caracterizado pelo percentual de nós na fila de ABERTOS, pois, conforme **Ã*** evolui, a quantidade de estados globais obtidos aumenta, sendo então necessário, inicialmente, ***desfuzificar*** um percentual para depois calcular a posição ***crisp*** atual.

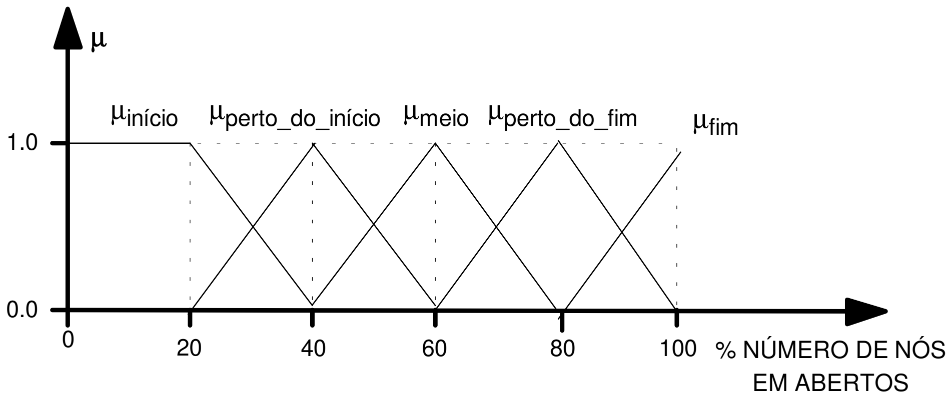



**Figura 3:** Funções de pertinência dos termos da variável lingüística ***posição***.

Além das variáveis lingüísticas ***semelhança*** e ***posição***, é definida também a variável ***custo*** (ver figura 4), que atribui termos nebulosos ao custo do nó do **SP,** equivalente ao nó **Suc** (um dos sucessores do nó em análise) do **SE**. O universo de discurso da variável ***custo*** é um conjunto cujos elementos são constituídos do número de passos de computação (ou disparo de transições). Uma vez que o número de passos de computação máximo de **SE** (sem repetições) não é conhecido, foi escolhido, como valor final do intervalo que compõe o universo de discurso de **U**, o maior caminho (sem repetições) de **SP** mais 60%, pois geralmente **SE** é maior que **SP**.

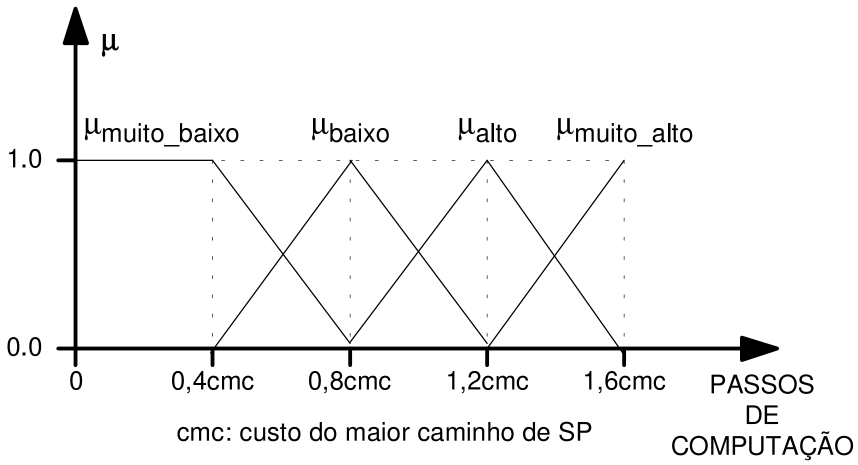



**Figura 4:** Funções de pertinência dos termos da variável lingüística ***custo***.

O grafo de execução do **SP** é previamente conhecido, tendo para cada um de seus nós, o registro do seu custo à raiz (***Cr***). Logo, se $\sigma_o$ for o nó do **SP** equivalente ao nó objetivo $\eta_o$ do **SE**, e se $\sigma_s$ for o nó do **SP** equivalente ao nó $\eta_s$ sucessor de **MELHOR_NÓ** (vide A*) do **SE**, então o custo de $\sigma_s$ ao nó $\sigma_o$ (***Co***) é dado por:

