

Uma Tradução de Redes de Petri Fuzzy Generalizadas para Gramáticas de Grafos com Atributos*

Júlia Krüger Vieira¹, Luciana Foss¹, Simone André da Costa Cavalheiro¹

¹Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação
Universidade Federal de Pelotas (UFPel) – Pelotas, RS – Brasil

{jkrueger, lfoss, simone.costa}@inf.ufpel.edu.br

Abstract. *In this paper, a method for translating a Generalized Fuzzy Petri Net to an Attributed Graph Grammar is presented. With this translation, the aim is to obtain a generic model to specify fuzzy systems which provides computational tools that allow to carry out property verifications and theorem proofs.*

Resumo. *Neste artigo é apresentado um método para traduzir uma Rede de Petri Fuzzy Generalizada para uma Gramática de Grafos com Atributos. Com esta tradução visa-se obter um modelo genérico para especificar sistemas fuzzy o qual disponibiliza ferramentas computacionais que permitem realizar verificações de propriedades e provas de teoremas.*

1. Introdução

No projeto descrito na dissertação apresentou-se uma tradução entre modelos partindo de um tipo de Rede de Petri Fuzzy (RPF) chamado de Generalizada (RPF_G) para se obter uma Gramática de Grafos com atributos (GGA) com o intuito de fornecer um novo método de especificação para sistemas fuzzy. Nesta tradução, obtém-se uma GGA Fuzzy que descreve o mesmo sistema de produções fuzzy que a RPF_G modela.

Dado que existem poucas ferramentas que permitem a análise de propriedades para sistemas modelados por RPF – são exemplos, a PACE e a PNeS que apenas simulam o sistema e analisam as propriedades referentes à estrutura da rede [GMBH 2008, Suraj 2018] – a contribuição central da tradução reside no fato de se obter um novo modelo que permite usar as ferramentas de análise disponíveis para GGA com o fim de provar propriedades referentes às classes das computações e realizar provas de teoremas utilizando a ferramenta Rodin [Cavalheiro et al. 2017, Cavalheiro 2010].

Para mostrar que a tradução está correta, deve-se garantir que a semântica foi preservada pela codificação, isto é, que a RPF_G e a GGA possuem comportamentos equivalentes.

A fim de verificar-se o estado da arte com relação ao tema abordado, foram analisados trabalhos sobre traduções já realizadas com outros tipos de redes de Petri e sobre Gramáticas de Grafos Fuzzy (GGs Fuzzy) que estão disponíveis. Ao que se refere às traduções já realizadas, a estratégia básica adotada é sempre a mesma, isto é, o grafo inicial da gramática é obtido a partir da marcação inicial da rede, cada transição gera uma

*O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

regra da gramática e, para gramáticas tipadas, o grafo tipo é composto pela representação de todos os lugares da rede. O que difere uma tradução da outra é o mapeamento dos elementos específicos presentes em cada tipo de Rede de Petri para a gramática gerada. Na tradução aqui proposta, a característica principal é a presença dos elementos e das operações da Lógica Fuzzy. Já com relação às GGs Fuzzy analisadas, as principais características destas gramáticas são considerar um tipo específico de Lógica Fuzzy e não permitir a escolha de diferentes operadores fuzzy. A GGA Fuzzy derivada da tradução proposta difere-se das demais por fornecer um modelo mais genérico que pode ser estendido com outros tipos de Lógica Fuzzy de forma simplificada, além disso, o novo modelo permite incluir situações não fuzzy na modelagem.

O restante deste trabalho está organizado da seguinte forma: as seções 2, 3 e 4 referem-se ao referencial teórico utilizado neste trabalho; na Seção 5 é apresentada a tradução propriamente dita; e a Seção 6 discorre sobre as considerações finais.

2. Lógica Fuzzy

A Lógica Fuzzy estende a lógica clássica inserindo a incerteza presente no mundo real. Assim, a função de pertinência é uma extensão da teoria dos conjuntos clássica que define o quanto um elemento pertence a um dado conjunto. Um conjunto fuzzy A em um universo de discurso U é caracterizado pela expressão $A = \{(x, u_A(x)) \mid x \in X\}$, onde $u_A : X \rightarrow [0, 1]$. Esta representação permite a extensão dos conjuntos clássicos provendo o mapeamento da pertinência no intervalo entre 0 e 1. Definições e conceitos referentes à Lógica Fuzzy e a sistemas fuzzy podem ser encontrados em [Jafelice et al. 2005].

