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

since
MCC 2021

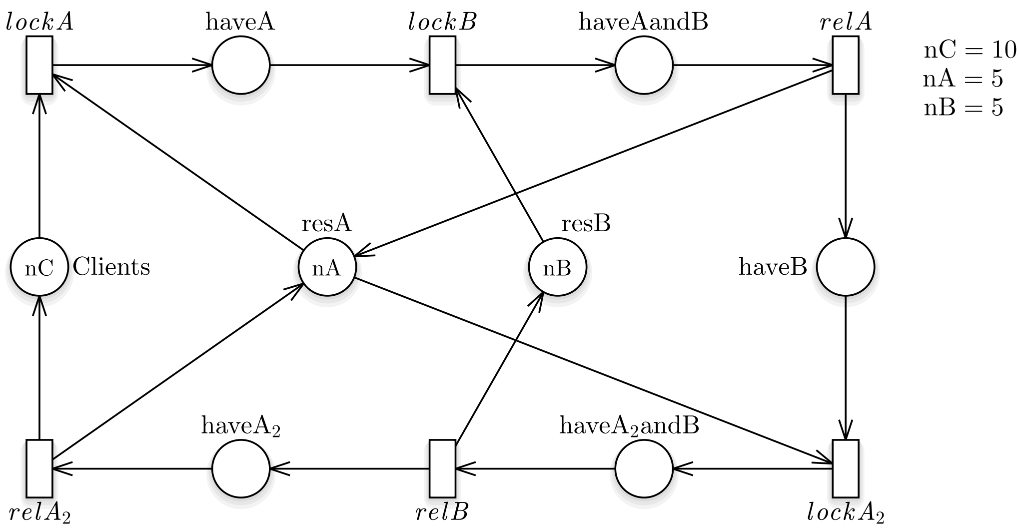
Elvio G. Amparore
amparore@di.unito.it

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

The model simulates a problematic condition where a badly-designed process violates the *two phase locking* (2PL) protocol rules. A process performing 2PL follows two phases: an *acquisition phase*, where resource can be obtained, and a *release phase*, where all resources must be released. Re-acquiring resources during the release phase is a 2PL protocol violation. 2PL, together with fixed-order resource acquisition, ensures deadlock avoidance.

In the Petri net model, a client process first acquires a resource of type A and one of type B. It then releases A, thus starting the release phase. However, after this first step, the process reacquires a new resource of type A, violating the 2PL rules. The process that releases both B and A. If the number of concurrently running *Clients* nC is equal or less than the sum of the resources $nA + nB$, a deadlock condition may form. The model is parametric in nC, the number of clients. For each value of nC, two model versions are proposed: Version N has $nC = 2 \cdot nA = 2 \cdot (nB - 1)$, resulting in no deadlocks; Version D has $nC = 2 \cdot nA = 2 \cdot nB$, generating deadlock states.



Graphical representation for $nC = 10$ (version D). Version N would have $nB = 6$.

References

Philip A. Bernstein, Vassos Hadzilacos, Nathan Goodman (1987): *Concurrency Control and Recovery in Database Systems*, Addison Wesley Publishing Company, ISBN 0-201-10715-5.

Scaling parameter

Parameter name	Parameter description	Chosen parameter values
N	Number of client processes.	4, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000

Elvio G. Amparore
amparore@di.unito.it

Size of the model

Although the model is parameterized, its size does not depend on parameter values.

number of places:	8
number of transitions:	6
number of arcs:	18

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) ✗

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* ✗ (p)
dead transition(s) — *one or more transitions cannot fire from any reachable marking* ✗ (q)
deadlock — *there exists a reachable marking from which no transition can be fired* ? (r)
reversible — *from every reachable marking, there is a transition path going back to the initial marking* ? (s)
live — *for every transition t, from every reachable marking, one can reach a marking in which t can fire* ? (t)

