Data Structures and Algorithms for Extended State Space and Structural Level Reduction of the GSPN Model

Donna S. Nielsen and Leonard Kleinrock
University of California, Los Angeles

Abstract: This paper extends the applicability of exact analysis of the GSPN model by providing methods to improve the time and space complexity of both state space and structural level reduction. For state space level reduction, we maximize the concurrent firing of immediate transitions. For structural level reduction, we minimize the number of generated replicas for timed transitions by using branch and bound techniques to create concurrent replicas that simulate the firing of the timed transition followed by the simultaneously firing of multiple immediate transitions.

1 Introduction

State space level reduction techniques eliminate vanishing markings by concurrently firing multiple immediate transitions in a single state space transition [1, 4]. Structural level reduction techniques eliminate immediate transitions, the source of vanishing markings, by iteratively creating replicas of a timed transition to simulate the firing of the timed transition followed by the firing of an immediate transition [2, 6].

This paper provides data structures and algorithms to extend the applicability and to efficiently implement both state space and structural level reduction of the GSPN model. With respect to state state space level reduction, we rely on knowledge of the given marking, as well as the GSPN structure, to maximize the concurrent firing of immediate transitions. In addition, we develop efficient algorithms to generate the concurrent transition firing combinations and their corresponding firing probabilities. With respect to structural level reduction, we avoid the generation of redundant replicas by applying state space level reduction techniques at the structural level to generate concurrent replicas which simulate the firing of a timed transition followed by the concurrent firing of multiple immediate transitions. In addition, we use branch and bound techniques to both

This work was sponsored by the Advanced Research Projects Agency of the U.S. Department of Defense under Contract MDA 972-91-J-1011, Advanced Networking and Distributed Systems.

avoid the generation of infeasible replicas and permit efficient determination of the feasibility of generated replicas.

The remainder of the paper is organized as follows: Section 2 defines the data structures to represent the GSPN model. Section 3 and Section 4 develop the algorithms for state space and structural level reduction, respectively. Finally, Section 5 provides concluding remarks.

2 Data Structures for the GSPN Model

We assume the reader is familiar with the structural level definition and basic properties of the GSPN model. The parameters in the formal eight-tuple are standard GSPN notation as defined in [3].

$$GSPN \triangleq (P, T, \Pi(\cdot), W^{-}(\cdot), W^{+}(\cdot), W^{H}(\cdot), M_0, \Lambda(\cdot)).$$

We restrict the GSPN model to include only two priority levels such that immediate transitions have priority over timed transitions. The notation $W^{-}(p,t)$, $W^{+}(p,t)$, and $W^{H}(p,t)$ denotes the multiplicity of place p in the input, output, and inhibitor functions of transition t. When appropriate, we employ a vector representation for a bag such that each vector component equals the corresponding bag multiplicity. Boldface type distinguishes the vector representation of a bag. If no inhibitor arc exists from place p to transition t, then the multiplicity of place p in the inhibitor bag of t is infinity. An inhibitor arc with weight infinity is equivalent to the absence of an inhibitor arc. We adopt the infinite arc weight representation to facilitate the use of bag operations.

In this section, we provide data structures to represent the structural level specification of the GSPN model. These data structures support the implementation of both state space and structural level reduction. Either a directed graph or a bag represents each component of the formal GSPN tuple.

2.1 Representation of Directed Graphs and Bags

We represent a directed graph with an adjacency list for each node in the graph. Let $G = (V, E)$ be a directed graph with the set of nodes V and set of arcs E. A node w is in the adjacency list of a node v if and only if there exists a directed arc from v to w.

We represent a bag B of a set S with a variable length ordered K-tuple $((n_k, x_k) : 1 \le k \le K \le |S|)$. The elements in S are assigned an arbitrary numerical ordering and each variable x_k indexes a set element. Each variable n_k specifies the multiplicity in bag B of the set element indexed by x_k. The ordering of the list requires the index x_k to be less than the index x_{k+1}. We use the ordered tuple to represent the input, output, and inhibitor functions of a GSPN. For an inhibitor function, the absence of a place in the tuple represents a multiplicity of infinity for that place in the corresponding bag.

The power set of a bag is the set of all subbags. Both state space and structural level algorithms defined in Section 3 and Section 4 require the determination of which subbags in a given power set satisfy application specific feasibility constraints.

2.2 Tree Representation for a Power Set

A tree structure representation of the power set facilitates the use of branch and bound techniques to avoid enumeration and examination of infeasible subbags in a given power set. Each node of the tree represents a subbag in the power set. The root node of the tree is at level zero and represents the empty set. The path from the root node to any given node defines the subbag corresponding to that node. Specifically, the arc label from level k to level $k+1$ on the path specifies the $(k+1)st$ ordered pair in the tuple that represents the subbag corresponding to the given node.

To perform a BFS generation of a power set tree for a bag B, a branching function defines the outgoing arcs of a generated node. Specifically, let the tuple $\{(n_1, x_1)(n_2, x_2)\ldots, (n_k, x_k)\}$ represent the subbag associated with a generated node u. The branching function creates a node v and a directed arc with label (n_{k+1}, x_{k+1}) from node u to v if and only if $x_k < x_{k+1} \leq |S|$ and the multiplicity n_{k+1} is less than or equal to the multiplicity in the bag B of the set element indexed by x_{k+1}. This branching function both ensures each node in the tree represents a subbag in the power set and avoids the generation of multiple instances of the same subbag. For each generated node, the evaluation of application specific feasibility constraints determines the feasibility of the subbag. Likewise, the evaluation of an application specific bounding function determines if all subbags in the subtree rooted at a generated node are infeasible, thereby avoiding the generation of identified infeasible subtrees. A good bounding function should prune a substantial number of nodes in the power set tree, while maintaining an efficient evaluation at each generated node.

2.3 Graphical Representation of Immediate Transitions

We represent the set of immediate transitions by a directed graph which portrays a partial order among transitions with respect to their enabling conditions. We assume the trivial restriction that the GSPN contains no source immediate transitions.

