Clean choice-concurrency Petri nets.

Alexandre B. Smirnov

3rd December 2001

1 Introduction

Following the [Bes87] consider PN properties as to belong to two groups: *behavioral* - marking-dependent properties defined using the PN execution rules, its reachability graph (RG) or any implications of the net dynamic behavior, and *structural* - marking-independent (with the only exception for sometimes considered initial marking) properties depending on the interconnection of the PN nodes.

Behavioral properties are those we are usually interested in, but due to the dynamic nature and usually exponential size of the PN RG (relative to the PN size) these properties are hard to analyze.

Structural properties' analysis is computationally easier for it only uses the usually compact PN structure - interconnection of the net vertices. Various PN classes have been proposed over the time from trivial to complicated with different correlation between the net structure and its behavior. For instance for state machines (SM) - PNs with no concurrency, the RG structure coincides with that of a PN.

Many conjectures relating the PN structural and behavioral properties hold for yet non-trivial extensively studied class of Petri nets called free-choice (FC) PNs.

In this work we attempt to mark out a PNs class that always mimics (while does not coincide with) the structure of its RG making the PN behavior clear from the net structure on one hand and still not trivial on the other. To facilitate the dynamic behavior analysis we rely not only on the net structure but also on the precomputed PN concurrency and choice relations.

2 CCC Petri nets.

Formally *Petri net (PN)* is a quadruple $\mathcal{N} = (P, T, F, m_0)$, where the P, T ($P \cup T = V$) are disjoint sets of vertices (v_i) called *places* (p_i) and *transitions* (t_i), F - set of arcs (f_i) ($F \subseteq P \times T \cup T \times P$) and m_0 is an *initial marking* ($m_0 \subset P$).

PN $\mathcal{N}$ is called *free-choice (FC)* if $\forall t_i, t_j \in T, t_i \neq t_j, \bullet(\bullet t_i) \cap \bullet(\bullet t_j) \not\equiv \emptyset \implies |\bullet(\bullet t_i)| = 1 = |(t_j\bullet)\bullet|$, where $\bullet t_i$ ($t_i\bullet$) denotes an arc input (output) for the transition t_i while $\bullet(\bullet t_i)$ ($(t_i\bullet)\bullet$) denotes its input (output) place.

Recall the PN *behavioral concurrency relations BCR* definition - two places p_i, p_j are *concurrent* if $\exists \mathsf{m}_k \in [\mathsf{m}_0\rangle \mid p_i, p_j \in \mathsf{m}_k$ i.e. there exists a reachable marking, such that both places are marked simultaneously. In turn two *transitions are concurrent* if they can occur concurrently from some reachable marking.

We represent the *structural concurrency relations (SCR)* by $\mathsf{CcS}(\epsilon_i)$ - a set of PN elements $\epsilon_j \in P \cup T \cup F$ concurrent to an element ϵ_i.

Definition 2.1 *PN SCRs.*
Following [KE96] the structural concurrency relations (SCR) *for PN nodes are:*

(i) $\forall p_i, p_j \in P : p_i \| p_j$ *iff* $p_i, p_j \in \mathsf{m}_0$ *i.e. the places marked in the initial marking are concurrent;*

(ii) $\forall p_i, p_j \in \bullet(\bullet t_k) : p_i \| p_j \Longrightarrow \forall p_l, p_m \in (t_k \bullet)\bullet : p_l \| p_m$ *i.e. if all input places of a transition are mutually concurrent - so are its output places;*

(iii) $\forall v_i \in P \cup T, t_j \in T : \bullet(\bullet t_j) \subseteq \mathsf{CcS}(v_i) \Longrightarrow (t_j \bullet)\bullet \subseteq \mathsf{CcS}(v_i)$ *i.e. a node concurrent to all input places of a transition is concurrent to all its output places;*

