This form is a summary description of the model entitled “Solitaire” 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

Solitaire is a popular board game requiring non-obvious solution strategies; see [wiki] for the rules of the game. The objective of the Petri nets is to generate one/some/all strategies (paths) to reach a solution, i.e., a state where just one stone is left. The auxiliary place counter gives the current number of stones on the board; added to simplify the specification of the target state (any state with $counter = 1$). Solitaire is played on different boards; we give Petri nets for the most popular ones: square board (0), English board (1), French board (2), each in two versions: with/out counter [H05]. The existence of a solution may depend on the initially empty field; all initial markings have been chosen to enable a solution. Encoding this game as coloured Petri net would permit the generation of arbitrary boards of scalable size.

References


Scaling parameter

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Parameter description</th>
<th>Chosen parameter values</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>shape and size of the board</td>
<td>5 × 5 square board (0), 7 × 7 English board (1), 7 × 7 French board (3)</td>
</tr>
</tbody>
</table>
Size of the model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Number of places</th>
<th>Number of transitions</th>
<th>Number of arcs</th>
</tr>
</thead>
<tbody>
<tr>
<td>B = 0</td>
<td>50</td>
<td>84</td>
<td>456</td>
</tr>
<tr>
<td>B = 0, with counter</td>
<td>51</td>
<td>84</td>
<td>540</td>
</tr>
<tr>
<td>B = 1</td>
<td>66</td>
<td>76</td>
<td>456</td>
</tr>
<tr>
<td>B = 1, with counter</td>
<td>67</td>
<td>76</td>
<td>532</td>
</tr>
<tr>
<td>B = 2</td>
<td>74</td>
<td>92</td>
<td>552</td>
</tr>
<tr>
<td>B = 2, with counter</td>
<td>75</td>
<td>92</td>
<td>644</td>
</tr>
</tbody>
</table>

Structural properties

- ordinary — all arcs have multiplicity one
- simple free choice — all transitions sharing a common input place have no other input place
- extended free choice — all transitions sharing a common input place have the same input places
- state machine — every transition has exactly one input place and exactly one output place
- marked graph — every place has exactly one input transition and exactly one output transition
- connected — there is an undirected path between every two nodes (places or transitions)
- strongly connected — there is a directed path between every two nodes (places or transitions)
- source place(s) — one or more places have no input transitions
- sink place(s) — one or more places have no output transitions
- source transition(s) — one or more transitions have no input places
- sink transitions(s) — one or more transitions have no output places
- loop-free — no transition has an input place that is also an output place
- conservative — for each transition, the number of input arcs equals the number of output arcs
- subconservative — for each transition, the number of input arcs equals or exceeds the number of output arcs
- nested units — places are structured into hierarchically nested sequential units

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 2.0 on all 6 instances \( B \in \{0, 1, 2\} \), with and without counter.
(b) stated by CÆSAR.BDD version 2.6 on all 6 instances \( B \in \{0, 1, 2\} \), with and without counter.
(c) stated by CÆSAR.BDD version 2.0 on all 6 instances \( B \in \{0, 1, 2\} \), with and without counter.
(d) stated by CÆSAR.BDD version 2.0 on all 6 instances \( B \in \{0, 1, 2\} \), with and without counter.
(e) stated by CÆSAR.BDD version 2.0 on all 6 instances \( B \in \{0, 1, 2\} \), with and without counter.
(f) stated by CÆSAR.BDD version 2.0 to be false on all 3 instances with counters, and true on all 3 instances without counters.
(g) stated by CÆSAR.BDD version 2.0 to be true on all 3 instances with counters, and false on all 3 instances without counters.
(h) stated by CÆSAR.BDD version 2.0 on all 6 instances \( B \in \{0, 1, 2\} \), with and without counter.
(i) stated by CÆSAR.BDD version 2.0 on all 6 instances \( B \in \{0, 1, 2\} \), with and without counter.
(j) stated by CÆSAR.BDD version 2.0 on all 6 instances \( B \in \{0, 1, 2\} \), with and without counter.
(k) stated by CÆSAR.BDD version 2.0 on all 6 instances \( B \in \{0, 1, 2\} \), with and without counter.
(l) stated by CÆSAR.BDD version 2.0 to be false on all 3 instances with counters, and true on all 3 instances without counters.
(m) stated by CÆSAR.BDD version 2.0 to be false on all 3 instances with counters, and unknown on the remaining 3 instance(s).
(n) the definition of Nested-Unit Petri Nets (NUPN) is available from http://mcc.lip6.fr/nupn.php
(o) special deadlocks (dead states) correspond to the solutions we are looking for; confirmed at MCC’2014 by Lola and Tapalal on all 6 instances.
Size of the marking graphs

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Number of reachable markings</th>
<th>Number of transition firings</th>
<th>Max. number of tokens per place</th>
<th>Max. number of tokens per marking</th>
</tr>
</thead>
<tbody>
<tr>
<td>( B = 0 )</td>
<td>( 1.6098 \times 10^7 )</td>
<td>( 2.1396 \times 10^2 )</td>
<td>1 ( (t) )</td>
<td>25 ( (t) )</td>
</tr>
<tr>
<td>( B = 0, ) with counter</td>
<td>?</td>
<td>?</td>
<td>24</td>
<td>49 ( (t) )</td>
</tr>
<tr>
<td>( B = 1 )</td>
<td>?</td>
<td>?</td>
<td>1</td>
<td>33 ( (v) )</td>
</tr>
<tr>
<td>( B = 1, ) with counter</td>
<td>?</td>
<td>?</td>
<td>32</td>
<td>65 ( (w) )</td>
</tr>
<tr>
<td>( B = 2 )</td>
<td>?</td>
<td>?</td>
<td>1</td>
<td>37 ( (x) )</td>
</tr>
<tr>
<td>( B = 2, ) with counter</td>
<td>?</td>
<td>?</td>
<td>36</td>
<td>73 ( (y) )</td>
</tr>
</tbody>
</table>

Other properties

Deadlocks (dead states) which correspond to a solution can be identified by: sum over all places \( T_{i,j} = 1 \), or counter=0. All places are covered by 1-P-invariants, except the counter place. All nets enjoy some symmetries.

---

\((q)\) computed at MCC’2014 by Marcie, PNMC, and PNXDD; exact value: 16,098,428.
\((t)\) computed at MCC’2014 by Marcie; exact value: 213,958,152.
\((s)\) computed at MCC’2014 by Marcie and PNMC.
\((u)\) number of initial tokens, because the net is sub-conservative.
\((v)\) number of initial tokens, because the net is sub-conservative.
\((w)\) number of initial tokens, because the net is sub-conservative.
\((x)\) number of initial tokens, because the net is sub-conservative.
\((y)\) number of initial tokens, because the net is sub-conservative.