2.3.1 Structural Enabling Cover Relation

The structural enabling cover relation provides sufficient structural conditions for the enabling of one transition to dictate the enabling of another transition. Transition t_i is a structural enabling cover (SEC) for transition t_j if and only if the input function of t_j is a subbag of the input function of t_i, and the inhibitor function of t_i is a subbag of the inhibitor function of t_j. For example, given the GSPN in Figure 1, t_2 is a structural enabling cover of t_1. The SEC relation is a transitive, antisymmetric, reflexive relation that provides a partial ordering on a set of immediate transitions. Formally,

$$t_i \; SEC \; t_j \text{ iff } W^-(t_j) \subseteq W^-(t_i) \cap W^H(t_i) \subseteq W^H(t_j). \tag{1}$$

2.3.2 Enabling Graph

The enabling graph (EG) is a directed acyclic graph that depicts the partial order among the set of immediate transitions with respect to the structural enabling cover relation. Each node v of an EG contains a set of transitions T_v such that there exists a path of zero length or more from the node containing transition t_i to the node containing transition t_j if and only if t_i is a structural enabling cover for t_j. An EG is minimal if there does not exist a subgraph of the EG which also reflects the SEC partial order. Figure 1 shows a GSPN immediate subnet and its corresponding enabling graph.

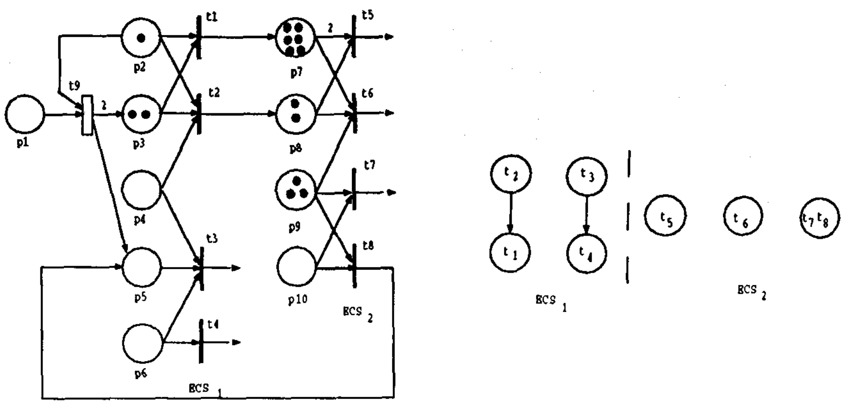


Figure 1: GSPN Subnet and Enabling Graph

2.4 Extended Conflict Sets

The partitioning of the set of immediate transitions into extended conflict sets (ECSs) effectively partitions the connected components of the enabling graph. In other words, transitions in the same connected components of the EG must be in the same ECS. Specifically, the SEC partial order depicted in the EG dictates that the transitions contained in any two nodes connected by a directed path are in symmetric structural conflict. And, the transitive and reflexive closure of the symmetric structural conflict relation dictates that the transitions contained in any two nodes connected by an undirected path are in the same ECS. Figure 1 shows the partitioning of the connected components of the EG into ECSs.

3 State Space Level Reduction Methods

The concurrent firing of immediate transitions in a single state space transition eliminates the intermediate vanishing markings that derive from all possible firing order permutations. In this section, we extend the reduction methods developed in [1] to permit the concurrent firing of transitions both in different ECSs and within an ECS of a structurally confused immediate subnet.

3.1 Enabling and Firing Rules for Concurrent Firings

The enabling bag $EB(M)$ is a bag of transitions that specifies not only which transitions are enabled in marking M, but also the number of enablings $EB(t, M)$ of each transition t. Formally,

$$EB(t,M) = \begin{cases} 0 & t \notin E(M) \\ \max\{\, n \in \mathbb{N} \mid nW^-(t) \subseteq M \,\} & \text{otherwise} \end{cases} . \tag{2}$$

A transition firing combination X is a bag of transitions such that the multiplicity of a transition in the bag specifies the number of times the transition fires. Formally, the concurrent firing of a bag of transitions X in marking M results in the new marking M':

$$M' = M - \sum_{t \in T} X(t)W^-(t) + \sum_{t \in T} X(t)W^+(t). \tag{3}$$

3.2 Marking Dependent Properties

The disabled status of certain transitions in a given vanishing marking can eliminate both confusion and conflict present in the GSPN structure. In this section, we define marking dependent counterparts of structural properties to identify the absence of confusion and conflict among transitions, thereby permitting the maximum concurrent firing of immediate transitions.

3.2.1 Marking Confusion-Free Property

In a given vanishing marking, an immediate subnet possibly containing structural confusion is marking confusion-free if for any enabled transition t_k and any disabled transition t_j in the same ECS, the causally connected set $CCS_{t_k}(t_j)$ contains no enabled transitions. In other words, the firing of an enabled transition in one ECS cannot enable a disabled transition in another ECS. For example, in Figure 1, the immediate subnet containing ECS_1 and ECS_2 is marking, but not structurally, confusion-free. Formally, for marking M, an immediate subnet is marking confusion-free if:

$$\forall t_k \in E(M) \;\; \forall t_j \in ECS(t_k) \cap \overline{E(M)}, \; CCS_{t_k}(t_j) \cap E(M) = \emptyset. \tag{4}$$

3.2.2 Marking Free-Choice Property

In a given vanishing marking, a non free-choice ECS exhibits the free-choice property for a specified number of transition firings if the set of enabled transitions in the ECS remains unchanged for these transition firings. Let n be the minimum number of enablings over all the enabled transitions in a given ECS. If the firing of any enabled transition in the ECS does not increment the inhibitor place of any other enabled transition in the ECS, then all the enabled transitions in the ECS remain enabled for n transition firings within the ECS. In addition, if the immediate subnet of the ECS is marking confusion-free then all disabled transitions in the given ECS remain disabled for these n transition firings. For example, ECS_2 in Figure 1 is not free-choice; however, ECS_2 is marking free-choice for the first two transition firings. Formally, for a a given marking M, an ECS in a marking confusion-free subnet is marking free-choice(n) if for all $t_i, t_j \in ECS \cap EB(M)$,

$$n = \min_{t_i}\{EB(t_i, M)\} \cap \left(W^+(t_j) - W^-(t_j)\right) \cap W^H(t_i) = \emptyset. \tag{5}$$

3.3 Concurrent Transition Firing Combinations

In this section, we define methods to identify which transition combinations can fire concurrently, along with their associated firing probabilities.

In a marking free-choice(n) ECS, the set of enabled transitions remains unchanged throughout the firing of any n transitions within the ECS. Thus, the firing probability of each of the n transitions is independent of its order in the firing sequence. This independence permits the concurrent firing of the first n transitions to fire within the ECS. Specifically, the set of concurrent transition firing combinations within a free-choice(n) ECS is the set of all unique transition combinations when selecting n transitions to fire, with replacement and without regard to order, from the set of enabled transitions in the given ECS.