In [KE96] these structural concurrency relations are shown to be generally a superset of the BCR and coincide with the latter for live safe FC nets. Polynomial complexity algorithms are proposed for computing the PN SCR defined above: in [Kov92] - of complexity $O(n^5)$ for an arbitrary PN, in [KE96] - $O(n^4)$ - for live PNs and if the PN is also FC and bounded - of complexity $O(n^3)$, where $n = |P| + |T|$. We extend the SCR defined in 2.1 with the CR for PN arcs :

Definition 2.2 *PN SCRs for arcs.*
Let $t(f_i)$ *be a transition being a head or a tail of the arc* f_i*. Then for PN arcs:*

(i) $\forall f_i, f_j \in F : f_i \| f_j$ *iff* $(\bullet f_j = \bullet f_i \vee f_i \bullet = f_j \bullet) \in T \vee t(f_i) \| t(f_j)$ *i.e. arcs are concurrent if their tail or head are concurrent or is the same transition;*

(ii) $\forall v_i \in P \cup T, f_j \in F : v_i \| f_j$ *iff* $\bullet f_j \| v_i \wedge f_j \bullet \| v_i$ *i.e. an arc is concurrent to a node if its head and tail are concurrent to that node;*

Based on the definitions 2.1, 2.2 we define CCC PN as:

Definition 2.3 *CCC Petri net.*
A safe, live Petri net $\mathcal{N} = (P, T, F)$ *is a* CCC net *if* $\forall f_i, f_j \in \bullet p_k : \mathsf{CcS}(f_i) \equiv \mathsf{CcS}(f_j) \wedge \forall f_i, f_j \in p_k \bullet : \mathsf{CcS}(f_i) \equiv \mathsf{CcS}(f_j)$ *i.e. the concurrency relations are identical for all input and for all output arcs of any PN place.*

To check if a PN is CCC we rely on the algorithms from [Kov92], [KE96] and 2.2 to decide safeness and obtain the concurrency relations for the net.

3 CCC PN properties.

To show that every live, safe, connected PN can be transformed to the form of CCC PN using the notion of simulation defined in [Bes87] we prove the following:

Theorem 3.1 (Equivalence of CCC and FC Petri nets w.r.t. simulation.)
Every live, safe, connected PN can be simulated by CCC PN.

For a Petri net $\mathcal{N} = (P, T, F)$ let *dual* net be $\mathcal{N}^d = (P^d, T^d, F^d) = (T, P, F^{-1})$ i.e. $\forall v_i \in \mathcal{N} : \exists v_i^d \in \mathcal{N}^d \mid v_i \in P, v_i^d \in T^d : v_i \overset{d}{\longleftrightarrow} v_i^d$ and vice versa: $v_i \in T, v_i^d \in P^d : v_i \overset{d}{\longleftrightarrow} v_i^d$ and the corresponding arcs are reversed.

To define the behavioral choice relations suppose we color in all feasible traces (sequences of event firings agreeing with the PN execution rules) of $\mathcal{N}$ the spans where a place $p_k \in P$ is marked. Then transitions t_l, t_m are said to be in *choice* relation ($t_l \asymp t_m$) if in neither feasible trace t_l, t_m are encountered in the same uncolored span. If such a place p_k exists it is called the *choice branching point*.

Theorem 3.2 (CCC Petri net properties.)
Let $\mathcal{N} = (P, T, F)$ be a live, safe, CCC Petri net. Then:

(i) for the dual PN $\mathcal{N}^d$ there exists an initial marking m_0' such that $\mathcal{N}^d$ also live, safe and CCC;

(ii) $\forall \epsilon_i, \epsilon_j \in \mathcal{N} : \epsilon_i \| \epsilon_j \Leftrightarrow \epsilon_i^d \asymp \epsilon_j^d$ i.e. any two concurrent net elements of a CCC net $\mathcal{N}$ are in choice relation in the net $\mathcal{N}^d$ are and vice versa;

(iii) $\forall \epsilon_i, \epsilon_i \in \mathcal{N} : \epsilon_i \| \epsilon_j \implies \epsilon_i \not\asymp \epsilon_j \wedge \epsilon_i \asymp \epsilon_j \implies \epsilon_i \not\| \epsilon_j$ i.e. concurrency relations never intersect with choice relations;

Some other properties proven for FC PNs also hold for CCC PNs.