Sistemas fuzzy são formularizados pelas etapas de fuzzificação, de inferência (base de regras fuzzy) e defuzzificação. Neste projeto, o foco é dado na etapa de inferência. Os sistemas fuzzy operam os elementos dos conjuntos fuzzy através dos conectivos fuzzy e das regras fuzzy. Os principais conectivos fuzzy são: as T-normas (E fuzzy), por exemplo, o mínimo $T_M(x, y) = \min(x, y)$ e o produto algébrico $T_P(x, y) = x \cdot y$; as T-conormas (OU fuzzy), por exemplo, de Lukasiewicz $S_{LK}(x, y) = \min(x + y, 1)$; e a negação fuzzy.

As regras fuzzy são produções do tipo *Se x é A então y é B* , computadas por uma operação de implicação fuzzy. O fato de se afirmar x é A ou y é B é representado por uma proposição fuzzy, por exemplo, a temperatura é baixa. As regras fuzzy podem ser classificadas em tipos considerando a posição dos conectivos (E/OU) fuzzy conforme a listagem a baixo [Chen et al. 1990].

- Tipo 1: R_i : **Se** d_j **então** d_k
- Tipo 2: R_i : **Se** d_{j1} E d_{j2} E... E d_{jn} **então** d_k
- Tipo 3: R_i : **Se** d_j **então** d_{k1} E d_{k2} E...E d_{kn}
- Tipo 4: R_i : **Se** d_{j1} OU d_{j2} OU...OU d_{jn} **então** d_k

Cada regra pode ser vinculada a um fator de certeza que indica o grau de credibilidade da mesma. Nesta abordagem, o fator de certeza é considerado na computação de cada regra [Reddy 2017].

3. Gramática de Grafos com Atributos

GGA é uma extensão da gramática de grafos convencional que comporta variáveis e operações em suas definições. A abordagem algébrica utilizada é a **Double Pushout** que

restringe a aplicação das regras utilizando a condição de colagem. Definições e conceitos referentes à GGAs podem ser encontradas em [Cavalheiro et al. 2017].

Cada GGA é definida com relação a uma especificação algébrica que descreve de forma abstrata os tipos de dados que podem ser utilizados nos seus componentes. Uma álgebra que implementa uma especificação algébrica *SPEC* é chamada de *SPEC*-álgebra. Os grafos com atributos são compostos por vértices, arestas (dirigidas) e atributos que permitem associar a cada vértice valores de uma *SPEC*-álgebra. Uma GGA é composta por um grafo tipo, um grafo inicial e um conjunto de regras. O grafo tipo define as restrições para a estrutura gráfica e para os tipos de valores dos atributos os quais podem ser usados durante a especificação do sistema na gramática. O grafo inicial é um grafo estado com atributos que define o estado inicial do sistema. Uma regra de transformação de grafos descreve como um grafo estado pode ser alterado para alcançar um novo estado na gramática. As regras são caracterizadas por um span de morfismos $r : L \xleftarrow{\alpha L} K \xrightarrow{\alpha R} R$, onde o grafo L é o lado esquerdo da regra (LHS) e define quais elementos devem estar presentes no grafo estado para que esta seja aplicada; o grafo K é a interface que define quais elementos devem ser preservados durante a aplicação da regra; e o grafo R é o lado direito da regra (RHS) indicando qual o efeito da aplicação da regra. O morfismo αL especifica quais elementos devem ser deletados (elementos que estão em L mas não estão em K) e o morfismo αR define quais elementos serão criados (elementos que estão em R mas não estão em K). Além disso, as regras podem ter equações que especificam as condições para que essa possa ser aplicada. A semântica de uma GGA é dada pela aplicação de uma regra sobre um grafo estado do sistema. Uma regra está habilitada e pode ser aplicada se existir uma ocorrência m do LHS no grafo estado, a qual deve satisfazer todas as suas equações. A deleção, a preservação e a criação dos elementos deve levar em conta os atributos, preservando a álgebra.

