Hugues Evrard and Frédéric Lang
{hugues.evrard,frederic.lang}@inria.fr

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

In distributed systems, synchronizations are usually between a pair of processes, but it is also convenient to synchronize more than two processes. Also, most process calculi (with the notable exception of CCS) allow rendezvous on more than two process. Thus, dedicated protocols are needed to provide such multiway synchronizations. Such protocols are often implemented above lower-level layers that only enable binary communications by asynchronous message passing.

The present P/T net was derived from the formal specification in LNT (*LOTOS New Technology*) of such a distributed synchronization protocol. It is worth noticing that all communications are binary, while the protocol itself implements multiway synchronizations. LNT combines functional languages (to describe data types and user-defined functions operating on typed values) and process calculi (to describe concurrent components that synchronize using rendezvous and communicate via message passing). The LNT specification is 1900-line long (including comments) and contains both manually-written and automatically-generated LNT code.

This specification was translated to LOTOS, and then to an interpreted Petri net using the CADP toolbox. Finally, the present P/T net was obtained by stripping out all dataflow-related information (variables, types, assignments, guards, etc.) from the interpreted Petri net, leading to a NUPN (*Nested-Unit Petri Net*) model translated to PNML using the CÆSAR.BDD tool.

Model-checking verification of multiway synchronization protocols revealed subtle bugs that had remained undetected so far.

References

Hugues Evrard and Frédéric Lang. *Formal Verification of Distributed Branching Multiway Synchronization Protocols*. Proceedings of the IFIP Joint International Conference on Formal Techniques for Distributed Systems (FORTE/FMOODS'13). Springer, Lecture Notes in Computer Science, vol. 7892 , April 2013. http://hal.inria.fr/hal-00818788/en

Scaling parameter

This model is not parameterized.

Size of the model

number of places:	222
number of transitions:	472
number of arcs:	1496
number of units:	28
HWB code (*height-width-bits*):	2–27–81

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]

[a] 477 arcs are not simple free choice, e.g., the arc from place 3 (which has 3 outgoing transitions) to transition 371 (which has 2 input places).
[b] transitions 380 and 371 share a common input place 3, but only the former transition has input place 11.
[c] 267 transitions are not of a state machine, e.g., transition 0.

Hugues Evrard and Frédéric Lang
{hugues.evrard,frederic.lang}@inria.fr

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)
deadlock — *there exists a reachable marking from which no transition can be fired* ..... ? (p)
reversible — *from every reachable marking, there is a transition path going back to the initial marking* ..... ?
quasi-live — *for every transition t, there exists a reachable marking in which t can fire* ..... ?
live — *for every transition t, from every reachable marking, one can reach a marking in which t can fire* ..... ?

Size of the marking graph

number of reachable markings:	5.2596E+19 (q)
number of transition firings:	2.1892E+21 (r)
max. number of tokens per place:	1 (s)
max. number of tokens per marking:	27

(d) 194 places are not of a marked graph, e.g., place 0.
(e) stated by CÆSAR.BDD version 1.5.
(f) from place 1 one cannot reach place 0.
(g) place 0 is a source place.
(h) stated by CÆSAR.BDD version 1.5.
(i) stated by CÆSAR.BDD version 1.5.
(j) stated by CÆSAR.BDD version 1.5.
(k) 6 transitions are not loop free, e.g., transition 279.
(l) 7 transitions are not conservative, e.g., transition 0.
(m) transition 0 is not subconservative.
(n) the definition of Nested-Unit Petri Nets (NUPN) is available from http://mcc.lip6.fr/nupn.php
(o) safe by construction – stated by the CÆSAR compiler.
(p) found to be false at MCC'2014 by GreatSPN.
(q) computed at MCC'2014 by Marcie, PNMC, and PNXDD.
(r) computed at MCC'2014 by Marcie.
(s) stated by the CÆSAR compiler; confirmed at MCC'2014 by GreatSPN and Marcie.