Model: DoubleLock
Type: P/T Net
Origin: Academic

since
MCC 2022

Y. Thierry-Mieg
yann.thierry-mieg@lip6.fr

This form is a summary description of the model entitled "DoubleLock" 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

This example is part of a suite that consists of 46 Petri nets that were used in the evaluation of BFC [1]. They originate from the analysis of concurrent C programs.

These examples model programs using multiple locks to control access to a shared resource.

This model was then used as one of the benchmarks for the tool Petrinizer in [2]. Models found in [3] where converted to PNML thanks to an ITS-Tools [4] library.

References

1. A. Kaiser, D. Kroening, and T. Wahl. Efficient coverability analysis by proof minimization. In CONCUR, volume 7454 of Lecture Notes in Computer Science, pages 500–515. Springer, 2012

2. J. Esparza, R. Ledesma-Garza, R. Majumdar, P. J. Meyer, and F. Niksic. An smt-based approach to coverability analysis. In CAV, volume 8559 of Lecture Notes in Computer Science, pages 603–619. Springer, 2014

3. Klara J. Meyer, Petrinizer repository, https://github.com/meyerphi/petrinizer.

4. Y. Thierry-Mieg, Homepage of ITS-tools https://lip6.github.io/ITSTools-web/

Scaling parameter

Parameter name	Parameter description	Chosen parameter values
p, s	parameters taken from the initial specifications	(1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)

Size of the model

Parameter	Number of places	Number of transitions	Number of arcs
(1, 1)	64	204	828
(1, 2)	570	7 600	30 784
(1, 2)	570	7 568	30 656
(2, 1)	64	212	860
(2, 2)	184	1 832	7 424
(3, 1)	46	80	324
(3, 2)	112	744	3 008
(3, 3)	306	3 136	12 672

Structural properties

ordinary — *all arcs have multiplicity one* .... yes
simple free choice — *all transitions sharing a common input place have no other input place* .... no [a]
extended free choice — *all transitions sharing a common input place have the same input places* .... no [b]

[a] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[b] transitions "t0" and "t2" share a common input place "s0", but only the former transition has input place "l0".

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

Behavioural properties

safe — *in every reachable marking, there is no more than one token on a place* .... no [o]
dead place(s) — *one or more places have no token in any reachable marking* .... yes [p]
dead transition(s) — *one or more transitions cannot fire from any reachable marking* .... yes [q]
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
(1, 1)	?	?	?	≥ 11 [r]
(1, 2)	?	?	?	≥ 11 [s]
(1, 3)	?	?	?	≥ 11 [t]
(2, 1)	?	?	?	≥ 11 [u]
(2, 2)	?	?	?	≥ 11 [v]
(3, 1)	?	?	?	≥ 11 [w]
(3, 2)	?	?	?	≥ 11 [x]
(3, 3)	?	?	?	≥ 11 [y]

[c] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[d] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[e] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[f] the net is not connected and, thus, not strongly connected.
[g] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[h] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[i] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[j] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[k] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[l] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[m] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[n] the definition of Nested-Unit Petri Nets (NUPN) is available from http://mcc.lip6.fr/nupn.php
[o] in the initial marking, there exists one place containing 10 tokens.
[p] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[q] stated by CÆSAR.BDD version 3.7 on all 8 instances ((1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (3, 2), (3, 2)).
[r] lower bound given by the number of initial tokens.
[s] lower bound given by the number of initial tokens.
[t] lower bound given by the number of initial tokens.
[u] lower bound given by the number of initial tokens.
[v] lower bound given by the number of initial tokens.
[w] lower bound given by the number of initial tokens.
[x] lower bound given by the number of initial tokens.
[y] lower bound given by the number of initial tokens.