Petri Nets and Matrix Graph Grammars: Reachability

Pedro Pablo Pérez Velasco, Juan de Lara

Escuela Politécnica Superior (Universidad Autónoma de Madrid)
pedro.perez@uam.es, jdelara@uam.es

Abstract. This paper contributes in two directions. First, several concepts of our matrix approach to graph transformation [1,2] such as coherence and minimal initial digraph are applied to Petri nets, especially to reachability criteria. Second, the state equation and related algebraic Petri net techniques for reachability are generalized using tensor algebra to cover a wider class of rewriting systems.

1 Applying Matrix Graph Grammar Techniques to Petri Nets

Our approach to graph transformation works with simple digraphs, which can be represented as a boolean matrix for edges, and a boolean vector for nodes. A production can be represented by two boolean matrices and two vectors $p = \left(L^E, R^E; L^N, R^N\right)$, where E stands for edges and N for nodes. The actions that can be performed by a production are deletion (e) and addition (r), having two associated matrices each one $\left(e^E, r^E; e^N, r^N\right)$. The output of a production p is defined by $R^E = r^E \vee \overline{e^E}\, L^E$ and similar for nodes. We call *compatibility* to the conditions that must be satisfied by the production in order for its result to be a simple digraph (e.g. avoid dangling edges).

Production concatenation $s_n = p_n; \ldots; p_1$ (p_1 applies first and p_n last) is said to be *coherent* if actions carried out by one production do not prevent the application of those coming afterwards, that is $\bigvee_{i=1}^{n} \left(R_i \bigtriangledown_{i+1}^{n} \left(\overline{e_x}\, r_y\right) \vee L_i \bigtriangleup_1^{i-1} \left(e_y\, \overline{r_x}\right)\right) = 0$ with $\bigtriangleup_{t_0}^{t_1} F(x,y) = \bigvee_{y=t_0}^{t_1} \left(\bigwedge_{x=y}^{t_1} F(x,y)\right)$ and $\bigtriangledown_{t_0}^{t_1} G(x,y) = \bigvee_{y=t_0}^{t_1} \left(\bigwedge_{x=t_0}^{y} G(x,y)\right)$.

The *minimal initial digraph* (MID) is the smallest graph that a coherent sequence oughts to start with in order to fulfill all operations specified by its productions. Given $s_n = p_n; \ldots; p_1$, its MID is defined by $M_n = \bigtriangledown_1^n \left(\overline{r_x} L_y\right)$ The image of concatenation s_n with MID M_n is given by $s_n\left(M_n\right) = \bigwedge_{i=1}^{n} \left(\overline{e_i} M_n\right) \vee \bigtriangleup_1^n \left(\overline{e_x}\, r_y\right)$.

A Petri net can be represented as a graph (where only tokens need to be depicted), together with productions for each of its transitions. Thus, it is possible to see Petri nets as a proper subset of graph grammars, where no dangling edges occur when applying productions, and each rule can only be applied in one part of the graph. Therefore, given a grammar $\mathfrak{G} = \left(M_0, \{p_1, \ldots, p_n\}\right)$, a state M_d is *reachable* from M_0, if there exists a coherent concatenation with MID contained in M_0 and image in M_d.

2 Extending the Petri Net State Equation for Graph Grammars

First, we consider *DPO-like matrix graph grammars*, in which rule applications do not generate dangling edges. In order to perform an *a priori* analysis it is mandatory to get rid of matches. To this end, node and edge types are considered instead of node and edges themselves, obtaining matrices with elements in $\mathbb{Z}$.