The general branch and bound generation of a power set tree as defined in Section 2 provides an efficient method to generate all concurrent transition firing combinations within a marking free-choice(n) ECS. Without loss of generality let t_1 through t_m be the set of enabled transitions in the ECS. Let T_n denote the bag of that contains a multiplicity of n for each transitions t_1 through t_m. The feasibility constraint dictates that a subbag of T_n is a concurrent transition firing combination if and only if the cardinality of the subbag is n. Likewise, the bounding function avoids the generation of all nodes that represents subbags with a cardinality that exceeds n. In other words the leaf nodes of the generated portion of the power set tree represent the concurrent transition firing combinations. To achieve a reduction in space requirements, the algorithm discards any infeasible node after the generation of all the node's outgoing arcs.

The firing probability associated with any concurrent transition firing combination X must equal the probability of firing the transitions in X one at a time in any different firing order. Specifically, the firing probability of X is simply the product of the firing probabilities for each transition in X times the number of permutations, defined by the multinomial coefficient, to account for all possible firing orders. Formally,

$$Prob\{X \text{ fires }\} = \frac{n!}{\prod_{t_i \in X} X(t_i)!} \prod_{t_i \in X} \left[\frac{\Lambda(t_i)}{\sum_{t_j \in E(M) \cap ECS(t_i)} \Lambda(t_j)} \right]^{X(t_i)} \tag{6}$$

By direct extension of the results in [3], the underlying Markovian process of the GSPN is independent of the firing order of transitions in different ECSs of a marking confusion-free immediate subnet. This independence permits the concurrent firing of transitions in the different ECSs. Thus, the Cartesian product of the sets of concurrent transition firing combinations over each ECS defines the set of concurrent transition firing combinations for the immediate subnet. Likewise, the firing probability associated with any concurrent transition firing combination for the immediate subnet is simply the product over the firing probabilities of the corresponding concurrent transition combinations for each ECS within the subnet.

3.4 Time and Space Complexity Analysis

In [1], they discuss the reduction in vanishing markings achieved by the concurrent firing of transitions in different ECSs. In this section, we provide a theoretical analysis of the reduction in the number of vanishing markings generated through the concurrent, rather than sequential, firing of transitions within an ECS. For a given marking free-choice(n) ECS with m distinct enabled transitions, the number of resulting markings from firing all feasible concurrent firing combinations is simply the number of ways to distribute n non-distinct objects into m distinct cells. The sequential firing of transitions generates the intermediate vanishing markings to account for all possible transitions firing combinations after the firing of each transition within the sequence. Formally, the reduction in the number of vanishing markings generated by the concurrent, rather than sequential firing of n transitions is:

$$\sum_{i=1}^{n-1} \binom{m+i-1}{m-1}. \tag{7}$$

4 Structural Level Reduction Methods

In this section, we modify the structural level reduction algorithms proposed in [2, 5] to avoid the generation of redundant replicas that simulate different firing order permutations of the same transition firing combination. To replicate a timed transition, the proposed structural reduction algorithm first computes all feasible bags of enabled immediate transitions directly after the firing of the given timed transition. Given a feasible bag of enabled immediate transitions, the state space reduction methods defined in Section 3 provide all the feasible concurrent transition firing combinations and their corresponding firing probabilities. The direct applicability of state space level reduction techniques at the structural level is due to the dependency of the state space reduction techniques on only the bag of enabled transitions, rather than the actual markings. Each generated replica corresponds to both a feasible bag of enabled transitions and a concurrent transition firing combination within the bag. The computed enabling conditions for the feasible bag of enabled transitions dictate the input and inhibitor functions of a replica, while the input and output functions of the transitions in the concurrent transition firing combination dictate the replica's output function. Iterative replications of a timed transition result in replicas that simulate the firing of the timed transition followed by each possible firing combination of immediate transitions. Within this section we provide an outline of the modified structural reduction algorithm and a corresponding example showing the reduction of transition t_9 in Figure 1. For simplification, the example does not include inhibitor arcs; however, all formulas fully account for the effect of any inhibitor arcs.

4.1 Maximum Bag of Enabled Transitions

A timed transition's maximum enabled bag (MEB) is a bag of immediate transitions such that the multiplicity of each transition in the MEB corresponds to its maximum possible number of enablings directly after the firing of the timed transition. Since all feasible bags of enabled transitions must be a subbag of the MEB, the cardinality of the MEB's power set specifies the potential number of replicas created during a single replication step. To minimize this exponential complexity, we extend the GSPN structural properties defined in [3] to establish stringent sufficient structural conditions which restrict the maximum number of direct enablings. These structural properties must also be inclusive enough to determine when the GSPN structure prohibits any enablings of immediate transitions, thereby signifying the completion of the iterative replication of the timed transition.

4.1.1 Structural Enable Relation

The structural enable ($SE(n)$) relation provides necessary structural conditions for the firing of a transition to achieve n direct enablings of another disabled transition. For the firing of transition t_i to enable transition t_j, the firing of t_i must either increment an input place or decrement an inhibitor place of t_j. If the incremented input place of t_j is also an input place to t_i then the arc weight from the input place to t_i must be less than the arc weight from the input place to t_j. This condition allows the incremented input place to contain sufficient tokens to enable t_i while concurrently disabling t_j. Analogous conditions apply to the decremented inhibitor place. If the firing of a timed transition can enable an immediate transition by decrementing an inhibitor place, then the maximum number of enablings of the immediate transition depends on the initial marking and is indeterminate with respect to the GSPN structure. Otherwise, the increased token count in the incremented input places of the immediate transition dictate an upper bound on the maximum number of enablings. For example, in Figure 1, the firing of t_9 structurally enables both t_1 and t_2 two times and t_3 one time. Formally, $t_i\ SE(n)\ t_j$ iff $\exists p \in P$,

$$\begin{aligned}&((W^+(p,t_i) - W^-(p,t_i) < 0) \cap (W^H(p,t_i) > W^H(p,t_j))) \cup \\ &\left((W^-(p,t_i) < W^-(p,t_j)) \cap \left\lceil \frac{W^+(p,t_i) - W^-(p,t_i)}{W^-(p,t_j)} \right\rceil \geq n\right).\end{aligned} \tag{8}$$

4.1.2 Structural Disable Relation

The structural disable ($SD(n)$) relation provides sufficient structural conditions for the firing of a transition to prohibit n direct enablings of another transition. Specifically, if the firing of transition t_i increments an inhibitor place of transition t_j by a token count which exceeds the weight of the inhibitor arc, then the structure prohibits the firing of t_i to directly enable t_j. Otherwise, if there exists a place which is an inhibitor place to t_i and an input place to t_j, then the enabling and subsequent firing of the t_i dictates an upper bound on the maximum number of direct enablings of t_j. This particular circumstance is somewhat unusual for a non-structurally reduced GSPN; however, during the iterative replication process this restriction on the enabling conditions of a replica specifies which markings map into which feasible bags of enabled transitions. Formally, $t_i\ SD(n)\ t_j$ iff $\exists p \in P$,

$$W^+(p,t_i) \geq W^H(p,t_j) \cup \left\lceil \frac{W^H(p,t_i) + W^+(p,t_i) - W^-(p,t_i)}{W^-(p,t_j)} \right\rceil < n. \tag{9}$$

Note that the structural disable relation provides sufficient conditions for the firing of a transition to guarantee the disabling of another transition; whereas, the structural conflict relation provides necessary conditions for the firing of a transition to have the potential to disable another transition.