Na Figura 1 é dado o exemplo da especificação do jogo Pacman descrito por uma GGA. Na especificação algébrica TPacman é definido o tipo (*sort*) de dado Nat (conjunto dos números Naturais) e a operação que pode ser realizada sobre este tipo é a *add* (soma). No grafo tipo T o vértice representado pelo círculo preto e a aresta conectada a ele definem o tipo dos vértices dos lugares do tabuleiro e a forma como eles estão conectados. A maçã, o fantasma e o Pacman podem estar conectados apenas a vértices de lugar e o vértice ContaMaças possui o atributo CM do tipo Nat. O grafo inicial G_0 define o estado inicial do jogo e possui um tabuleiro com doze lugares conectados entre si, um Pacman, um fantasma e uma maçã. O atributo CM é inicializado com 0 (nenhuma maçã foi consumida). A regra r é formada pelos grafos L , K e R e modela a ação de comer a maçã realizada pelo Pacman. Note que r está habilitada, já que existe uma ocorrência (indicada na Figura 1 por um retângulo tracejado em vermelho) de L em G_0 que satisfaz a equação da regra. Na construção da ocorrência o atributo CM de L cujo valor é a variável *cont* será casado com o atributo CM de G_0 cujo valor é 0, atribuindo o valor 0 à variável *cont*. O efeito da aplicação desta regra em G_0 será: (i) a aresta que conecta a maçã ao seu lugar e o atributo CM serão deletados; (ii) uma nova aresta será criada conectando a maçã em outro lugar e o atributo CM será recriado com o valor da variável *cont1* definido pela equação $cont1 = add(cont, 1)$.

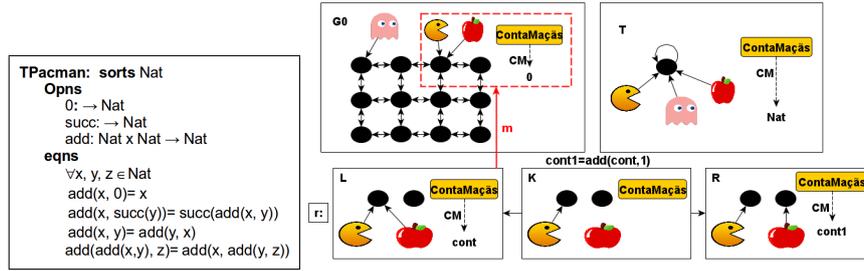


Figura 1. Exemplo reduzido da especificação do jogo Pacman

4. Redes de Petri Fuzzy Generalizadas

A RPF é uma extensão da RPF que vincula triplas de operadores fuzzy às transições visando uma flexibilidade na escolha das operações fuzzy usadas. Definições e conceitos referentes às RPFs podem ser encontrados em [Suraj 2012].