(a) 2 arcs are not simple free choice, e.g., the arc from place "resA" (which has 2 outgoing transitions) to transition "lockA" (which has 2 input places).
(b) transitions "lockA2" and "lockA" share a common input place "resA", but only the former transition has input place "haveB".
(c) 6 transitions are not of a state machine, e.g., transition "relB".
(d) place "resA" is not of a marked graph.
(e) stated by CÆSAR.BDD version 3.5 on all 22 instances ($nC \in \{4, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$, version D or N).
(f) stated by CÆSAR.BDD version 3.5 on all 22 instances ($nC \in \{4, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$, version D or N).
(g) stated by CÆSAR.BDD version 3.5 on all 22 instances ($nC \in \{4, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$, version D or N).
(h) stated by CÆSAR.BDD version 3.5 on all 22 instances ($nC \in \{4, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$, version D or N).
(i) stated by CÆSAR.BDD version 3.5 on all 22 instances ($nC \in \{4, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$, version D or N).
(j) stated by CÆSAR.BDD version 3.5 on all 22 instances ($nC \in \{4, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$, version D or N).
(k) stated by CÆSAR.BDD version 3.5 on all 22 instances ($nC \in \{4, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$, version D or N).
(l) 6 transitions are not conservative, e.g., transition "relB".
(m) 3 transitions are not subconservative, e.g., transition "relB".
(n) the definition of Nested-Unit Petri Nets (NUPN) is available from `http://mcc.lip6.fr/nupn.php`
(o) stated by CÆSAR.BDD version 3.5 on all 22 instances ($nC \in \{4, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$, version D or N).
(p) stated by CÆSAR.BDD version 3.5 on all 22 instances ($nC \in \{4, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$, version D or N).
(q) stated by CÆSAR.BDD version 3.5 on all 22 instances ($nC \in \{4, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$, version D or N).
(r) ✔ for the D version, ✗ for the N version.
(s) ✔ for the D version, ✗ for the N version.
(t) ✔ for the D version, ✗ for the N version.

Elvio G. Amparore
amparore@di.unito.it

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
$nC = 4$ version D	32	57	4	8
$nC = 4$ version N	45	84	4	9
$nC = 10$ version D	?	?	?	≥ 20 (u)
$nC = 10$ version N	?	?	?	≥ 21 (v)
$nC = 20$ version D	?	?	?	≥ 40 (w)
$nC = 20$ version N	?	?	?	≥ 41 (x)
$nC = 50$ version D	?	?	?	≥ 100 (y)
$nC = 50$ version N	?	?	?	≥ 101 (z)
$nC = 100$ version D	?	?	?	≥ 200 (aa)
$nC = 100$ version N	?	?	?	≥ 201 (ab)
$nC = 200$ version D	?	?	?	≥ 400 (ac)
$nC = 200$ version N	?	?	?	≥ 401 (ad)
$nC = 500$ version D	?	?	?	≥ 1000 (ae)
$nC = 500$ version N	?	?	?	≥ 1001 (af)
$nC = 1000$ version D	?	?	?	≥ 2000 (ag)
$nC = 1000$ version N	?	?	?	≥ 2001 (ah)
$nC = 2000$ version D	?	?	?	≥ 4000 (ai)
$nC = 2000$ version N	?	?	?	≥ 4001 (aj)
$nC = 5000$ version D	?	?	?	≥ 10000 (ak)
$nC = 5000$ version N	?	?	?	≥ 10001 (al)
$nC = 10000$ version D	?	?	?	≥ 20000 (am)
$nC = 10000$ version N	?	?	?	≥ 20001 (an)

(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.
(z) lower bound given by the number of initial tokens.
(aa) lower bound given by the number of initial tokens.
(ab) lower bound given by the number of initial tokens.
(ac) lower bound given by the number of initial tokens.
(ad) lower bound given by the number of initial tokens.
(ae) lower bound given by the number of initial tokens.
(af) lower bound given by the number of initial tokens.
(ag) lower bound given by the number of initial tokens.
(ah) lower bound given by the number of initial tokens.
(ai) lower bound given by the number of initial tokens.
(aj) lower bound given by the number of initial tokens.
(ak) lower bound given by the number of initial tokens.
(al) lower bound given by the number of initial tokens.
(am) lower bound given by the number of initial tokens.
(an) lower bound given by the number of initial tokens.