4.1.3 Structurally Maximum Bag of Enabled Transitions

The structural enable and disable relations provide sufficient structural conditions to restrict the maximum number of direct enablings of an immediate transition upon firing a given timed transition. In addition, the structural enabling cover relation dictates that if $t_j\ SEC\ t_k$ then the maximum number of enablings of transition t_j cannot exceed the maximum number of enablings of t_k. For example, in Figure 1, since t_3 is a structural enabling cover for t_4 and the firing of t_9 permits no enabling of t_4, then, even though the firing of t_9 increments an input place of t_3, t_3 also must have no enablings. Formally, the multiplicity of immediate transition t_j in the maximum enabling bag of t_i is:

$$\max\left\{n \mid t_i\ SE(n)\ t_j \cap t_i\ \overline{SD(n)}\ t_j \cap\ MEB(t_k) \geq n\ \forall t_k : t_j\ SEC\ t_k\right\}. \tag{10}$$

The MEB for timed transition t_9 in Figure 1 is $\{2t_1, 2t_2\}$. In [5], we provide an algorithm that generates the set of immediate transitions that can be directly enabled after the firing of a timed transition. This algorithm only requires minor modification, with no additional cost in complexity, to also determine the maximum number of enablings of each transition.

4.1.4 Indeterminate Transitions in an MEB

To simplify the algorithm for a transition replication step, we assume the timed transition's MEB contains no indeterminate transitions. A transition in the MEB is indeterminate with respect to the GSPN structure if the maximum number of direct or indirect enablings, after the firing of a timed transition and prior to reaching a tangible marking, is dependent on the initial marking. The structural enable relation establishes necessary conditions for a transition in the MEB to be indeterminate. Specifically, this indeterminacy occurs either if an incremented input place in the bag evaluated in the first inequality of Equation 8 participates in a cycle of immediate transitions or if, as stated previously, there is a decremented inhibitor place in the bag evaluated in the second inequality of Equation 8. In [5] we provide methods to perform structural reduction with indeterminate transitions.

4.1.5 Enabling Graph Representation of the MEB

An enabling graph representation of a given MEB facilitates the generation of all structurally feasible bags of enabled transitions within the MEB. This enabling graph depicts the *SEC* partial order, not only among the transitions in the MEB, but also among each distinct number of enablings from one to a transition's multiplicity in the MEB. Specifically, each node v of the EG contains a set of transitions T_v such that there exists a path from the node containing n_i enablings of transition t_i to the node containing n_j enablings of transition t_j if and only if n_i enablings of transition t_i is a structural enabling cover for n_j enablings of transition t_j. Figure 2a shows the augmented EG for $MEB(t_9)$ corresponding the GSPN in Figure 1. All future references to an enabling graph refer to the enabling graph which represents the MEB of the replicated timed transition.

We represent a subset of nodes in the EG of the MEB with the standard variable length ordered K-tuple $((x_k) : 1 \leq k \leq K \leq |V_{EG}|)$ such that each x_k indexes a node in the EG. In turn, we represent a subbag of the MEB with a subset of the nodes in the EG such that the subbag of the MEB is simply the union of the bag of transitions contained in the subset of nodes.

4.2 Feasible Bags of Enabled Transitions

The GSPN structure restricts which subbags in the MEB can be simultaneously enabled, without forcing the enabling of any remaining transitions in the MEB. In this section, we develop methods to generate these structurally feasible bags of enabled transitions. We rely on branch and bound techniques to avoid the enumeration and examination of the entire power set of the MEB.

4.2.1 Extended Input and Inhibitor Functions

The extended input and inhibitor functions $W^-(\cdot)$ and $W^H(\cdot)$ map bags of transitions into bags of places and define necessary marking conditions to achieve the specified number of enablings of each transition in the bag. Formally, for a non-empty bag of transitions T_B:

$$W^-(T_B) = \bigcup_{t \in T_B} T_B(t) W^-(t); \quad W^H(T_B) = \bigcap_{t \in T_B} W^H(t). \tag{11}$$

A bag of transitions T_B is enabled in marking M if and only if $\mathbf{W}^-(T_B) \leq \mathbf{M} < \mathbf{W}^H(T_B)$.

4.2.2 Enabling Tree

The general branch and bound generation of a power set tree as defined in Section 2 provides an efficient method to generate all structurally feasible bags of enabled transition for a given MEB. Since a subset of nodes in the EG represents a subbag of the MEB, the generated enabling tree essentially represents the power set, with pruning, of the set of nodes in the EG. Let EB_v be the subbag of the MEB corresponding to node v of the enabling tree.

The GSPN structure imposes two feasibility constraints on any feasible bag of enabled immediate transitions. The first structural constraint ensures the enablings of the transitions in a feasible bag do not force the enabling of any remaining transitions in the MEB. Specifically, if an enabled bag EB has n_i enablings of transition t_i and these enablings are a structural enabling cover for n_j enablings of transition t_j then EB must also contain at least n_j enablings of t_j. For example, in Figure 2a, a feasible enabling bag which contains two enablings of transition t_2 must also contain two enablings of transition t_1.

The second structural feasibility constraint prohibits a feasible enabled bag from containing structurally mutual exclusive (SME) transitions. To simplify the algorithm for a transition replication step, we assume the given MEB contains no structurally mutual exclusive transitions. In [5], we provide structural reduction algorithms which incorporate this SME feasibility constraint.

