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.

In March 2020, Pierre Bouvier and Hubert Garavel provided a decomposition of one instance of this model into a network of communicating automata. This network is expressed as a Nested-Unit Petri Net (NUPN) that can be found in the “toolspecific” section of the corresponding PNML file.

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>
<th>Number of units</th>
<th>HWB code</th>
</tr>
</thead>
<tbody>
<tr>
<td>B = 0</td>
<td>50</td>
<td>84</td>
<td>456</td>
<td>26</td>
<td>1–25–25</td>
</tr>
<tr>
<td>B = 0, with counter</td>
<td>51</td>
<td>84</td>
<td>540</td>
<td>–</td>
<td>– – 51</td>
</tr>
<tr>
<td>B = 1</td>
<td>66</td>
<td>76</td>
<td>456</td>
<td>–</td>
<td>– – 66</td>
</tr>
<tr>
<td>B = 1, with counter</td>
<td>67</td>
<td>76</td>
<td>532</td>
<td>–</td>
<td>– – 67</td>
</tr>
<tr>
<td>B = 2</td>
<td>74</td>
<td>92</td>
<td>552</td>
<td>–</td>
<td>– – 74</td>
</tr>
<tr>
<td>B = 2, with counter</td>
<td>75</td>
<td>92</td>
<td>644</td>
<td>–</td>
<td>– – 75</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    (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) .............................................. ✓

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 transition(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(n) .............................................. ✓

Behavioural properties

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

dead place(s) — one or more places have no token in any reachable marking .............................................. ✓

(a) stated by CÆSAR.BDD version 2.0 on all 6 instances (B ∈ {0, 1, 2}, with and without counter).
(b) stated by CÆSAR.BDD version 2.6 on all 6 instances (B ∈ {0, 1, 2}, with and without counter).
(c) stated by CÆSAR.BDD version 2.0 on all 6 instances (B ∈ {0, 1, 2}, with and without counter).
(d) stated by CÆSAR.BDD version 2.0 on all 6 instances (B ∈ {0, 1, 2}, with and without counter).
(e) stated by CÆSAR.BDD version 2.0 on all 6 instances (B ∈ {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 ∈ {0, 1, 2}, with and without counter).
(i) stated by CÆSAR.BDD version 2.0 on all 6 instances (B ∈ {0, 1, 2}, with and without counter).
(j) stated by CÆSAR.BDD version 2.0 on all 6 instances (B ∈ {0, 1, 2}, with and without counter).
(k) stated by CÆSAR.BDD version 2.0 to be false on all 3 instances with counters, and true on all 3 instances without counters.
(l) stated by CÆSAR.BDD version 2.0 to be true on all 3 instances with counters, and false on all 3 instances without counters.
(m) the definition of Nested-Unit Petri Nets (NUPN) is available from http://mcc.lip6.fr/nupn.php
(n) the definition of Nested-Unit Petri Nets (NUPN) is available from http://mcc.lip6.fr/nupn.php
(o) stated by CÆSAR.BDD version 3.3 to be true on 1 instance(s) out of 6, and false on the remaining 5 instance(s).
(p) the nets corresponding to instances without counters are safe because they are covered with P-invariants having a single token in the initial place — found by CÆSAR.BDD version 3.3 to be true on 1 instance(s) out of 6, false on 3 instance(s), and unknown on the remaining 2 instance(s).
(q) stated by CÆSAR.BDD version 3.3 to be false on 3 instance(s) out of 6, and unknown on the remaining 3 instance(s).
dead transition(s) — one or more transitions cannot fire from any reachable marking ................. X
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 ................. X
live — for every transition t, from every reachable marking, one can reach a marking in which t can fire ................. X

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 × 10^{17} (s)</td>
<td>2.1396 × 10^{8} (t)</td>
<td>1 (d)</td>
<td>25 (v)</td>
</tr>
<tr>
<td>B = 0, with counter</td>
<td>?</td>
<td>?</td>
<td>24</td>
<td>49 (w)</td>
</tr>
<tr>
<td>B = 1</td>
<td>?</td>
<td>?</td>
<td>1</td>
<td>33 (x)</td>
</tr>
<tr>
<td>B = 1, with counter</td>
<td>?</td>
<td>?</td>
<td>32</td>
<td>65 (y)</td>
</tr>
<tr>
<td>B = 2</td>
<td>?</td>
<td>?</td>
<td>1</td>
<td>37 (z)</td>
</tr>
<tr>
<td>B = 2, with counter</td>
<td>?</td>
<td>?</td>
<td>36</td>
<td>73 (aa)</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.

---

(s) special deadlocks (dead states) correspond to the solutions we are looking for; confirmed at MCC’2014 by Lola and Tapaal on all 6 instances.
(t) computed at MCC’2014 by Marcie; exact value: 213,958,152.
(u) computed at MCC’2014 by Marcie and PNMC.
(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.
(z) number of initial tokens, because the net is sub-conservative.
(aa) number of initial tokens, because the net is sub-conservative.