$$Co(\sigma_s) = Cr(\sigma_o) - Cr(\sigma_s) \tag{7}$$

O custo do nó em **SP,** equivalente ao nó sucessor para o objetivo do **SE**, deve ser ***fuzificado*** segundo o método do triângulo. Esse é então comparado às regras de inferência de um SLN para se deduzir a posição nebulosa do nó sucessor de **MELHOR_NÓ** na fila de nós abertos. Antes de serem executadas as regras do SLN que conduz à ordenação da fila de ABERTOS (SLN B), deve-se processar um outro SLN (SLN A) que obtenha a semelhança entre os estados globais ou nós do **SE** e do **SP**. As regras desse último SLN são mostradas a seguir para o caso de duas entidades comunicantes, onde aparecem: $\mathbf{e_i}$ e $\mathbf{e_j}$, compondo um estado global do **SE**; e, $\mathbf{e_k}$ e $\mathbf{e_l}$, compondo o estado global do **SP** equivalente ao do **SE**.

**<u>Regras do SLN A que acha a semelhança entre os estados globais do SE e do SP</u>**

⇒ SE ({$e_i$ , $e_k$}É igual) E ({$e_j$ , $e_l$}É igual) ENTÃO {($e_i$ , $e_j$) , ($e_k$, $e_l$)} É igual
⇒ SE (({$e_i$ , $e_k$}É igual) OU ({$e_i$ , $e_k$}É muito_semelhante)) E
(({$e_j$ , $e_l$}É igual) OU ({$e_j$ , $e_l$}É muito_semelhante)) ENTÃO
{($e_i$ , $e_j$) , ($e_k$, $e_l$)} É muito_semelhante
⇒ SE ({$e_i$ , $e_k$}É igual) E ({$e_j$ , $e_l$}É semelhante) ENTÃO
{($e_i$ , $e_j$) , ($e_k$, $e_l$)} É muito_semelhante
⇒ SE ({$e_i$ , $e_k$}É semelhante) E ({$e_j$ , $e_l$}É igual) ENTÃO
{($e_i$ , $e_j$) , ($e_k$, $e_l$)} É muito_semelhante
⇒ SE ({$e_i$ , $e_k$}É igual) E
(({$e_j$ , $e_l$}É pouco_semelhante) OU ({$e_j$ , $e_l$}É diferente)) ENTÃO
{($e_i$ , $e_j$) , ($e_k$, $e_l$)} É semelhante

⇒ SE (({$e_i$ , $e_k$ }É pouco_semelhante) OU ({$e_i$ , $e_k$ }É diferente)) E
({$e_j$ , $e_l$ }É igual) ENTÃO {($e_i$ , $e_j$) , ($e_k$, $e_l$)} É semelhante

⇒ SE (({$e_i$ , $e_k$ }É semelhante) OU ({$e_i$ , $e_k$ }É muito_semelhante)) E
(({$e_j$ , $e_l$ }É semelhante) OU ({$e_j$ , $e_l$ }É muito_semelhante))
ENTÃO {($e_i$ , $e_j$) , ($e_k$, $e_l$)} É semelhante

⇒ SE (({$e_i$ , $e_k$ }É semelhante) OU ({$e_i$ , $e_k$ }É pouco_semelhante)) E
(({$e_j$ , $e_l$ }É semelhante) OU ({$e_j$ , $e_l$ }É pouco_semelhante))
ENTÃO {($e_i$ , $e_j$) , ($e_k$, $e_l$)} É pouco_semelhante

⇒ SE (({$e_i$ , $e_k$ }É diferente) OU ({$e_i$ , $e_k$ }É pouco_semelhante)) E
(({$e_j$ , $e_l$ }É diferente) OU ({$e_j$ , $e_l$ }É pouco_semelhante))
ENTÃO {($e_i$ , $e_j$) , ($e_k$, $e_l$)} É diferente

As regras de inferência do SLN B para ordenar a fila de ABERTOS do algoritmo **Ã*** são apresentadas logo abaixo, onde $\boldsymbol{\eta_s}$ é o estado sucessor de **MELHOR_NÓ** do **SE**, equivalente ao nó $\boldsymbol{\sigma_s}$ do **SP**; $\tilde{C}o(\sigma_s)$ é o custo nebuloso de $\boldsymbol{\sigma_s}$ ao nó $\boldsymbol{\sigma_o}$ (objetivo no **SP**); e, $\tilde{P}(\eta_s)$ é a posição nebulosa de $\boldsymbol{\eta_s}$ na fila de ABERTOS.