The bounding function requires a topological ordering of the nodes in the EG such that if there exists a path from node x to node y in the EG then the index of node x is greater than the index of node y. As shown for the EG in Figure 2a, the numbering of the nodes in increasing order as they are post-visited in a DFS traversal achieves this topological ordering. The bounding function prunes any subtree of the enabling tree that is rooted at an infeasible node. Specifically, if EB_v does not satisfy the feasibility constraint then there must exist a transition t_i with multiplicity n_i in EB_v and a transition t_j with multiplicity n_j in $\overline{EB_v}$ such that $n_i t_i \; SEC \; n_j t_j$. For any node w in the subtree rooted at node v, since EB_v is a subbag of EB_w then EB_w also contains n_i enablings of transition t_i.

In addition, due to the topological ordering of the nodes in the EG, EB_w also cannot contain n_j enablings of transition t_j. Thus EB_w is also infeasible.

Let $(x_1, x_2, \ldots x_k)$ represent the set of EG nodes associated with node u of the enabling tree. The following steps define the BFS generation of all direct descendents of node u.

Branching Function: For all x_{k+1} such that $x_k < x_{k+1} \leq |V_{EG}|$, generate node v and create an arc from u to v with label x_{k+1}.

Node attributes:

$EB_v = EB_u \cup T_{x_{k+1}}$.

$W^-(EB_v) = W^-(EB_u) \cup W^-(T_{x_{k+1}})$.

$W^H(EB_v) = W^H(EB_u) \cap W^H(T_{x_{k+1}})$.

Feasibility Constraints: EB_v not feasible if $\forall t_i, t_j \in EB_v$, $EB_v(t_i) = n_i \Rightarrow EB_v(t_j) \geq n_j$ if $n_i t_j$ SEC $n_j t_j$.

Bounding Function: Prune subtree rooted at v, if EB_v is infeasible.

The potential for complexity reduction in this branch and bound technique is two-fold: the bounding eliminates the generation of some subbags in the power set of the MEB and the systematic tree generation permits a worst case complexity of $O(|P| + |MEB|)$ to determine the feasibility for a generated subbag. Figure 2b shows the enabling tree associated with $MEB(t_9)$. The numbering of the nodes in the enabling tree is in accordance with the order of the BFS node generation.

4.2.3 Enabling Function

Given the enabling tree, the replication of the timed transition requires the determination of the enabling conditions corresponding to each structurally feasible bag. The function $Enab(\cdot)$ maps a structurally feasible enabled bag into a set of markings such that marking M is an element of $Enab(T_B)$ if and only if T_B is enabled in marking M. Formally,

$$Enab(T_B) = \{\, M \mid \mathbf{W}^-(T_B) \leq \mathbf{M} < \mathbf{W}^H(T_B) \,\}. \tag{12}$$

Any marking in the enabling function of a given feasible enabled bag ensures the enabling of all transitions in the given bag, but the marking does not prohibit the enabling of any remaining transitions in the MEB. Thus, a single marking may map into more than one enabling function. The complete specification of a feasible bag's enabling conditions requires more stringent marking conditions to ensure the disabling of the remaining transitions in the MEB.

4.3 Feasible Bags of Disabled Transitions

Corresponding to each structurally feasible bag T_B of enabled transitions is a feasible bag of disabled transitions $\overline{T_B}$. In this section, we establish the marking conditions that disable the transitions in a given feasible bag of disabled transitions, without forcing the disabling of any remaining transitions in the MEB.

4.3.1 Disabling Input and Inhibitor Function

The disabling input and inhibitor functions $W_D^-(\cdot)$ and $W_D^H(\cdot)$ map bags of transitions into bags of places and define necessary structural conditions to disable, by restricting the token count of single place, the specified number of enablings of each transition in the bag. The subscript 'D' emphasizes that the resulting bags of places define disabling conditions rather than enabling conditions. Formally, for a non-empty bag of transitions T_B:

$$W_D^-(T_B) = \bigcap_{t \in T_B} T_B(t)W^-(t); \quad W_D^H(T_B) = \bigcup_{t \in T_B} W^H(t). \tag{13}$$

The specified multiplicity of each transition in the bag T_B is disabled in marking M if $\mathbf{W}_D^-(T_B) \not\leq \mathbf{M} \cup \mathbf{M} \not< \mathbf{W}_D^H(T_B)$.

4.3.2 Disabling Tree

In this section, we identify all feasible bags of disabled transitions that satisfy necessary structural conditions to permit the disabling of the transitions in the bag, by restricting the token count in a single place, without forcing the disabling of any remaining transitions in the MEB. We refer to these bags as singularly disabled bags.

The branch and bound generation of a power set tree provides an efficient method to generate all singularly disabled bags of transitions. Analogous to the generation of the enabling tree, the generated disabling tree essentially represents the power set, with pruning, of the set of nodes in the EG. Let the disabled bag DB_v be the subbag of an MEB associated with node v of the disabling tree.

The structural feasibility constraint for a bag of singularly disabled transitions requires the existence of at least one place such that the multiplicity of that place in the disabling function of DB_v exceeds the multiplicity of that place in the enabling function of $\overline{DB_v}$ or the multiplicity of that place in the disabling inhibitor function of DB_v is less than the multiplicity of that place in the inhibitor function of $\overline{DB_v}$. For example, given $MEB(t_9)$ for the GSPN in Figure 1, any bag of transitions that contains t_1 and does not contain t_2 is not a feasible disabled bag since the disabling of t_1 always forces the disabling of t_2. Also, the bag $\{2t_1, t_2\}$ is not a feasible disabled bag since the disabling of these transition enablings forces the disabling of t_1.

To both retain the EG node numbering that was used to generate the enabling tree and apply the bounding function, we alter the branching function for the generation of the disabling tree such that $x_{k+1} < x_k \leq |V_{EG}|$. The bounding function prunes any subtree rooted at any node v if there does not exist a complement node in the enabling tree corresponding to $\overline{DB_v}$. Specifically, using arguments analogous to those described for the enabling tree, if $\overline{DB_v}$ is not a feasible enabled bag then DB_v is not a feasible disabled bag because the disabling of a transition in DB_v forces the disabling of a transition in $\overline{DB_v}$. Due to the topological ordering of the nodes in the EG and the specified branching function, for any node w in the subtree rooted at node v, the bag

DB_w will contain this same infeasibility. Thus, an efficient BFS generation of the disabling tree requires a parallel reverse BFS traversal of the enabling tree to determine the existence of the complement node, as well as efficiently access the complement node attributes to efficiently evaluate the feasibility constraint. A second bounding function prunes all nodes of a subtree rooted at a node that is infeasible because the corresponding bag of transitions does not share a common disabling condition.