We use tensors of the form ${}^{E}_{0}\mathrm{A}^{i}_{j}$, with top left index indicating nodes (N) or edges (E). Bottom left index specifies the position inside a sequence, if any. Top right and bottom right are contravariant and covariant indexes, respectively. Given a grammar $\mathfrak{G} = ({}_{0}M, \{p_1, \ldots, p_n\})$, the incidence matrix for nodes ${}^{N}\mathrm{A} = \left(\mathrm{A}^{i}_{k}\right)$ ($i \in \{1, \ldots, n\}$, $k \in \{1, \ldots, m\}$, with m the number of different types of nodes in $\mathfrak{G}$) is defined by $+/-r$ if production k adds (resp. deletes) r nodes of type i. The state equation for nodes is thus given as: ${}^{N}_{d}\mathrm{M}^{i} = {}^{N}_{0}\mathrm{M}^{i} + \sum_{k=1}^{n} {}^{N}\mathrm{A}^{i}_{k}\mathrm{x}^{k}$.

The case for edges is similar, but they are represented by matrices instead of vectors and thus the incidence matrix becomes the *incidence tensor* ${}^{E}\mathrm{A}^{i}_{jk}$. Initial nodes will be assumed to have a contravariant behaviour (index on top, i) while terminal nodes (first index, j) and productions (second index, k) will behave covariantly (index on bottom).

We represent contraction w.r.t. second covariant index (productions) by $\mathfrak{C}_{(0,2)}$ and $\mathrm{M} = {}_{d}\mathrm{M} - {}_{0}\mathrm{M}$. Then, a necessary condition for state ${}_{d}\mathrm{M}$ to be reachable from state ${}_{0}\mathrm{M}$ is ${}^{E}\mathrm{M} = {}^{E}\mathrm{M}^{i}_{j} = \sum_{k=1}^{n} {}^{E}\mathrm{A}^{i}_{jk}x^{k} = \mathfrak{C}_{(0,2)}\left({}^{E}\mathrm{A} \otimes x\right)$, where $i, j \in \{1, \ldots, m\}$. It is possible to derive a unique equation considering both nodes and edges, by extending the incidence matrix M as the right of Fig. 1 shows.

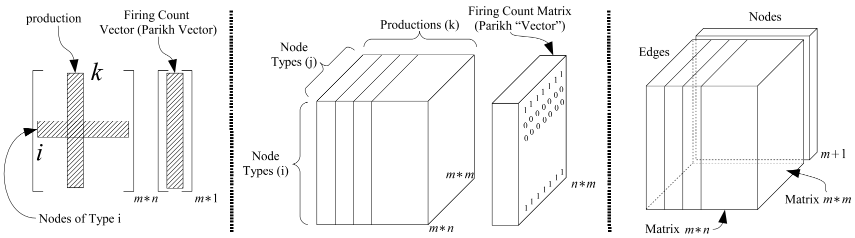

Fig. 1. Matrix Representation for Nodes, Tensor for Edges and Their Coupling.

We also consider *SPO-like grammars*, where a production can be applied even if dangling edges appear (and are deleted). We distinguish two cases, grammars with external and internal $\varepsilon-$productions. In the former, rules act only on edges that appear in the initial state, so they can be advanced to the beginning of the sequence. Internal $\varepsilon-$productions delete edges added or used by productions preceding them. Altogether, the necessary condition for state ${}_{d}\mathrm{M}$ to be reachable from ${}_{0}\mathrm{M}$ is $M^{i}_{j} = \sum_{k=1}^{n}\left(A^{i}_{jk} + V\right)x^{k} + b^{i}_{j}$, with V being the modification of the effects of productions (i.e. deletion of extra dangling edges) due to internal $\varepsilon-$productions, and b^{i}_{j} an inequality constraint produced by external $\varepsilon-$productions.

References

1. Pérez Velasco, P. P., de Lara, J. 2006. *Towards a New Algebraic Approach to Graph Transformation: Long Version.* Tech. Rep. of the School of Comp. Sci., Univ. Autónoma Madrid. Available at http://www.ii.uam.es/∼jlara/investigacion/techrep_03_06.pdf.
2. Pérez Velasco, P. P., de Lara, J. 2006. *Matrix Approach to Graph Transformation: Matchings and Sequences.* To appear in Proc. ICGT'06, LNCS 4178, pp.: 122-137, Springer.