**<u>Regras do SLN B que ordena a fila de ABERTOS do algoritmo Ã*</u>**

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É igual) E ($\tilde{C}o(\sigma_s)$ É muito_baixo) ENTÃO $\tilde{P}(\eta_s)$ É início

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É igual) E ($\tilde{C}o(\sigma_s)$ É baixo) ENTÃO $\tilde{P}(\eta_s)$ É perto_do_início

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É igual) E ($\tilde{C}o(\sigma_s)$ É alto) ENTÃO $\tilde{P}(\eta_s)$ É meio

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É igual) E ($\tilde{C}o(\sigma_s)$ É muito_alto) ENTÃO $\tilde{P}(\eta_s)$ É fim

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É muito_semelhante) E ($\tilde{C}o(\sigma_s)$ É muito_baixo)
ENTÃO $\tilde{P}(\eta_s)$ É início

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É muito_semelhante) E ($\tilde{C}o(\sigma_s)$ É baixo) ENTÃO
$\tilde{P}(\eta_s)$ É perto_do_início

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É muito_semelhante) E ($\tilde{C}o(\sigma_s)$ É alto) ENTÃO $\tilde{P}(\eta_s)$ É meio

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É muito_semelhante) E ($\tilde{C}o(\sigma_s)$ É muito_alto)
ENTÃO $\tilde{P}(\eta_s)$ É perto_do_fim

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É semelhante) E ($\tilde{C}o(\sigma_s)$ É muito_baixo)
ENTÃO $\tilde{P}(\eta_s)$ É perto_do_início

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É semelhante) E ($\tilde{C}o(\sigma_s)$ É baixo) ENTÃO $\tilde{P}(\eta_s)$ É meio

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É semelhante) E ($\tilde{C}o(\sigma_s)$ É alto) ENTÃO $\tilde{P}(\eta_s)$ É perto_do_fim

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É semelhante) E ($\tilde{C}o(\sigma_s)$ É muito_alto) ENTÃO $\tilde{P}(\eta_s)$ É fim

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É pouco_semelhante) E ($\tilde{C}o(\sigma_s)$ É muito_baixo) ENTÃO
$\tilde{P}(\eta_s)$ É meio

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É pouco_ semelhante) E ($\tilde{C}o(\sigma_s)$ É baixo)
ENTÃO $\tilde{P}(\eta_s)$ É perto_do_fim

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É pouco_ semelhante) E ($\tilde{C}o(\sigma_s)$ É alto) ENTÃO $\tilde{P}(\eta_s)$ É fim

⇒ SE ({$\boldsymbol{\eta_s}$ , $\boldsymbol{\sigma_s}$}É pouco_ semelhante) E ($\tilde{C}o(\sigma_s)$ É muito_alto)
ENTÃO $\tilde{P}(\eta_s)$ É fim

$\Rightarrow$ SE $\{\eta_s , \sigma_s\}$É diferente ENTÃO $\tilde{P}\,(\eta_s)$ É fim

# 5 EXEMPLO DE APLICAÇÃO

Para aplicar a técnica de verificação será utilizado um protocolo para transmissão de dados multiponto-multiponto para texto-conferência [13]. O serviço de comunicação desse protocolo oferece as seguintes facilidades aos seus usuários: meios de estabelecimento de uma conexão entre as estações participantes com fim de trocar SDUs (***service data unit***); meios para inserção, de uma ou mais estações como participante em uma conexão já estabelecida de forma negociada; ou meios para liberação, negociada ou não, de uma ou mais estações; e meios para transferência de SDUs durante a conexão.

Uma vez que o protocolo multiponto-multiponto de [13] é extenso, será visto, nesse exemplo, apenas o serviço de estabelecimento de conexão, cujas primitivas são mostradas na tabela 1 e PDUs (***protocol data unit***), na tabela 2.

