Niels Lohmann
niels.lohmann@uni-rostock.de

This form is a summary description of the model entitled "Echo Algorithm" 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

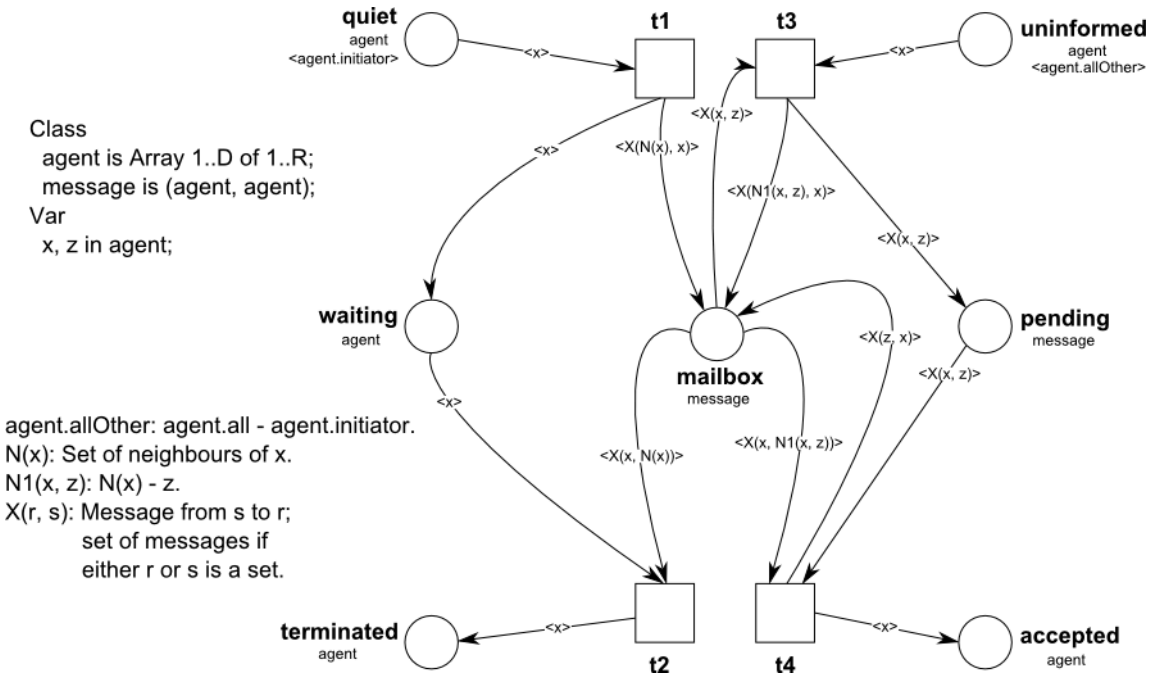
This file specifies the Echo Algorithm (see [Reisig98]) for grid like networks. Echo is a protocol for propagation of information with feedback in a network. The algorithm operates in an incomplete, but connected bidirectional network of agents. A distinguished agent (initiator), starts the distribution of a message by sending it to all its neighbors. On receiving some first message, every other agent forwards the message to all its neighbors, except the one it received its first message from. Then it awaits messages from all recipients of its forwards (regardless whether these messages had been intended as forwards or acknowledgments) and replies to the agent where it received its first message from. As soon as the initiator receives a message from all its neighbors, the protocol terminates.

In this example, agents are arranged in a hypercube. The network can be scaled in two values: the number of *dimensions* and the number of *agents per dimensions*. For instance, a network with two dimensions and three agents per row would consist of $3^2 = 9$ agents while a network with three dimensions and 4 agents per row would consist of $4^3 = 64$ agents.

Regardless of the chosen values for these dimensions, we always connect agents that are immediate neighbors in one of the dimensions of the hypercube. We place the initiator into the center of the cube which means that the number of agents per row should be an odd number.

Unfolded versions of the Echo Algorithm are also provided for a variety of scaling parameters. These nets are given in LoLA low-level format and PNML.

The model is sketched in the figure. The two sorts D and R model the scaling factors dimensions and agents per row, respectively. Messages are modeled as pairs (receiver, sender).



References

Wolfgang Reisig. *Elements of Distributed Algorithms. Modeling and Analysis with Petri Nets.*, Springer, 1998.

