

This form is a summary description of the model entitled “DLC shifumi” 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 DLC compiler [2,3,4] has been developed to automatically generate a distributed implementation of a concurrent system described using the LNT language. The implementation generated by DLC consists of processes (in the C language) executing in parallel and connected with POSIX sockets. These processes synchronize together and communicate using a distributed protocol for value-passing multiway rendezvous. Besides generating a distributed implementation, the DLC compiler can also produce an LNT model of this implementation by combining the source LNT description of the system with the protocol itself [1]. This implementation model can then be used to check the correctness of the distributed implementation using the [CADP](#) toolbox.

This collection of P/T nets was obtained by using DLC to generate implementation models to various instances of the “rock-paper-scissor” game (also known as *shifumi*). This game can be extended to $N \geq 2$ players who interact using 2 among N rendezvous. Each generated LNT model was translated automatically to LOTOS, and then to an interpreted Petri net using the [CADP](#) toolbox. Finally, a P/T net was obtained by stripping out all data-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.

Each instance of the model is parameterized by the number N of players.

Each instance is also parameterized by its version V , which specifies how the NUPN has been produced from the LOTOS specification. V is either equal to “a” if the NUPN has been generated *after* applying all the structural and data-flow optimizations of the [CÆSAR](#) compiler for LOTOS, or to “b” if the NUPN has been generated *before* these optimizations.

References

- [1] 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’2013), Florence, Italy. LNCS 7892, pages 146-160, Springer, 2013. Available from <https://hal.inria.fr/hal-00818788>.
- [2] Hugues Evrard and Frédéric Lang. *Automatic Distributed Code Generation from Formal Models of Asynchronous Concurrent Processes*. Proceedings of the 23rd Euromicro International Conference on Parallel, Distributed and Network-based Processing, Special Session on Formal Approaches to Parallel and Distributed Systems (PDP/4PAD’2015), Turku, Finland. IEEE, 2015. Available from <https://hal.inria.fr/hal-01086522>.
- [3] Hugues Evrard. *DLC: Compiling a Concurrent System Formal Specification to a Distributed Implementation*. Proceedings of the 22nd International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS’2016), Eindhoven, Netherlands. Springer, 2016.
- [4] <http://hevrard.org/DLC>

Scaling parameter

Parameter name	Parameter description	Chosen parameter values
(N, V)	N is the number of players and V is the version defined above	$\{2, 3, 4, 5, 6\} \times \{a, b\}$

Size of the model

Parameter	Number of places	Number of transitions	Number of arcs	Number of units	HWB code
$N = 2, V = a$	188	888	3166	80	2-79-116
$N = 2, V = b$	2483	3183	7756	157	63-79-445
$N = 3, V = a$	544	3097	11545	283	2-282-370
$N = 3, V = b$	7058	9611	24573	563	241-282-1535
$N = 4, V = a$	1178	7504	28610	692	2-691-854
$N = 4, V = b$	15015	21341	56284	1381	613-691-3691
$N = 5, V = a$	2162	14865	57457	1379	2-1378-1640
$N = 5, V = b$	27146	39849	107425	2755	1251-1378-7273
$N = 6, V = a$	3568	25936	101182	2416	2-2415-2800
$N = 6, V = b$	44243	66611	182532	4829	2227-2415-12641

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)
- 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 ?

(a) stated by [CÆSAR.BDD](#) version 2.6 on all 10 instances (5 values of $N \times 2$ values of V).

(b) stated by [CÆSAR.BDD](#) version 2.6 on all 10 instances (5 values of $N \times 2$ values of V).

(c) stated by [CÆSAR.BDD](#) version 2.6 on all 10 instances (5 values of $N \times 2$ values of V).

(d) stated by [CÆSAR.BDD](#) version 2.6 on all 10 instances (5 values of $N \times 2$ values of V).

(e) stated by [CÆSAR.BDD](#) version 2.6 on all 10 instances (5 values of $N \times 2$ values of V).

(f) from place 1 one cannot reach place 0.

(g) place 0 is a source place.

(h) stated by [CÆSAR.BDD](#) version 2.6 on all 10 instances (5 values of $N \times 2$ values of V).

(i) stated by [CÆSAR.BDD](#) version 2.6 on all 10 instances (5 values of $N \times 2$ values of V).

(j) stated by [CÆSAR.BDD](#) version 2.6 on all 10 instances (5 values of $N \times 2$ values of V).

(k) stated by [CÆSAR.BDD](#) version 2.6 to be true on 5 instance(s) out of 10, and false on the remaining 5 instance(s).

(l) stated by [CÆSAR.BDD](#) version 2.6 on all 10 instances (5 values of $N \times 2$ values of V).

(m) stated by [CÆSAR.BDD](#) version 2.6 on all 10 instances (5 values of $N \times 2$ values of V).

(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) stated by [CÆSAR.BDD](#) version 2.6 to be false on 4 instance(s) out of 10, and unknown on the remaining 6 instance(s).

quasi-live — for every transition t , there exists a reachable marking in which t can fire? ^(q)
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 reach-able markings	Number of tran-sition firings	Max. number of tokens per place	Max. number of tokens per marking
$N = 2, V = a$	4.74756e+14 ^(r)	?	1	79
$N = 2, V = b$?	?	1 ^(s)	$\in [2, 79]$ ^(t)
$N = 3, V = a$	9.09544e+35 ^(u)	?	1	282
$N = 3, V = b$?	?	1 ^(v)	$\in [2, 282]$ ^(w)
$N = 4, V = a$	2.41187e+67 ^(x)	?	1	691
$N = 4, V = b$?	?	1 ^(y)	$\in [2, 691]$ ^(z)
$N = 5, V = a$?	?	1	1378
$N = 5, V = b$?	?	1 ^(aa)	$\in [2, 1378]$ ^(ab)
$N = 6, V = a$?	?	1 ^(ac)	2415
$N = 6, V = b$?	?	1 ^(ad)	$\in [2, 2415]$ ^(ae)

^(q) stated by CÆSAR.BDD version 2.6 to be true on 4 instance(s) out of 10, and unknown on the remaining 6 instance(s).

^(r) stated by CÆSAR.BDD version 2.6.

^(s) stated by the CÆSAR compiler.

^(t) lower and upper bounds given by the number of initial tokens and the number of leaf units.

^(u) stated by CÆSAR.BDD version 2.6.

^(v) stated by the CÆSAR compiler.

^(w) lower and upper bounds given by the number of initial tokens and the number of leaf units.

^(x) stated by CÆSAR.BDD version 2.6.

^(y) stated by the CÆSAR compiler.

^(z) lower and upper bounds given by the number of initial tokens and the number of leaf units.

^(aa) stated by the CÆSAR compiler.

^(ab) lower and upper bounds given by the number of initial tokens and the number of leaf units.

^(ac) stated by the CÆSAR compiler.

^(ad) stated by the CÆSAR compiler.

^(ae) lower and upper bounds given by the number of initial tokens and the number of leaf units.