Uma RPF é formada por um conjunto de lugares, um conjunto de proposições, transições, um conjunto de operadores e pelas marcações. Considere a RPF utilizada como exemplo mostrada na Figura 2. Os círculos são os lugares da rede e vinculados à cada lugar tem-se: uma única proposição fuzzy d_i ; um único rótulo p_i que indica o nome do lugar; e um único token (pontos pretos) cujo valor é definido no intervalo $[0, 1]$ o qual determina o valor verdade da proposição associada. Os retângulos modelam as transições. Cada transição da rede representa uma regra fuzzy cujos tipos foram enumerados na Seção 2. Na Figura 2 é ilustrado o conjunto de regras fuzzy que a RPF do exemplo modela através da sua configuração. Vinculados à cada transição tem-se: os lugares de entrada ($I(t)$ pré-condições); os lugares de saída ($O(t)$ pós-condições); um fator de certeza (μ_i); um limiar (λ_i) usado para (des)habilitar uma transição em uma dada marcação; e uma tripla de operadores fuzzy ($Op_{IN}, Op_{Out1}, Op_{Out2}$), onde, Op_{IN} define a operação de agregação das entradas, Op_{Out1} define a operação de aplicação do fator de certeza e Op_{Out2} define a operação de agregação de cada saída. Os operadores usados no exemplo são: min e $*$ (T-normas mínimo e produto algébrico, respectivamente), o operador S_LK (T-conorma de Lukasiewicz) e o operador ID (identidade, usado como operador de entrada para regras do Tipo 1). As marcações definem os valores dos tokens para cada estado do sistema, lugares que não possuem tokens têm valor igual a 0. A primeira disposição dos tokens na rede é chamada de marcação inicial, neste exemplo, $M_0 = (0.6, 0.4, 0.7, 0.5, 0, 0, 0)$. A semântica da rede dá-se pela dinâmica de suas marcações que evoluem conforme os disparos das transições. Uma transição está habilitada em uma dada marcação M e pode disparar se o valor de saída do operador de entrada é positivo e maior ou igual ao valor limiar, isto é, $Op_{In}(M(p_1), \dots, M(p_i)) \geq \lambda(t) > 0, \forall p_i \in I(t)$. Os disparos atualizam os valores das proposições e somente transições que não compartilham lugares podem disparar paralelamente. Alterações nos demais lugares da rede não são realizadas. Para os disparos das transições são considerados dois modos. O Modo 1 não preserva os tokens das pré-condições, já o Modo 2, preserva os tokens das pré-condições, e a computação da saída para ambos os modos é dada por $M'(p) = Op_{Out2}(Op_{Out1}(Op_{In}(M(p_1), \dots, M(p_i)), \mu(t)), M(p))$, Se $p \in O(t)$. Analisando a semântica na rede do exemplo, inicialmente as transições $t1, t2$ e $t3$ estão habilitadas na marcação M_0 . Pois, para $t1$ tem-se $min(0.6, 0.4) \geq 0.4$, para $t2$ tem-se $ID(0.7) \geq 0.2$ e para $t3$ tem-se $S_LK(0.5, 0.6) \geq 0.2$. Porém as transições compartilham lugares e

somente uma pode disparar. Considerando os dois modos, se t_1 disparar, um novo estado da rede definido pela marcação M' é alcançado. De acordo com o Modo 1, o novo estado alcançado está exemplificado na Figura 2 (a) e a computação de M' é realizada da seguinte forma: $M'(p_1) = 0$, $M'(p_2) = 0$, $M'(p_5) = S_LK(*(\min(0.6, 0.4), 0.4), 0) = 0.16$, $M'(p_6) = S_LK(*(\min(0.6, 0.4), 0.4), 0) = 0.16$. Demais lugares da rede permanecem inalterados. Já de acordo com o Modo 2, o novo estado alcançado está exemplificado na Figura 2 (b) e a computação de M' é realizada da forma quase análoga, a única diferença é que os tokens presentes em p_1 e p_2 não são consumidos durante o disparo.

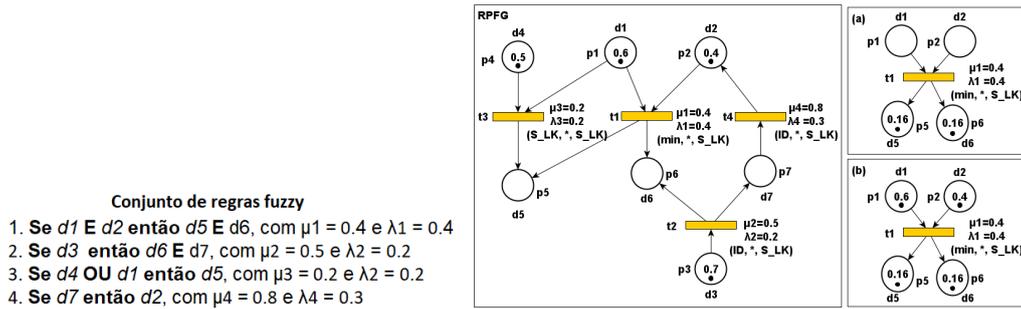


Figura 2. Regras fuzzy; RPF de exemplo; (a) t_1 após o disparo (Modo 1); (b) t_1 após o disparo (Modo 2).

5. Tradução da Rede de Petri Fuzzy Generalizada para Gramática de Grafos com Atributos

Uma GGA é obtida a partir de uma RPF, extraindo-se os elementos da GGA a partir da estrutura da RPF. Tais elementos são: a especificação algébrica, o grafo tipo, o grafo inicial e o conjunto de regras.

