Hugues Evrard
h.evrard@imperial.ac.uk

This form is a summary description of the model entitled "DLCround" 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 [2,3,4,5] 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 "musical chairs" game (the module is named "round" for the sake of brievety). This game consists of several rounds where N players compete to sit on a chair each among $N-1$ chairs; at each round the player which did not manage to sit is disqualified and a chair is removed. 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 at the game start.

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] Hugues Evrard and Frédéric Lang. *Automatic Distributed Code Generation from Formal Models of Asynchronous Processes Interacting by Multiway Rendezvous.* Journal of Logical and Algebraic Methods in Programming, vol. 88, pages 121–153, Elsevier, 2017.

[5] 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	$\{3,4,5,6,7,8,9,10,11,12,13\} \times \{a, b\}$

Hugues Evrard
h.evrard@imperial.ac.uk

Size of the model

Parameter	Number of places	Number of transitions	Number of arcs	Number of units	HWB code
$N = 03, V = a$	113	617	2269	53	2–52–73
$N = 03, V = b$	1383	1887	4809	103	45–52–283
$N = 04, V = a$	139	823	3074	70	2–69–93
$N = 04, V = b$	1680	2364	6156	137	61–69–371
$N = 05, V = a$	167	1055	3985	89	2–88–115
$N = 05, V = b$	1999	2887	7649	175	79–88–469
$N = 06, V = a$	197	1313	5002	110	2–109–139
$N = 06, V = b$	2340	3456	9288	217	99–109–577
$N = 07, V = a$	229	1597	6125	133	2–132–165
$N = 07, V = b$	2703	4071	11073	263	121–132–695
$N = 08, V = a$	263	1907	7354	158	2–157–193
$N = 08, V = b$	3088	4732	13004	313	145–157–823
$N = 09, V = a$	299	2243	8689	185	2–184–223
$N = 09, V = b$	3495	5439	15081	367	171–184–961
$N = 10, V = a$	337	2605	10130	214	2–213–255
$N = 10, V = b$	3924	6192	17304	425	199–213–1109
$N = 11, V = a$	377	2993	11677	245	2–244–289
$N = 11, V = b$	4375	6991	19673	487	229–244–1267
$N = 12, V = a$	419	3407	13330	278	2–277–325
$N = 12, V = b$	4848	7836	22188	553	261–277–1435
$N = 13, V = a$	463	3847	15089	313	2–312–363
$N = 13, V = b$	5343	8727	24849	623	295–312–1613

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]

[a] stated by CÆSAR.BDD version 2.7 on all 22 instances (11 values of N × 2 values of V).
[b] stated by CÆSAR.BDD version 2.7 on all 22 instances (11 values of N × 2 values of V).
[c] stated by CÆSAR.BDD version 2.7 on all 22 instances (11 values of N × 2 values of V).
[d] stated by CÆSAR.BDD version 2.7 on all 22 instances (11 values of N × 2 values of V).
[e] stated by CÆSAR.BDD version 2.7 on all 22 instances (11 values of N × 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 2.7 on all 22 instances (11 values of N × 2 values of V).
[i] stated by CÆSAR.BDD version 2.7 on all 22 instances (11 values of N × 2 values of V).
[j] stated by CÆSAR.BDD version 2.7 on all 22 instances (11 values of N × 2 values of V).
[k] stated by CÆSAR.BDD version 2.7 to be true on 11 instance(s) out of 22, and false on the remaining 11 instance(s).
[l] stated by CÆSAR.BDD version 2.7 on all 22 instances (11 values of N × 2 values of V).
[m] stated by CÆSAR.BDD version 2.7 on all 22 instances (11 values of N × 2 values of V).

Hugues Evrard
h.evrard@imperial.ac.uk

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* .... ?
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
$N = 03, V = a$	2.401e+07 [s]	?	1	52
$N = 03, V = b$	$\geq$ 2.85271e+46 [t]	?	1 [u]	52
$N = 04, V = a$	2.401e+08 [v]	?	1	69
$N = 04, V = b$	$\geq$ 3.60097e+60 [w]	?	1 [x]	69
$N = 05, V = a$	2.401e+09 [y]	?	1	88
$N = 05, V = b$	?	?	1 [z]	88
$N = 06, V = a$	2.401e+10 [aa]	?	1	109
$N = 06, V = b$	?	?	1 [ab]	109
$N = 07, V = a$	2.401e+11 [ac]	?	1	132
$N = 07, V = b$	?	?	1 [ad]	132
$N = 08, V = a$	2.401e+12 [ae]	?	1	157
$N = 08, V = b$	?	?	1 [af]	157
$N = 09, V = a$	2.401e+13 [ag]	?	1	184
$N = 09, V = b$	?	?	1 [ah]	184
$N = 10, V = a$	2.401e+14 [ai]	?	1	213
$N = 10, V = b$	?	?	1 [aj]	213
$N = 11, V = a$	2.401e+15 [ak]	?	1	244
$N = 11, V = b$	?	?	1 [al]	244
$N = 12, V = a$	2.401e+16 [am]	?	1	277
$N = 12, V = b$	?	?	1 [an]	277
$N = 13, V = a$	2.401e+17 [ao]	?	1	312
$N = 13, V = b$	?	?	1 [ap]	312

[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 3.3 to be false on 11 instance(s) out of 22, and unknown on the remaining 11 instance(s).
[q] stated by CÆSAR.BDD version 2.7 to be false on 11 instance(s) out of 22, and unknown on the remaining 11 instance(s).
[r] stated by CÆSAR.BDD version 2.7 to be false on 11 instance(s) out of 22, and unknown on the remaining 11 instance(s).
[s] stated by CÆSAR.BDD version 2.7.
[t] stated by CÆSAR.BDD version 2.7.
[u] stated by the CÆSAR compiler.
[v] stated by CÆSAR.BDD version 2.7.
[w] stated by CÆSAR.BDD version 3.3.
[x] stated by the CÆSAR compiler.
[y] stated by CÆSAR.BDD version 2.7.
[z] stated by the CÆSAR compiler.
[aa] stated by CÆSAR.BDD version 2.7.
[ab] stated by the CÆSAR compiler.
[ac] stated by CÆSAR.BDD version 2.7.
[ad] stated by the CÆSAR compiler.
[ae] stated by CÆSAR.BDD version 2.7.

Hugues Evrard
h.evrard@imperial.ac.uk

[af] stated by the CÆSAR compiler.
[ag] stated by CÆSAR.BDD version 2.7.
[ah] stated by the CÆSAR compiler.
[ai] stated by CÆSAR.BDD version 2.7.
[aj] stated by the CÆSAR compiler.
[ak] stated by CÆSAR.BDD version 2.7.
[al] stated by the CÆSAR compiler.
[am] stated by CÆSAR.BDD version 2.7.
[an] stated by the CÆSAR compiler.
[ao] stated by CÆSAR.BDD version 2.7.
[ap] stated by the CÆSAR compiler.