Rosa Mendas, Fabrice Kordon
Fabrice.Kordon@lip6.fr

This form is a summary description of the model entitled "CANInsertWithFailure" 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 Petri net models the construction of a CAN (Content-Addressable Network) [1]. Nodes successively request insertion into the DHT witch is empty at the beginning. Each node execute the same code : either it is the first one to get into the CAN, or the CAN already exists and it must request for insertion (that can be accepted or rejected by the node receiving the request).

The following hypotheses are considered:

- first insertion is always in an empty CAN,
- Communications between processes are asynchronous,
- Nodes addresses are known so new nodes may communicate directly,
- Some failures are considered.

The model was elaborated during a student project at the bachelor level.

References

1. S. Ratnasamy, P. Francis, M. Handley, R. M. Karp, and S. Shenker. A scalable content-addressable network. In R. L. Cruz and G. Varghese, editors, Proceedings of the ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 27-31, 2001, San Diego, CA, USA, pages 161–172. ACM, 2001.

Scaling parameter

Parameter name	Parameter description	Chosen parameter values
N	N is the maximum number of nodes to be inserted	5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100

Size of the model

Parameter	Number of places	Number of transitions	Number of arcs
5	114	180	500
10	324	660	1 800
20	1 044	2 520	6 800
30	2 164	580	15 000
40	3 684	9 840	26 400
50	5 604	13 300	41 000
60	7 924	21 960	58 800
70	10 644	29 820	79 800
80	13 764	38 880	104 000
90	17 284	49 140	121 400
100	21 204	60 600	162 000

Rosa Mendas, Fabrice Kordon
Fabrice.Kordon@lip6.fr

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]
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)* ..... yes [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* ..... ? [o]
dead place(s) — *one or more places have no token in any reachable marking* ..... ?
dead transition(s) — *one or more transitions cannot fire from any reachable marking* ..... ?
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* ..... ?

[a] stated by CÆSAR.BDD version 3.7 on all 11 instances (5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).
[b] transitions "Node0NextInCAN" and "Node1NextInCAN" share a common input place "InsertedCounter", but only the former transition has input place "Node0ReadyToInsert".
[c] stated by CÆSAR.BDD version 3.7 on all 11 instances (5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).
[d] stated by CÆSAR.BDD version 3.7 on all 11 instances (5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).
[e] stated by CÆSAR.BDD version 3.7 on all 11 instances (5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).
[f] from place "TankCounter" one cannot reach place "TankCounter".
[g] stated by CÆSAR.BDD version 3.7 on all 11 instances (5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).
[h] place "TotalFailure" is a sink place.
[i] stated by CÆSAR.BDD version 3.7 on all 11 instances (5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).
[j] stated by CÆSAR.BDD version 3.7 on all 11 instances (5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).
[k] stated by CÆSAR.BDD version 3.7 on all 11 instances (5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).
[l] stated by CÆSAR.BDD version 3.7 on all 11 instances (5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).
[m] stated by CÆSAR.BDD version 3.7 on all 11 instances (5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).
[n] the definition of Nested-Unit Petri Nets (NUPN) is available from http://mcc.lip6.fr/nupn.php
[o] stated by CÆSAR.BDD version 3.7 to be false on 2 instance(s) out of 11, and unknown on the remaining 9 instance(s).

Rosa Mendas, Fabrice Kordon
Fabrice.Kordon@lip6.fr

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
5	$\geq$ 200143 [p]	?	?	$\geq$ 11
10	$\geq$ 3.76522e+10 [q]	?	?	$\geq$ 21
20	?	?	?	$\geq$ 41
30	?	?	?	$\geq$ 61
40	?	?	?	$\geq$ 81
50	?	?	?	$\geq$ 101
60	?	?	?	$\geq$ 62 [r]
70	?	?	?	$\geq$ 72 [s]
80	?	?	?	$\geq$ 82 [t]
90	?	?	?	$\geq$ 92 [u]
100	?	?	?	$\geq$ 102 [v]

[p] stated by CÆSAR.BDD version 3.7.
[q] stated by CÆSAR.BDD version 3.7.
[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.