A especificação algébrica TFuzzy ilustrada na Figura 3 é uma construção genérica que usa a especificação algébrica NatBool (álgebra dos números naturais e dos booleanos), também ilustrada na Figura 3. Em TFuzzy é definido o tipo de dados $[0, 1]$, o qual é uma discretização (números racionais) do conjunto dos valores fuzzy referentes ao intervalo $[0, 1]$ de reais. As assinaturas das operações fixas estão definidas em **opns** e são: as constantes 0 e 1 (bordas do intervalo); a constante undef (valor indefinido); a operação $./$ (divisão entre dois números naturais que permite realizar a discretização); as operações \leq_R e \neq_R (comparações entre valores); e a operação $+$ (soma). As propriedades algébricas das operações devem ser enumeradas em **eqns** (omitido por questões de espaço). A especificação Tfuzzy para uma tradução é obtida adicionando-se as definições das operações fuzzy da rede. Na Figura 4 é ilustrada a especificação TFuzzy estendida com os operadores da rede de exemplo. Em **opns** foram inseridas as assinaturas dos operadores min , $*$, ID e S_LK . Em **eqns** deve-se adicionar as propriedades algébricas de tais operadores.

A construção do grafo tipo da GGA dá-se por meio da função $Trad_T(N)$ cujo argumento N é a RPF. Todos os grafos da GGA são tipados sobre o grafo tipo construído, onde o morfismo de tipagem é injetor. Isto é, em cada grafo existe apenas uma instância de cada tipo de elemento. Um grafo tipo é obtido a partir de uma RPF da seguinte forma: o conjunto de vértices do grafo é formado pelo conjunto de lugares da rede; o conjunto de

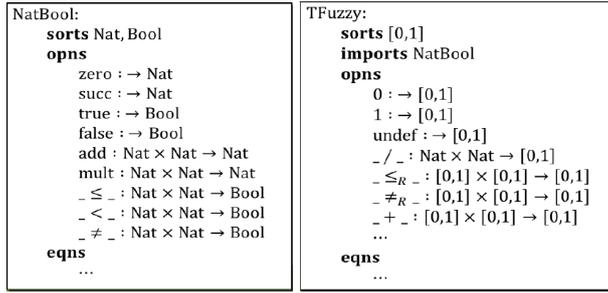


Figura 3. Especificações algébricas: NatBool (naturais e booleanos) e TFuzzy (fuzzy)

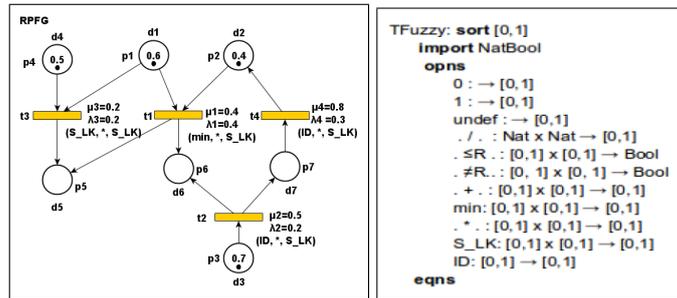


Figura 4. RPF de exemplo e TFuzzy estendida com as operações da rede

arestas é vazio (grafos discretos); os atributos vinculados aos vértices são definidos pelas proposições; vértices e atributos são associados de acordo com o mapeamento bijetivo existente entre lugares e proposições; os atributos estão associados ao seu tipo de dado construído na especificação algébrica TFuzzy ($[0, 1]$). Na Figura 5 é ilustrado o grafo tipo T obtido da rede de exemplo. Pode-se ver que cada lugar da rede transformou-se em um vértice do grafo, pois, cada tipo de token é definido pelo lugar da rede no qual se encontra, já que cada lugar contém um único token. As proposições da rede transformaram-se nos atributos do grafo e cada atributo está vinculado ao tipo $[0, 1]$.

