Thomas Hujsa
Thomas.Hujsa@onera.fr

This form is a summary description of the model entitled "RefineWMG" proposed for the Model Checking Contest @ Petri Nets. Models can be given in several instances parameterized by scaling parameters. Colored nets can be accompanied by one or many equivalent, unfolded P/T nets. Models are given together with property files (possibly, one per model instance) giving a set of properties to be checked on the model.

Description

Weighted Marked Graphs (WMGs for short) form a well-known and useful subclass of weighted Petri nets in which each place has at most one input and one output. They can model Synchronous Dataflow Graphs [1] (SDFGs), which have been used to model various applications [2,3], notably for designing embedded systems such as Digital Signal Processing (DSPs) applications. These graphs benefit from efficient methods checking their structure and behavior [4,5,6].

However, in general, behavioral properties that are fundamental for many real-world applications, such as liveness and boundedness, remain hard to check in larger classes of Petri nets. So as to alleviate this difficulty, they are often analyzed together with their structural version, namely structural liveness and structural boundedness, which designate respectively the existence of a live marking and the boundedness for every marking. Taken together, structural liveness and structural boundedness define well-formedness. In the sequel, we focus on well-formed nets.

The homogeneous asymmetric-choice (HAC) nets form a subclass of weighted Petri nets in which each pair p, p' of input places of any synchronization satisfies the following: all the outputs of p are also outputs of p', or conversely; moreover, the homogeneity constraint specifies that for each place p, all the output weights of p are equal. This class generalizes the WMGs and benefits from several structural and behavioral results [7,8,9,10]: some polynomial-time sufficient conditions ensure liveness and boundedness together with their monotonicity, meaning their preservation upon any increase of the marking considered. Such results often rely on decompositions into well-formed join-free components, i.e. well-formed Petri subnets without synchronizations.

The aim is to define first the underlying WMG structure of the system, whose properties are pretty well understood, and then to specify the behavior of each process as a module. Figure 1 pictures such a model, refining some processes of a WMG by merging the corresponding transitions with well-formed, unit-weighted, HAC nets. The marking provided is monotonically live and bounded for each $n \geq 1$.

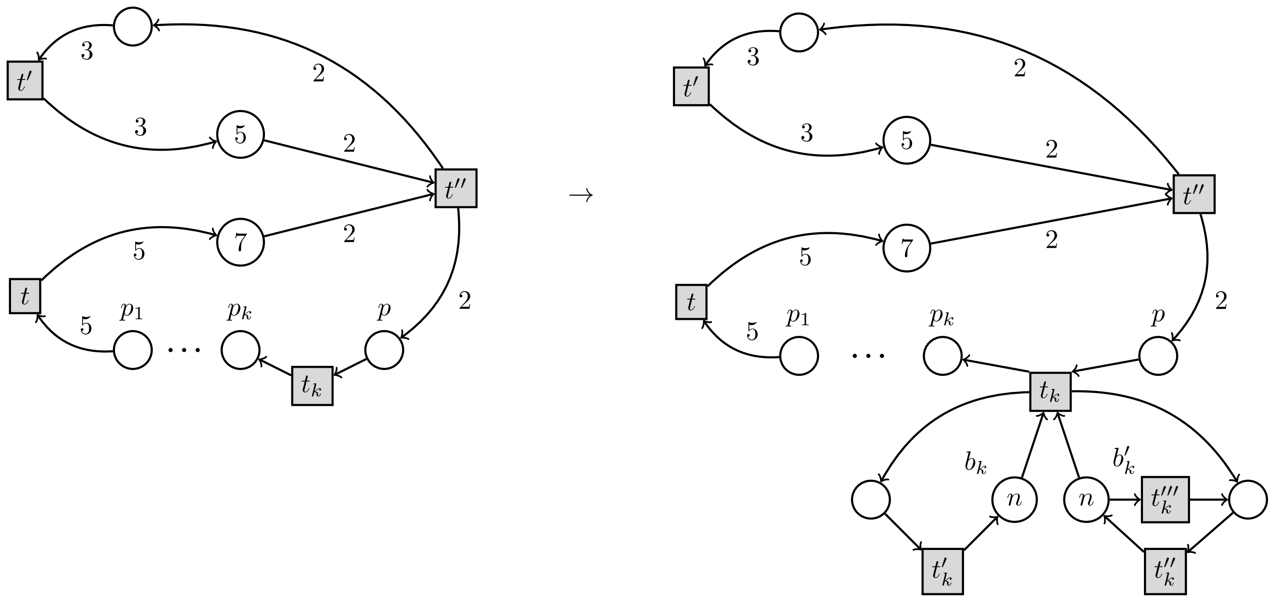

Figure 1: The system on the left is a non-refined WMG with $k \geq 1$ transitions and places on the path $p, t_k, p_k \ldots, p_1$ from p to t. On the right, a refinement of the WMG on the left is obtained by merging each transition t_i on this path, $i \in \{1, \ldots, k\}$, with a transition of the i-th copy N_i of a unit-weighted HAC system. Two buffers (places) b_i and b'_i of N_i contain n tokens each. The HAC system obtained is live and bounded for each $n \geq 1$.