The original algorithm has been modeled as an algebraic Petri net in LoLA high-level format, see http://service-technology.org/files/lola/lola.pdf

Niels Lohmann
niels.lohmann@uni-rostock.de

Scaling parameter

Parameter name	Parameter description	Chosen parameter values
(dimensions d, agents per row a)	see description	(2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), (5,3)

Size of the model

Parameter	Number of places	Number of transitions	Number of arcs
$(d = 2, a = 9)$	735	570	3220
$(d = 2, a = 11)$	1119	874	4996
$(d = 2, a = 15)$	2127	1674	9700
$(d = 2, a = 19)$	3455	2730	15940
$(d = 3, a = 3)$	265	206	1252
$(d = 3, a = 5)$	1445	1190	8260
$(d = 3, a = 7)$	4209	3518	25540
$(d = 4, a = 3)$	1019	850	6340
$(d = 5, a = 3)$	3717	3222	28404

Structural properties

free choice — *all (different) transitions with a shared input place have no other input place* ..... ✗ [a]
state machine — *every transition has exactly one input place and exactly one output place* ..... ✗ [b]
marked graph — *every place has exactly one input transition and exactly one output transition* ..... ✗ [c]
connected — *there is a undirected path between every two nodes (places or transitions)* ..... ✓ [d]
strongly connected — *there is a directed path between every two nodes (places or transitions)* ..... ✗ [e]
source place(s) — *one or more places have no input transitions* ..... ✓ [f]
sink place(s) — *one or more places have no output transitions* ..... ✓ [g]
source transition(s) — *one or more transitions have no input places* ..... ✗ [h]
sink transitions(s) — *one or more transitions have no output places* ..... ✗ [i]
loop-free — *no transition has an input place that is also an output place* ..... ✓ [j]
conservative — *for each transition, the number of input arcs equals the number of output arcs* ..... ✗ [k]
subconservative — *for each transition, the number of input arcs equals or exceeds the number of output arcs* ..... ✗ [l]

Behavioural properties

safe — *in every reachable marking, there is no more than one token on a place* ..... ✓
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* ..... ✗
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* ..... ?

[a] stated by CÆSAR.BDD version 1.7 on all 9 instances ((2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), and (5,3)).
[b] stated by CÆSAR.BDD version 1.7 on all 9 instances ((2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), and (5,3)).
[c] stated by CÆSAR.BDD version 1.7 on all 9 instances ((2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), and (5,3)).
[d] stated by CÆSAR.BDD version 1.7 on all 9 instances ((2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), and (5,3)).
[e] stated by CÆSAR.BDD version 1.7 on all 9 instances ((2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), and (5,3)).
[f] stated by CÆSAR.BDD version 1.7 on all 9 instances ((2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), and (5,3)).
[g] stated by CÆSAR.BDD version 1.7 on all 9 instances ((2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), and (5,3)).
[h] stated by CÆSAR.BDD version 1.7 on all 9 instances ((2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), and (5,3)).
[i] stated by CÆSAR.BDD version 1.7 on all 9 instances ((2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), and (5,3)).
[j] stated by CÆSAR.BDD version 1.7 on all 9 instances ((2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), and (5,3)).
[k] stated by CÆSAR.BDD version 1.7 on all 9 instances ((2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), and (5,3)).
[l] stated by CÆSAR.BDD version 1.7 on all 9 instances ((2,9), (2,11), (2,15), (2,19), (3,3), (3,5), (3,7), (4,3), and (5,3)).

Niels Lohmann
niels.lohmann@uni-rostock.de

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
$(d = 2, a = 9)$	?	?	?	≥ 81 [m]
$(d = 2, a = 11)$	?	?	?	≥ 121
$(d = 2, a = 15)$	?	?	?	≥ 225
$(d = 2, a = 19)$	?	?	?	≥ 361
$(d = 3, a = 3)$	≥ 26465 [n]	?	?	≥ 27
$(d = 3, a = 5)$	?	?	?	≥ 125
$(d = 3, a = 7)$	?	?	?	≥ 343
$(d = 4, a = 3)$	≥ 4 [o]	?	?	≥ 81
$(d = 5, a = 3)$	?	?	?	≥ 243

Other properties

The intuitive description of the Echo Algorithm can be modeled as a CTL formula

(**AF** "initiator terminated") $\wedge$ (**A** $\neg$"initiator terminated" **U** "all other sites accepted")

This formula is given for the unfolded low-level models.

[m] lower bound given by the number of initial tokens.
[n] stated by CÆSAR.BDD version 2.0.
[o] stated by CÆSAR.BDD version 2.0.