A construção de um grafo estado da GGA dá-se por meio da função $Trad_G(M, N)$ cujos argumentos são a marcação M e a RPF. Um grafo estado é um grafo tipado com atributos obtido a partir de uma marcação da rede como segue: o conjunto de vértices do grafo é formado pelos lugares da rede; o conjunto de arestas é vazio (grafos discretos); os atributos vinculados aos vértices são obtidos através das proposições da rede; vértices e atributos são associados de acordo com o mapeamento bijetivo existente entre lugares e proposições; e os valores associados aos atributos são obtidos a partir da marcação da rede e representam os valores dos tokens. O grafo estado é tipado sobre o grafo tipo, onde cada elemento é mapeado para ele mesmo no grafo tipo. O Grafo inicial é um grafo estado obtido a partir da marcação inicial da rede. Na Figura 5 é ilustrado o grafo inicial G_0 derivado da rede de exemplo. Nele pode-se constatar que os valores dos atributos definem exatamente os valores determinados pela marcação M_0 (primeiro estado) da rede.

Para cada transição da rede é gerada uma regra da gramática. O grafo L e o grafo R são compostos por vértices obtidos das pré-condições ($I(t)$) e das pós-condições ($O(t)$) de cada transição. O grafo de interface K é vazio, pois, todos os elementos serão consumidos e recriados pela aplicação da regra. Nos grafos das regras os valores dos atributos

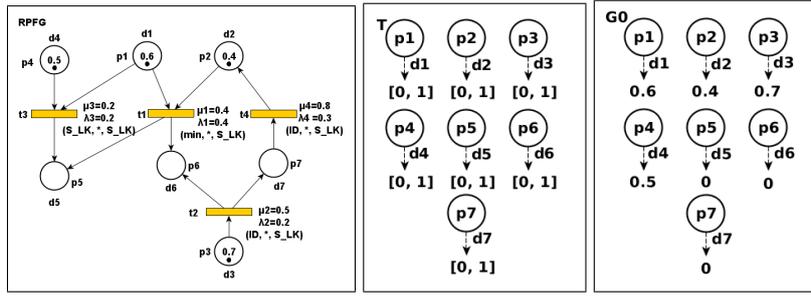


Figura 5. RPPG de exemplo; grafo tipo T; grafo inicial G0

são variáveis, e estas, distinguem os elementos. As variáveis x_i representam os valores obtidos das pré-condições e as y_i representam os valores obtidos das pós-condições. De acordo com o modo de disparo da RPPG obtêm-se dois conjuntos de regras distintos que se diferenciam apenas pela construção do conjunto de equações. Para o conjunto de regras que simula o Modo 1 as equações são construídas como segue: o primeiro tipo de equação é o $\lambda(t_i) \leq Op_{In}(x_{d_1}, \dots, x_{d_n}) = true$ que testa se o limiar da rede é satisfeito; o segundo tipo de equação é o $y'_d = Op_{Out2}(Op_{Out1}(Op_{In}(x_{d_1}, \dots, x_{d_n}, \mu(t)), y_d)$, um para cada lugar de saída da transição, que define os novos valores das proposições derivadas das pós-condições da transição; e o terceiro tipo de equação é o $x'_d = 0$, um para cada lugar de entrada da transição, que zera os valores das proposições derivadas das pré-condições da transição. Para o conjunto de regras que simula o Modo 2 as equações são construídas de forma quase análoga diferenciando-se apenas no terceiro tipo de equação, o qual é o $x'_d = x_d$, que mantêm os valores das proposições derivadas das pré-condições. Na Figura 6 é mostrada a regra $r1$ obtida da tradução da transição t_1 de acordo com o Modo 1. As pré-condições de t_1 (p_1 e p_2) e as pós-condições (p_5 e p_6) se transformaram nos vértices dos grafos $L1$ e $R1$. O grafo K é omitido. Na construção do conjunto de equações da regra, as ocorrências de μ e λ foram substituídas pelos seus respectivos valores definidos em t_1 . Os operadores fuzzy também foram substituídos pelos seus respectivos símbolos. A primeira equação da regra corresponde à condição que habilita a transição t_1 na rede, a segunda e a terceira atualizam os valores dos atributos derivados das pós-condições (y'_5 e y'_6). E as duas últimas equações zeram os valores dos atributos derivados das pré-condições.