Thomas Hujsa
Thomas.Hujsa@onera.fr

The model is based on two parameters k and n:

- k denotes the number of HAC modules,
- n is the initial number of tokens in each buffer $b_1, b'_1 \ldots, b_k, b'_k$.

This model has been regenerated in April 2019 for the 2020 edition of the MCC and onwards, because of a bug of disappearing arcs, fixed in the instances from 010-010 to 050-051.

References

[1] E. A. Lee and D. G. Messerschmitt. *Synchronous dataflow.* Proceedings of the IEEE, 75(9), pages 1235-1245, 1987.

[2] J. L. Pino, S. S. Bhattacharyya, and E. A. Lee. *A hierarchical multiprocessor scheduling framework for synchronous dataflow graphs.* Technical report, University of California, Berkeley, 1995.

[3] S. Sriram and S. Bhattacharyya. *Embedded multiprocessors: Scheduling and synchronization.* CRC, 2009.

[4] E. Teruel, P. Chrzastowski-Wachtel, J. M. Colom, and M. Silva, *On Weighted T-systems*, in Application and Theory of Petri Nets, pages 348-367, 1992.

[5] O. Marchetti and A. Munier-Kordon, *A sufficient condition for the liveness of weighted event graphs*, European Journal of Operational Research, vol. 197, no. 2, pages 532-540, 2009.

[6] R. Devillers and T. Hujsa. *Analysis and Synthesis of Weighted Marked Graph Petri Nets.* To appear in the proceedings of the 39th International Conference on Application and Theory of Petri Nets and Concurrency (Petri Nets'18), Springer International Publishing, 21 pages, 2018.

[7] L. Jiao, T.Y. Cheung, W. Lu. *On liveness and boundedness of asymmetric choice nets.* Theor. Comput. Sci. 311(1-3), pages 165-197, 2004.

[8] T. Hujsa and R. Devillers. *On Liveness and Deadlockability in Subclasses of Weighted Petri Nets.* In proceedings of the 38th International Conference on Application and Theory of Petri Nets and Concurrency (Petri Nets'17), Springer International Publishing, pages 267-287, 2017.

[9] T. Hujsa. *Contribution to the study of weighted Petri nets.* PhD thesis, 2014.

[10] E. Teruel and M. Silva. *Structure theory of Equal Conflict systems.* Theoretical Computer Science, 153(1&2): 271-300, 1996.

Scaling parameter

Parameter name	Parameter description	Chosen parameter values
k, n	k, the number of modules, and n, the number of initial tokens in each buffer b_i, b'_i in the module N_i for each $i \in \{1, \ldots, k\}$.	$(2,2), (2,3), (5,5), (5,6), (7,7), (7,8), (10,10), (10,11), (15,15), (15,16), (25,25), (25,26), (50,50), (50,51), (100,100), (100,101)$

Size of the model

Parameter	Number of places	Number of transitions	Number of arcs
$k = 2, n = 2$	14	11	32
$k = 2, n = 3$	14	11	32
$k = 5, n = 5$	29	23	68
$k = 5, n = 6$	29	23	68
$k = 7, n = 7$	39	31	92
$k = 7, n = 8$	39	31	92
$k = 10, n = 10$	54	43	128
$k = 10, n = 11$	54	43	128
$k = 15, n = 15$	79	63	188
$k = 15, n = 16$	79	63	188
$k = 25, n = 25$	129	103	308
$k = 25, n = 26$	129	103	308
$k = 50, n = 50$	254	203	608
$k = 50, n = 51$	254	203	608
$k = 100, n = 100$	504	403	1208
$k = 100, n = 101$	504	403	1208

Structural properties

ordinary — *all arcs have multiplicity one* .... ✗
simple free choice — *all transitions sharing a common input place have no other input place* .... ✗ [a]
extended free choice — *all transitions sharing a common input place have the same input places* .... ✗ [b]
state machine — *every transition has exactly one input place and exactly one output place* .... ✗ [c]
marked graph — *every place has exactly one input transition and exactly one output transition* .... ✗ [d]
connected — *there is an undirected path between every two nodes (places or transitions)* .... ✔ [e]
strongly connected — *there is a directed path between every two nodes (places or transitions)* .... ✔ [f]
source place(s) — *one or more places have no input transitions* .... ✗ [g]
sink place(s) — *one or more places have no output transitions* .... ✗ [h]
source transition(s) — *one or more transitions have no input places* .... ✗ [i]
sink transitions(s) — *one or more transitions have no output places* .... ✗ [j]
loop-free — *no transition has an input place that is also an output place* .... ✔ [k]
conservative — *for each transition, the number of input arcs equals the number of output arcs* .... ✔ [l]
subconservative — *for each transition, the number of input arcs equals or exceeds the number of output arcs* .... ✔ [m]
nested units — *places are structured into hierarchically nested sequential units* [n] .... ✗