**Tabela 1**: Primitivas de estabelecimento de conexão do protocolo multiponto-multiponto.

| ***PRIMITIVA*** | ***PARÂMETROS*** |
|---|---|
| MPDL_ESTABLISH.Request | -------- |
| MPDL_ESTABLISH.Indication | Identificação da estação origem. |
| MPDL_ESTABLISH.Confirm | Modo da conexão e identificação das estações conectadas. |

**Tabela 2**: PDUs de estabelecimento de conexão do protocolo multiponto-multiponto.

| ***Código da MPPDU*** | ***Nome da MPPDU*** |
|---|---|
| SMPARM | Set MultiPoint Asynchronous Request Mode |
| MPARMAC | MultiPoint Asynchronous Request Mode ACcept |
| PLIST | Present LIST |

O protocolo multiponto-multiponto permite que uma estação estabeleça conexão, iniciando uma conferência (estação mestra) ou entrando em uma conferência já em andamento (estação receptora). Em ambos os casos é acionada, pelo usuário do protocolo, a primitiva de requisição de conexão MPDL_ESTABLISH.Request, sendo, em seguida, enviada uma MPPDU SMPARM para o endereço coletivo (***multicast***). A partir de então, cada estação do grupo, indica, ao seu usuário, através da primitiva MPDL_ESTABLISH.Indication, que existe uma nova participante, confirmando o recebimento com a MPPDU MPARMAC. Somente após o recebimento de todas as confirmações, o usuário que iniciou a conferência recebe a primitiva MPDL_ESTABLISH.Confirm.

Os estados parciais do serviço de estabelecimento de conexão do protocolo multiponto a multiponto são apresentados na tabela3 a seguir:

**Tabela 3**: Estados do estabelecimento de conexão do protocolo multiponto-multiponto.

| ***Estado Parcial*** | ***Descrição*** |
|---|---|
| CLOSED | A conexão multiponto está fechada e, no que diz respeito à estação, não há conferência em andamento. |
| ESTABLISH | Foi enviado um pedido de conexão, e a estação aguarda a confirmação de todas as possíveis estações participantes. |
| ESTABLISH RECEIVER | Uma estação pode entrar nesse estado em duas condições: se após receber uma primitiva de requisição de conexão já estabelecida e que, por isso, a estação deverá entrar na conferência como receptora. O segundo motivo da entrada de uma estação que esteja em recepção nesse estado é o recebimento de uma PDU PLIST, que contém a lista de participantes da conferência, sem a sua identificação. Isso configura que a estação não está na conferência. Em resposta a esse acontecimento é enviado um pedido de conexão para a estação Mestra e, em seguida, a estação vai para esse estado esperar a confirmação da conexão, para poder voltar a operar como receptora. |
| OPEN RECEIVER | Nesse estado só é permitida a recepção de dados. |
| OPEN MASTER | Nesse estado a estação é a Mestra da conferência, podendo transmitir dados via canal principal. |

Na figura 5, encontra-se representada a especificação com rede de Petri do trecho da abertura de conexão do protocolo multiponto-multiponto.

Uma vez que foi introduzido o **SE**, como sendo o protocolo multiponto-multiponto, agora pode ser observado na figura 6, a rede de Petri que modela uma entidade com o comportamento iniciador do serviço de abertura de conexão do **SP**; na figura 7, o modelo da mesma entidade com o comportamento respondedor do serviço; e, na figura 8, o modelo do meio de comunicação entre as entidades iniciadora e respondedora da conexão do **SP**.