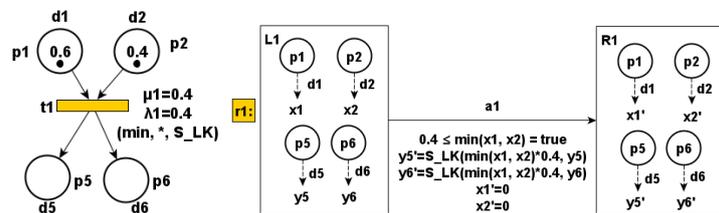


Figura 6. Transição t_1 da RPPG de exemplo e regra r_1 obtida da tradução

A semântica das RPPGs deve ser preservada pela tradução, independente do modo de disparo utilizado. Intuitivamente, considerando um mesmo estado em ambos os modelos, se na RPPG há uma transição habilitada neste estado cujo disparo gera uma nova marcação, então a regra da GGA correspondente à essa transição, também está habilitada no mesmo estado, e sua aplicação gera um grafo estado que corresponde à marcação ob-

tida pelo disparo da transição na RPFPG. A relação inversa também deve ser válida. O Teorema 1 garante que a GGA obtida da tradução proposta preserva a semântica da rede generalizada. A prova deste teorema pode ser encontrada em [Vieira 2021].

Teorema 1 (Preservação da Semântica): Dada uma RPFPG $N = (P, Tr, D, I, O, \mu, \beta, \lambda, Op, \delta, M_0)$ e sua tradução para GGA (considerando o Modo 1 ou 2 de disparo) $GG = (T, G_0, Rules)$, uma transição $t \in Tr$ está habilitada em uma marcação M de N e seu disparo deriva M' (considerando o Modo 1 ou 2 de disparo) se e somente se existe uma ocorrência m para a regra $p_t \in Rules$ em G (um grafo estado da GGA) resultando em um grafo H , de modo que $Trad_G(M, N) = G$ e $Trad_G(M', N) = H$.

6. Conclusões e Trabalhos Futuros

Neste resumo foi apresentada a tradução de uma RPFPG para GGA. Nesta tradução, definiu-se a construção de uma GGA fuzzy cujo conjunto de regras pode ser obtido de acordo com os dois modos de disparo da rede. Além disso, foi apresentada a prova da preservação da semântica de forma sucinta. Durante a prova da preservação da semântica foram identificadas algumas incorreções na tradução, as quais foram corrigidas, o que ressalta a importância deste tipo de prova. Como trabalhos futuros propõem-se construir um novo modelo de AGGs Fuzzy, usando como base a tradução definida neste trabalho e utilizar a ferramenta Rodin, mais especificamente o provador de teoremas, para realizar a verificação de propriedades dos sistemas especificados com este novo modelo. Além disso, pretende-se propor novas especificações algébricas para considerar outros tipos de Lógica Fuzzy.

Referências

- Cavalheiro, S. A. C. (2010). Relational approach of graph grammars. Master's thesis, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- Cavalheiro, S. A. d. C., Foss, L., and Ribeiro, L. (2017). Theorem proving graph grammars with attributes and negative application conditions. *Theoretical Computer Science*, 686:25–77.
- Chen, S.-M., Ke, J.-S., and Chang, J.-F. (1990). Knowledge representation using fuzzy petri nets. *IEEE Transactions on Knowledge and Data Engineering*, 2(3):311–319.
- GMBH, I. (2008). Cookbook pace 2008. Accessed in 10 of january of 2020.
- Jafelice, R. S. d. M. et al. (2005). Notas em matemática aplicada 17. Sociedade Brasileira de Matemática Aplicada e Computacional, São Carlos/SP.
- Reddy, P. V. S. (2017). Fuzzy logic based on belief and disbelief membership functions. *Fuzzy Information and Engineering*, 9(4):405–422.
- Suraj, Z. (2012). Generalised fuzzy petri nets for approximate reasoning in decision support systems. In *CS&P*, pages 370–381. Citeseer.
- Suraj, Z. (2018). Pnes: Tools for the design and analysis petri nets. In *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 391–396.
- Vieira, J. K. (2021). Uma tradução de redes de petri fuzzy generalizadas para gramáticas de grafos com atributos. Dissertação (mestrado em ciência da computação), Universidade Federal de Pelotas.