Given the GSPN in Figure 1 and the enabling tree in Figure 2b, Figure 2c shows the disabling tree associated with $MEB(t)$. The numbering of the nodes in the tree facilitates the generation of another tree and is not in accordance with the order of the BFS generation. Let $(x_1, x_2, \ldots x_k)$ represent the set of EG nodes associated with node u of the disabling tree. The following steps define the BFS generation of all direct descendents of node u.

Branching Function: For all x_{k+1} such that $x_{k+1} < x_k \leq |V_{EG}|$, generate node v and create an arc from u to v with label x_{k+1}.

Node attributes:

$DB_v = DB_u \cup T_{x_{i+1}}$

$W_D^-(DB_v) = W_D^-(DB_u) \cap W^-(T_{x_{i+1}})$.

$W_D^H(DB_v) = W_D^H(DB_u) \cup W^H(T_{x_{i+1}})$.

Feasibility Constraints: DB_v feasible if

$W_D^-(DB_v) - W^-(\overline{DB_v}) \neq \emptyset \cup W^H(\overline{DB_v}) - W_D^H(DB_v) \neq \emptyset$

Bounding Function: Prune subtree rooted at v, if $\overline{DB_v}$ is not a feasible enabled bag or $W_D^-(DB_v) = \emptyset \cap W_D^H(DB_v)$ contains a multiplicity of infinity for each place.

4.3.3 Disabling Function

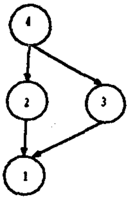
The function $Disab(\cdot)$ maps a bag of transitions into a set of markings such that a marking M is an element of $Disab(T_B)$ if and only if the token count of a single place marking in M disables T_B without disabling the remaining transitions in the MEB. $Disab(T_B)$ defines the set of markings in terms of an expression that is a union of single place marking inequalities. For example, given the disabling tree in Figure 2c, $Disab(\{2t_1, 2t_2\}) = (m_1 < 2) \cup (m_2 < 2)$. Formally, $Disab(T_B) =$

$$\{ M \mid \mathbf{W}^-(\overline{T_B}) \leq \mathbf{M} \ngeq \mathbf{W}_D^-(T_B) \cup \mathbf{W}_D^H(T_B) \nless \mathbf{M} < \mathbf{W}^H(\overline{T_B})\} \qquad (14)$$

In the following section we remove the restriction that the disabling of a feasible bag of disabled transitions must occur through single place marking inequalities.

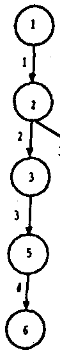
4.3.4 Minimal Disabling Sets

We define a minimal disabling set to be a set of singular disabled bags such that the union of the singular disabled bags over any proper subset of the minimal disabling set does not equal the union over the entire disabling set. Let the bag of transitions T_B equal the union of all the singular disabled bags in a given minimal disabling set. The intersection of the respective disabling functions of all the singular disabled bags defines a set of markings which disables T_B without forcing the disabling of any of the remaining transitions in the MEB. The minimality property ensures that the intersection of the disabling functions associated with the disabled bags in any proper subset of the minimal set results in the disabling of only a proper subbag of T_B.

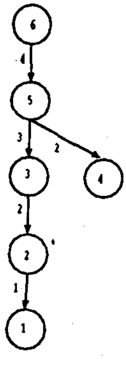


v	T_v
1	t_1
2	t_2
3	$2t_1$
4	$2t_2$

2a



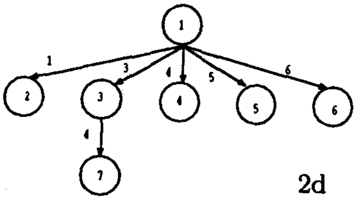
2b



2c

v_{EB}	v_{DB}	EB_v	DB_v	$W^-(EB_v)$	$W_D^-(DB_v)$
1	1	$\emptyset$	$\{t_1t_2\}$	-	$\{p_1p_2\}$
2	2	$\{t_1\}$	$\{2t_1t_2\}$	$\{p_1p_2\}$	$\{p_1p_2\}$
3	3	$\{t_1t_2\}$	$\{2t_12t_2\}$	$\{p_1p_2p_3\}$	$\{2p_12p_2\}$
4	4	$\{2t_1\}$	$\{t_2\}$	$\{2p_12p_2\}$	$\{p_1p_2p_3\}$
5	5	$\{2t_1t_2\}$	$\{2t_2\}$	$\{2p_12p_2p_3\}$	$\{2p_12p_22p_3\}$
6	6	$\{2t_12t_2\}$	$\emptyset$	$\{2p_12p_22p_3\}$	-

Node Attributes



2d

v	MS_v
1	$\emptyset$
2	$\{\{t_1t_2\}\}$
3	$\{\{2t_12t_2\}\}$
4	$\{\{t_2\}\}$
5	$\{\{2t_2\}\}$
6	$\{\{2t_12t_2\}\{t_2\}\}$

Figure 2: Structural Reduction Graphs

The general branch and bound generation of a power set tree provides an efficient method to generate all minimal disabling sets. In this application, the generated minimal tree represents the power set, with pruning, of the set of feasible nodes in the disabling tree. In turn, each feasible node in the disabling tree represents a singular disabled bag.

Efficient evaluation of the feasibility constraint requires a topological ordering of the nodes in the disabling tree such that if node x is less than node y then the singular disabled bag DB_x is not a subbag of DB_y. The numbering of the nodes in decreasing order as they are visited in a BFS traversal, as shown in Figure 2c, achieves the required ordering. Let MS_v be the minimal set represented by node v in the minimal tree and let $T(MS_v)$ be the union of the singular disabled bags in MS_v. The feasibility constraint ensures the minimality property by dictating that any added singular disabled bag must contain a transition not in $T(MS_v)$. Without the specified topological ordering of the disabling tree, the feasibility constraint would also have to ensure none of the singular disabled bags in MS_v were a subbag of this added singular disabled bag.

Clearly the bounding function can prune all subtrees rooted at an infeasible node. Figure 2d shows the minimal tree corresponding to the disabling tree in Figure 2c. Let the k-tuple $(x_1, x_2, \ldots x_i)$ represent minimal set MS_u of disabled bags.

Branching Function: For all x_{i+1} such that $x_i < x_{i+1} \leq |V_{DT}|$, generate node v and create an arc from u to v with label x_{i+1}.

Node attributes:

$MS_v = MS_u \cup \{MDS_{x_{i+1}}\}$.