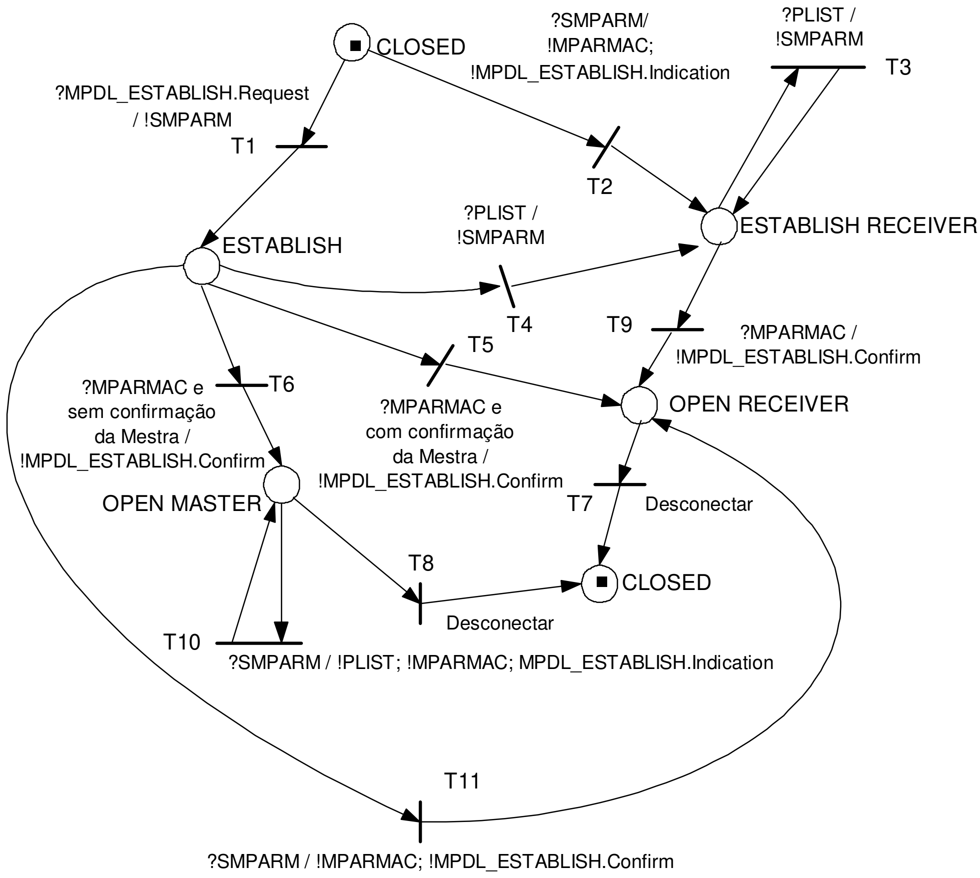



**Figura 5:** Rede de Petri de uma entidade do protocolo Multiponto-Multiponto.

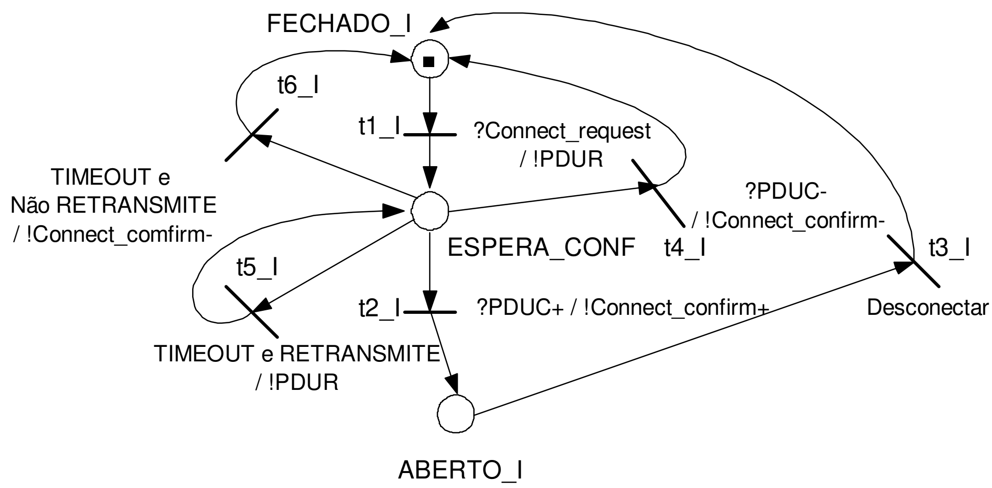



**Figura 6:** Rede de Petri de uma entidade iniciadora da abertura de conexão genérica.

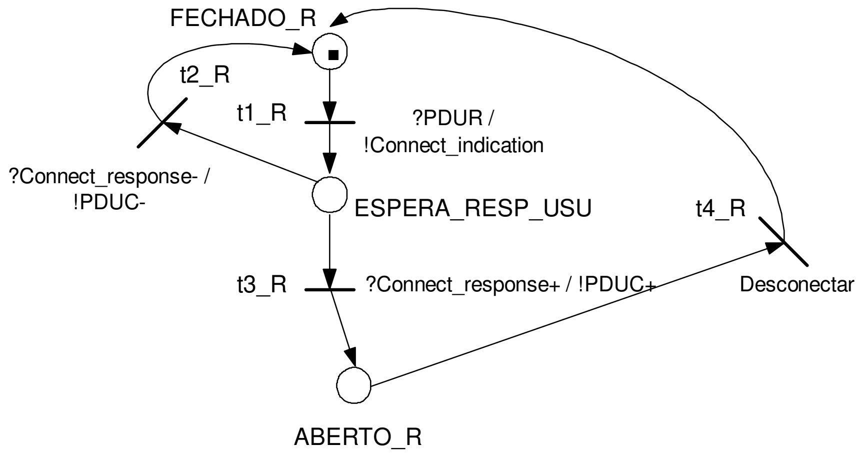



**Figura 7:** Rede de Petri de uma entidade respondedora da abertura de conexão genérica.

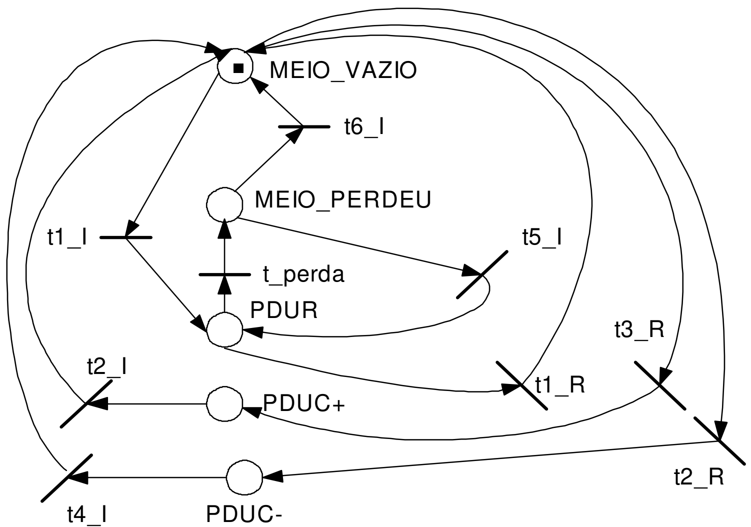



**Figura 8:** Rede de Petri do meio de comunicação do serviço de abertura de conexão genérico.

Aplicando-se agora a técnica de verificação, devemos inicialmente estabelecer a relação nebulosa $\tilde{R}_{E_{\mathcal{E}}}(e, \boldsymbol{e})$ e um estado global objetivo $\eta_o$. Antes disso, convém destacar que os estados parciais FECHADO_I e FECHADO_R, assim como, ABERTO_I e ABERTO_R do **SP** são os mesmos no caso de suas entidades apresentarem os dois comportamentos para a conexão, como é o caso da entidade do protocolo multiponto-multiponto que compõe o **SE**.

Sejam:

**E** = {*CLOSED, ESTABLISH, ESTABLISH RECEIVER, OPEN RECEIVER, OPEN MASTER*};
$\mathsf{E}$ = {*FECHADO, ESPERA_CONF, ABERTO, ESPERA_RESP_USU*};

$$\tilde{R}_{E_{\mathsf{E}}}(e, \mathit{e}) = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.6 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.8 & 0.0 \\ 0.0 & 0.0 & 0.8 & 0.0 \end{bmatrix};$$

$\eta_o$ = (*OPEN MASTER, OPEN RECEIVE, OPEN RECEIVE, OPEN RECEIVE*).

Na matriz $\tilde{R}_{E_{\mathsf{E}}}(e, \mathit{e})$, as linhas caracterizam os elementos de **E** na ordem em que se vê acima, enquanto as colunas são os elementos de $\mathsf{E}$, também na mesma ordem em que se vê acima. Logo, cada elemento de $\tilde{R}_{E_{\mathsf{E}}}(e, \mathit{e})$ é o grau de equivalência entre um elemento de **E** e um elemento de $\mathsf{E}$, por exemplo, o valor **0.8** na quarta linha e terceira coluna é a equivalência ou semelhança entre *OPEN RECEIVER* E *ABERTO*.