[a] the net is not ordinary.
[b] the net is not ordinary.
[c] the net is not ordinary.
[d] the net is not ordinary.
[e] stated by CÆSAR.BDD version 3.3 on all 16 instances (see all aforementioned parameter values).
[f] stated by CÆSAR.BDD version 3.3 on all 16 instances (see all aforementioned parameter values).
[g] stated by CÆSAR.BDD version 2.7 on all 16 instances (see all aforementioned parameter values).
[h] stated by CÆSAR.BDD version 3.3 on all 16 instances (see all aforementioned parameter values).
[i] stated by CÆSAR.BDD version 2.7 on all 16 instances (see all aforementioned parameter values).
[j] stated by CÆSAR.BDD version 2.7 on all 16 instances (see all aforementioned parameter values).
[k] stated by CÆSAR.BDD version 2.7 on all 16 instances (see all aforementioned parameter values).
[l] stated by PNML2NUPN 2.3.0 on all 16 instances (see all aforementioned parameter values).
[m] stated by PNML2NUPN 2.3.0 on all 16 instances (see all aforementioned parameter values).
[n] the definition of Nested-Unit Petri Nets (NUPN) is available from `http://mcc.lip6.fr/nupn.php`

Thomas Hujsa
Thomas.Hujsa@onera.fr

Behavioural properties

safe — *in every reachable marking, there is no more than one token on a place* .... ✗ (o)
dead place(s) — *one or more places have no token in any reachable marking* .... ?
dead transition(s) — *one or more transitions cannot fire from any reachable marking* .... ✗
deadlock — *there exists a reachable marking from which no transition can be fired* .... ✗
reversible — *from every reachable marking, there is a transition path going back to the initial marking* .... ✓
live — *for every transition t, from every reachable marking, one can reach a marking in which t can fire* .... ✓

Size of the marking graphs

Parameter	Number of reachable markings	Number of transition firings	Max. number of tokens per place	Max. number of tokens per marking
$k = 2, n = 2$	58 320 (p)	321 732 (q)	≤ 7 (r)	20 (s)
$k = 2, n = 3$	?	?	?	24 (t)
$k = 5, n = 5$	?	?	≤ 7 (u)	62 (v)
$k = 5, n = 6$	?	?	?	72 (w)
$k = 7, n = 7$	?	?	?	110 (x)
$k = 7, n = 8$	?	?	?	124 (y)
$k = 10, n = 10$	?	?	?	212 (z)
$k = 10, n = 11$	?	?	?	232 (aa)
$k = 15, n = 15$	?	?	?	462 (ab)
$k = 15, n = 16$	?	?	?	492 (ac)
$k = 25, n = 25$	?	?	?	1262 (ad)
$k = 25, n = 26$	?	?	?	1312 (ae)
$k = 50, n = 50$	?	?	?	5012 (af)
$k = 50, n = 51$	?	?	?	5112 (ag)
$k = 100, n = 100$	?	?	?	20012 (ah)
$k = 100, n = 101$	?	?	?	20212 (ai)

Other properties

P- and T- semiflows exist and cover the net. The system is live, bounded and reversible for each $n \geq 1$.

(o) stated by CÆSAR.BDD version 2.7 on all 16 instances (see all aforementioned parameter values).
(p) value provided by the author of the model.
(q) value provided by the author of the model.
(r) value provided by the author of the model.
(s) number of initial tokens, because the net is conservative.
(t) number of initial tokens, because the net is conservative.
(u) value provided by the author of the model.
(v) number of initial tokens, because the net is conservative.
(w) number of initial tokens, because the net is conservative.
(x) number of initial tokens, because the net is conservative.
(y) number of initial tokens, because the net is conservative.
(z) number of initial tokens, because the net is conservative.
(aa) number of initial tokens, because the net is conservative.
(ab) number of initial tokens, because the net is conservative.
(ac) number of initial tokens, because the net is conservative.
(ad) number of initial tokens, because the net is conservative.
(ae) number of initial tokens, because the net is conservative.
(af) number of initial tokens, because the net is conservative.
(ag) number of initial tokens, because the net is conservative.
(ah) number of initial tokens, because the net is conservative.
(ai) number of initial tokens, because the net is conservative.