$T(MS_v) = T(MS_u) \cup MDS_{x_{i+1}}$.

Feasibility Constraints: MS_v is feasible if $DB_{x_{i+1}} \not\subseteq T(MS_u)$.

Bounding Function: prune subtree rooted at node v if MS_v is not minimal.

4.4 Marking Function

The marking function $M(T_B)$ maps a structurally feasible bag of enabled transitions into a set of markings. A marking is an element of $M(T_B)$ if and only if the marking enables all the transitions in T_B and disables all the remaining transitions $\overline{T_B}$. Formally, $M(T_B)$ defines the set of markings in terms of a disjunctive normal form expression of inequalities on places. For example, given $MEB(t_9)$ for the GSPN in Figure 1:

$$M(t_1, t_2) = \{M \mid (1 \leq m_1 < 2 \cap 1 \leq m_2 \cap 1 \leq m_3) \cup (1 \leq m_1 \cap 1 \leq m_2 < 2 \cap 1 \leq m_3)\} \tag{15}$$

The set of markings defined by an 'and' clause of $M(T_B)$ must be a subset of $Enab(T_B)$ to ensure the enabling of T_B. Additional place marking inequalities of an 'and' clause guarantee the disabling of all transitions in $\overline{T_B}$. The number of 'and' clauses represent the different ways to accomplish the disabling of the transitions in $\overline{T_B}$, without the disabling of the transitions in T_B. The generation of minimal disabling sets permits the generation of marking functions with the minimum number of 'and' clauses. Specifically, a minimal disabling set in $MS(\overline{T_B})$ is associated with each 'and' clause such that there exists exactly one place marking inequality for each singular disabling bag that results in the disabling of all the transitions in $\overline{T_B}$. Formally,

$$M(T_B) = \bigcup_{MS \in MS(\overline{T_B})} Enab(T_B) \cap \bigcap_{T_{B'} \in MS} (Disab(T_{B'}). \tag{16}$$

In disjunctive normal form, we represent the k_{th} 'and' clause of $M(T_B)$ with the two bags of places, $M^-(k,T_B)$ and $M^H(k,T_B)$ such that

$$M(T_B) = \{ M \mid \bigcup_k (\mathbf{M}^-(k,T_B) \leq \mathbf{M} \leq \mathbf{M}^H(k,T_B))\}. \tag{17}$$

The marking functions of each structurally feasible bag of enabled transitions in an MEB effectively partitions the set of all markings. In other words, each marking must result in the enabling of exactly one of the feasible enabled bags and the disabling of the remaining transitions in the MEB. However, a marking can satisfy multiple 'and' clauses of the marking function for a given feasible bag of enabled transitions.

4.5 Generation of Replicas

For all structurally feasible bags of transitions T_B enabled directly after the firing of a given timed transition t, for all 'and' clauses k in the marking function of T_B, and for all concurrent transition firing combinations X given T_B, a single transition replication step creates the replicas (t, X, T_B, k). This replica simulates the firing of the timed transition t followed by the concurrent firing of the transitions in X given T_B is the bag of enabled immediate transitions.

Given the marking bags $M^-(k,T_B)$ and $M^H(k,T_B)$, the computations of the input and inhibitor functions for replica (t, X, k, T_B) are straightforward bag operations. The input and inhibitor functions of the replica must permit the enabling of the replicated transition t and upon firing t produce a marking which satisfies the k^{th} 'and' clause of T_B's marking function. The output function of the replica simply produces the change in marking which results from the firing of both the replaced timed transition and the replica's concurrent transition firing combination. Formally, the attributes of replica (t, X, k, T_B) are:

$$\begin{aligned}
W^-(t,X,k,T_B) &= W^-(t) + \left(M^-(k,T_B) - W^+(t)\right) \\
W^H(t,X,k,T_B) &= W^H(t) \cap \left(W^-(t) + \left(M^H(k,T_B) - W^+(t)\right)\right) \\
W^+(t,X,k,T_B) &= \left(W^+(t) - W^-(X)\right) + W^+(X) \\
\Lambda(X,k,T_B) &= \Lambda(t)\text{Prob}\ \{X \text{ fires } \mid T_B \text{ enabled}\ \}.
\end{aligned}$$

4.6 GSPN Decomposition

In this section, we develop a method to decompose the GSPN into subnets, perform structural level reduction on each of these subnets, and aggregate the generated subnet replicas to construct the structurally reduced net corresponding to the original GSPN. This proposed technique is analogous to the state space level reduction method which decomposes a state into immediate submarkings, performs state space evolution of each immediate subnet, and aggregates the resulting tangible submarkings to generate the reachable tangible markings.

4.6.1 Augmented Immediate Subnets

The GSPN decomposition step first augments each immediate subnet to include all timed transition such that the transition's MEB contains a transition in the immediate subnet. Each included timed transition retains its corresponding input and inhibitor functions. The output function of an included timed transition equals the output function of the timed transition after the removal of all places that are not in the given immediate subnet. Each timed transition in an augmented immediate subnet has a firing rate of one.

For each timed transition, the decomposition also creates a subnet consisting solely of the single timed transition. In this subnet, the timed transition retains its input function, inhibitor function, and firing rate. The output function of the timed transition in the subnet equals the output function of the timed transition after the removal of all places that are in any immediate subnet. Figure 3b shows the decomposed GSPN subnets corresponding to the GSPN shown in Figure 3a.

4.6.2 Aggregation of Transition Replicas

The structural reduction of the subnets in the decomposed GSPN and the subsequent aggregation of the replicas among these subnets constructs the same reduced net created by structural reduction of the original GSPN. The Cartesian product of the replicas for a given timed transition in each structurally reduced subnet defines the concurrent replicas for this transition in the structurally reduced net of the original GSPN. In other words, each concurrent replica of a timed transition simulates the concurrent firing of one replica for this timed transition from each of the structurally reduced subnets that contains replicas corresponding to the given timed transition. The input and output functions of a concurrent replica are simply the union of the input and output functions, respectively, of the corresponding replicas in the reduced subnets. Likewise, the inhibitor function of a concurrent replica is simply the intersection of the inhibitor functions of the corresponding replicas in the subnets. The firing rate of the concurrent replica is the product of the firing rates of the corresponding subnet replicas. Figure 3c shows the structurally reduced GSPN corresponding to the GSPN, while Figure 3d shows the structurally reduced subnets corresponding to the decomposed GSPN subnets. The direct derivation of a GSPN's reachability set from the structurally reduced subnets, rather than actual construction of the aggregated structurally reduced GSPN, achieves additional improvements in time and space complexity.