O estado global objetivo $\eta_o$ dado caracteriza uma conferência, em andamento, com quatro estações conectadas, de maneira que todas as estações devem chegar a esses estados parciais a partir de um nó raiz, onde as mesmas encontram-se no estado *CLOSED*. Apesar do **SP** apresentado ter sido idealizado para uma arquitetura somente com duas entidades comunicantes, ainda assim, ele pode ser utilizado como paradigma do protocolo multiponto a multiponto, pois basta que **SP** seja uma aproximação do **SE** e possua alguns dos elementos básicos de seu serviço.

Fazendo-se então a ***fuzificação*** pelo método simples dos elementos de **E**; procedendo com a composição de relações dada pela equação (6); e, logo em seguida, obtendo-se a ***desfuzificação*** de $\tilde{O}(\mathit{e})$, chega-se a conclusão que $\sigma_o$ = (*ABERTO, ABERTO*), ou seja, um par, pois o grafo de **SP** que deve ser previamente conhecido é de um sistema com duas entidades.

De posse de $\sigma_o$, o algoritmo **A*** começa a funcionar com o primeiro nó de ABERTOS, que é a raiz $\eta_r$ e, o seu custo à $\eta_o$, deve ser estimado a partir de **SP**. Na listagem abaixo é fornecido o grafo de execução de **SP** feito com o pograma ARP [14], onde pode-se observar, que o estado **M6** é obtido após **4** disparos de transição. Pode-se ver também, na mesma listagem, que o estado **M6** caracteriza o objetivo em **SP**, ou seja, $\sigma_o$ = (*ABERTO, ABERTO*).

**Grafo de acessibilidade da rede de SP:**

```
*------------------------------------------------------*
M0  :(t1_I: M1)
M1  :(t_perda: M2) (t1_R: M3)
M2  :(t5_I: M1) (t6_I: M0)
M3  :(t2_R: M4) (t3_R: M5)
M4  :(t4_I: M0)
M5  :(t2_I: M6) (t4_R: M11)
M6  :(t3_I: M7) (t4_R: M10)
M7  :(t1_I: M8) (t4_R: M0)
```

**M8 :(t_perda: M9) (t4_R: M1)**
**M9 :(t5_I: M8) (t6_I: M7) (t4_R: M2)**
**M10 :(t3_I: M0)**
**M11 :(t2_I: M10)**
***------------------------------------------------------***

**Estados acessiveis pela rede de SP:**
***------------------------------------------------------***
**M0 :{FECHADO_I, FECHADO_R, MEIO_VAZIO}**
**M1 :{FECHADO_R, ESPERA_CONF, PDUR}**
**M2 :{FECHADO_R, ESPERA_CONF, MEIO_PERDEU}**
**M3 :{MEIO_VAZIO, ESPERA_CONF, ESPERA_RESP_USU}**
**M4 :{FECHADO_R, ESPERA_CONF, PDUC-}**
**M5 :{ABERTO_R, ESPERA_CONF, PDUC+}**
**M6 :{MEIO_VAZIO, ABERTO_I, ABERTO_R}**
**M7 :{FECHADO_I, MEIO_VAZIO, ABERTO_R}**
**M8 :{ABERTO_R, ESPERA_CONF, PDUR}**
**M9 :{ABERTO_R, ESPERA_CONF, MEIO_PERDEU}**
**M10 :{FECHADO_R, MEIO_VAZIO, ABERTO_I}**
**M11 :{FECHADO_R, ESPERA_CONF, PDUC+}**
***------------------------------------------------------***

No grafo de **SP**, pode-se ver ainda que o custo do maior caminho é **7** ao atingir o estado **M9**, pois as demais marcações são alcançadas com menos disparos de transição. Logo, o valor máximo para o eixo das abscissas na figura 4, seria (1.6 x 7 = 11.2) e, nesse caso, o custo **4** da raiz ao objetivo em **SP** fica entre **muito_baixo** e **baixo**.