Among those that distinguish the CCC nets is the possibility of locally deciding using the precomputed structural concurrency and choice relations if certain transformation (such as node/arc insertion/removal) would preserve the PN CCC (and therefore liveness and safeness of the net). Moreover the concurrency and choice relations can be 'locally' recomputed for every PN modification that preserves CCC. These properties have been partially used in [SSKG01] and [Tax] for interactive refinement.

4 CCC PN applications.

We used CCC PN for structural logic synthesis from *Signal Transition Graph (STG)* specification [KSS98] as well as STG analysis and refinement [SSKG01]. (CCC has been applied for STGs and called respectively Unique Partial State (UPS) and Unique State Factorization (USF) in these works.)

Indeed any STG with underlying CCC PN features the following property important for structural STG analysis, interactive refinement and logic synthesis.

Let *STG* be $\mathcal{G} = (\mathcal{N}, Y, \Delta)$ where $\mathcal{N}$ is a PN, $Y = X \cup Z$ is a set of signals (environment and circuit respectively) and $\Delta : T' \to (Y) \times \{+, -\}$ is a mapping of the PN transitions on signal transitions.

Let *subcut* C be a set of mutually concurrent places and a *cut* $\mathsf{C}^{\mathsf{max}}$ - a maximal subcut. We call a (sub)cut $\mathsf{C_i}$ *active* for a reachable marking $\mathsf{m}_j \in [\mathsf{m_0}\rangle$ when $\mathsf{C_i} \subseteq \mathsf{m}_j$ i.e. all its places are marked (contain a marker). Then as soon as a marking uniquely represents the state of the PN and therefore of the system modeled by PN if a cut is active the state of all STG signals is defined and known. In turn a subcut being active represents a set of PN markings.

Lemma 4.1 *Let $\mathcal{G} = (\mathcal{N}, Y, \Delta)$ be an STG and its underlying net be a CCC PN. Then for any* m_i *such that* C *is active the state of signals* $y \in Y$ *is uniquely defined (or always undefined).*

Our experience of using the CCC PNs in [Tax] makes us believe that besides the STG analysis, interactive refinement and structural logic synthesis the PN class presented here can be advantageous for other applications requiring a representation of causal behavior with concurrency and choice featuring:

- simplicity for human perception,
- reduced complexity of the behavioral properties' analysis,
- possibility of 'on-line' concurrency and choice recomputation for certain transformations allowing for iterative manipulation of the behavior specification.

References

[Bes87] E. Best. Structure theory of petri nets: the free choice hiatus. In Brauer, W., Reisig, W., and Rozenberg, G., editors, *Lecture Notes in Computer Science: Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part I, Proceedings of an Advanced Course, Bad Honnef, September 1986*, volume 254, pages 168–205. Springer-Verlag, 1987. NewsletterInfo: 27.

[KE96] A. Kovalyov and J. Esparza. A polynomial algorithm to compute the concurrency relation of free-choice signal transition graphs. In *Proceedings of the International Workshop on Discrete Event Syst ems, WODES'96*, pages 1–6, August 1996.

[Kov92] A. Kovalyov. Concurrency relations and the safety problem for petri nets. In *Proceedings of the International Conference on Application and Theory of Petri Nets, WODES'96*, August 1992.

[KSS98] Ilya V. Krotchkov, Alexander B. Smirnov, and Nikolai A. Starodoubtsev. Verification driven synthesis of asynchronous circuits from STG specification. In Anne-Marie Trullemans-Anckaert and Jens Sparsø, editors, *Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pages 377–386, October 1998.

[SSKG01] Alexandre B Smirnov, Nikolai Starodoubtsev, Ilya V Klotchkov, and Michael Goncharov. A technique to automate stg analysis and refinement for csc and normalcy. In *Proc. Power and Timing Modeling, Optimization and Simulation (PATMOS)*, September 2001.

[Tax] TaxoSynthesis CAD tool. `http://www.lsi.upc.es/~alexs/synth/`.