4.7 Time and Space Complexity Analysis

Branch and bound methods avoid the generation of most structurally infeasible replicas, while concurrent replicas avoid the generation of most redundant replicas. In addition, the systematic generation of the tree structures, which represent each replica's attributes, permits the efficient determination of feasible replicas. In [5], we provide theoretical complexity analysis for the branch

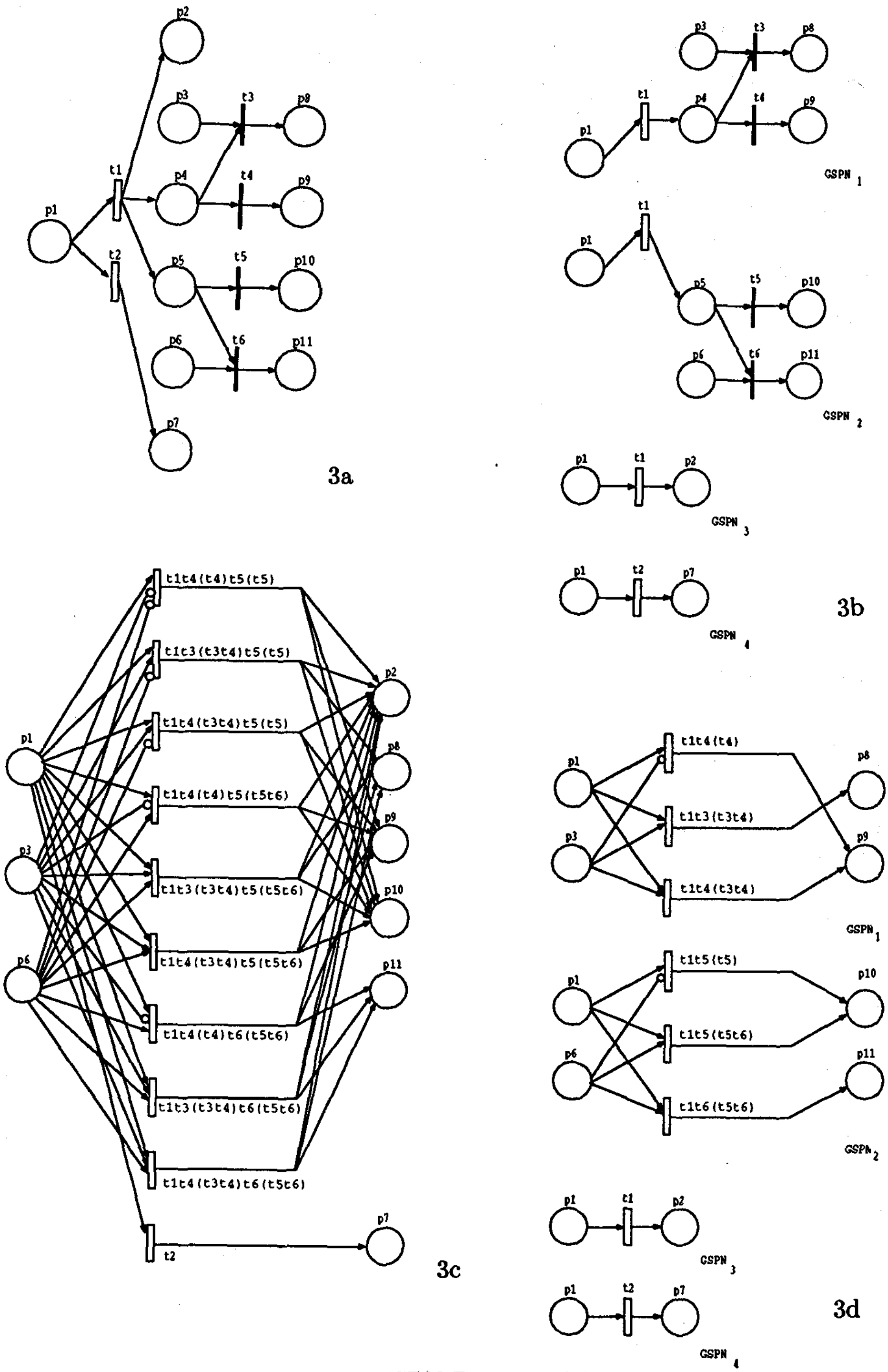


Figure 3: GSPN Decomposition

and bound generation of non-concurrent replicas and the complexity analysis for state space level reduction is directly applicable to the generation of concurrent replicas.

5 Conclusion

The contribution of this paper is to provide algorithms and corresponding data structures to efficiently implement both state space and structural level reduction on the GSPN model. Structural level reduction has some inherent advantages over state space level reduction. Since the GSPN structure can provide a factorization of state space transitions into significantly fewer net transitions, there is the potential for a corresponding factorization of complexity between state space and structural level reduction. In other words, the elimination of a single immediate transition can achieve a reduction equivalent to the elimination of several state space transitions and their adjacent vanishing markings. On the other hand, structural level reduction may generate replicas that are not enabled by any marking in the reachability set.

References

[1] G. Balbo, G. Chiola, G. Franceschinis, and G. Roet. On the Efficient Construction of the Tangible Reachability Graph of Generalized Stochastic Petri Nets. In *Proceedings of the International Workshop on Petri Nets and Performance Models*, pages 136–145, August 1987.

[2] Giovanni Chiola, Susanna Donatelli, and Giuliana Franceschinis. GSPNs versus SPNs: What is the Actual Role of Immediate Transitions? In *Proceedings of the Fourth International Workshop on Petri Nets and Performance Models*, pages 20–31, December 1991.

[3] Giovanni Chiola, Marco Marsan, Gianfranco Balbo, and Gianni Conte. Generalized Stochastic Petri Nets: A Definition at the Net Level and its Implications. *IEEE Transactions on Software Engineering*, 19(2):89–107, February 1993.

[4] Mark A. Holliday and Mary K. Vernon. A Generalized Timed Petri Net Model for Performance Analysis. *IEEE Transactions on Software Engineering*, SE-13(12):1297–1310, December 1987.

[5] Donna S. Nielsen. *Performance Evaluation of Distributed Systems Using Structural Level Analysis of Performance Petri Nets*. PhD thesis, University of California, Los Angeles, June 1994.

[6] C. Simone and M. Ajmone Marsan. The Application of EB-Equivalence Rules to the Structural Reduction of GSPN Models. *Journal of Parallel and Distributed Computing*, 15(3):296–302, July 1992.