O algoritmo **A*** começa a sua execução, obtendo os sucessores de $\eta_r$ que então, serão ***fuzificados*** para se obter os seus equivalentes em **SP** que após entrar no SLN A, infere-se a semelhança entre os estados globais sucessores, passando, em seguida, para o SLN B a fim de determinar a posição desses sucessores da raiz na fila de ABERTOS. O algoritmo A* prossegue, pegando os sucessores do primeiro nó da fila de ABERTOS e assim por diante.

# 6 COMENTÁRIOS FINAIS

A comparação do **SE** com um **SP**, livra o usuário de codificar o seu conhecimento a respeito do projeto de protocolos de tal forma que a utilização de um algoritmo de busca informado, como o **A***, pode ser implementado de maneira mais automática. O emprego da Lógica Nebulosa na formalização dessa comparação, entre o **SE** e o **SP**, mostrou-se promissor pelo fato da mesma dispor de recursos adequados e conhecidos para lidar com aproximações, âmago da técnica de verificação.

A técnica de verificação, aqui apresentada, faz uma comparação entre o **SE** e o **SP**, levando-se em consideração apenas os estados parciais das suas respectivas especificações. No sentido de aperfeiçoar ainda mais o desempenho da técnica, é possível reescrever as regras de inferência do SLN B, incorporando a manipulação da medida de semelhança entre outros objetos das especificações do **SE** e do **SP** como, por exemplo: as transições; as variáveis locais; e, as mensagens nas filas.

Além da incorporação de novos objetos para comparar os dois sistemas (**SE** e **SP**), pode-se também construir a técnica com uma comparação de prefixos de computação ou trechos de caminhos dos grafos de **SE** e **SP**. Com isso, seria possível averiguar melhor o grau de semelhança entre os dois sistemas. Uma vez que a técnica de verificação, encontra-se em fase de desenvolvimento, tais possibilidades, ainda serão aplicadas e comparadas entre si.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS


[1] GOUDA, M. G. e HAN, J. Y., *Protocol validation by fair progress state exploration*, Computer Networks & ISDN System, 9, 1985.

[2] ZHAO, Z. R. e BOCHMANN, G. V., *Reduced reatchability analysis of communication protocols: A new approach*, Technical report, University Montreal, 1986.

[3] HSIEH, W. S., NAIN, T. S., YANG, M. S., LU, C. S., HUANG, K. C. e TSENG, J. R., *A Fast Method of Protocol Validation Using Reduced Stable State Exploration Technique*, Microprocessing and Microprogramming, North-Holland, 1991.

[4] JARD, C., *Valider les protocoles en simulant*, Colloque Francophone Sur L'Ingénierie des Protocoles - CFIP'88, Bordeaux, setembro, 1988.

[5] PAGEOT, J. M., *Une expérience de simulation guidée*, Colloque Francophone Sur L'Ingénierie des Protocoles - CFIP'88, Bordeaux, setembro, 1988.

[6] WEST, C. H., *Protocol validation - principles and applications*, Computer Networks and ISDN Systems, North-Holland, 1992.

[7] ZIMMERMANN, H. J., *Fuzzy Set Theory - and Its Applications*, Kluwer-Nijhoff Publishing, 1985.

[8] KANDEL, A., *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley Publishing Co., 1986.

[9] MENDEL, J. M., *Fuzzy Logic Systems for Engineering: A Tutorial*, Proceedings of the IEEE, vol. 83, nº 3, março, 1995.

[10] ZADEH, L. A., *Fuzzy Sets*, Information and Control, 1965.

[11] SILVA, A. P. da, *Escalamento Heurístico Baseado em Lógica Difusa*, Tese de Mestrado, Programa de Engenharia Elétrica, COPPE/UFRJ, 1995.

[12] RICH, E. e KNIGHT, K., *Artificial Intelligence*, Second Edition, International Edition, McGraw-Hill Inc., 1991.

[13] VIDAL, M. T. V. L., *Especificação Formal, Verificação e Implementação de um Protocolo de Comunicação Multiponto-a-Multiponto e uma Aplicação de Texto-Conferência*, Tese de Mestrado, Programa de Engenharia Elétrica, COPPE/UFRJ, 1994.

[14] ARP 2.3, Analisador de Redes de Petri, Tese de Mestrado, LCMI-EEL-UFSC.