Description

We consider a system where a set of machines are places in a ring, numbered 0 to \(N\). Each machine \(i\) only knows its own state and the state of its left neighbor, i.e., machine \((i - 1)\%(N + 1)\). Machine number 0 plays a special role, and it is called "the bottom machine". The state of each machine is an integer number in \([0, N]\). We will note \(S(i)\) the state of machine \(i\).

Each machine may determine if it has a so-called "privilege" based on its state and the one of its left neighbor. A privilege is in this context the right to perform an operation. After performing its operation, the machine updates its own state, and may lose its privilege.

The objective of the protocol is to reach a stable state for the system. In a stable state:

- there is a unique machine with a privilege,
- any subsequent state is also a stable state.

Furthermore, the protocol ensures non-starvation: at any time, any machine is sure to have a privilege after a finite number of steps.

The detail of the protocol is different for the bottom machine, and the other machines:

- The bottom machine has the privilege if its left neighbor state is equal to its own state. In this case, the bottom machine updates its status by incrementing it: if \(S(N) = S(0)\) then \(S(0) := (S(0) + 1)\%N\)
- Any other machine \(i > 0\) has the privilege if its left neighbor state is different from its own state. In this case, machine \(i\) updates its status by setting it to the value of the left machine: if \(S(i - 1) = S(i)\) then \(S(i) := S(i - 1)\).

Graphical representation for \(N = 5\)

```class Process is 0..5;
couple is <Process, Process>;
var i, x, y in Process;

MainProcess
<0, x++1> + <5, x> <0, x> + <5, x>

State
<0, 0>, <1, 1>, <2, 2>, <3, 3>, <4, 4>, <5, 5>
<i, x> + <i-1, y> <i, y> + <i-1, y>

OtherProcess
[i <> 0 and x <> y]
```
References

http://www.cs.utexas.edu/~EWD/ewd04xx/EWD426.PDF

Scaling parameter

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Parameter description</th>
<th>Chosen parameter values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>Number of processes. It has an impact on the initial marking of the place State, as well as most of the colored functions.</td>
<td>5, 10, 15, 20, 30, 40, 50</td>
</tr>
</tbody>
</table>

Size of the colored net model

- number of places: 1
- number of transitions: 2
- number of arcs: 4

Size of the derived P/T model instances

<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>( N = 5 )</td>
<td>36</td>
<td>156</td>
<td>624</td>
</tr>
<tr>
<td>( N = 10 )</td>
<td>121</td>
<td>1111</td>
<td>4444</td>
</tr>
<tr>
<td>( N = 15 )</td>
<td>256</td>
<td>3616</td>
<td>14464</td>
</tr>
<tr>
<td>( N = 20 )</td>
<td>441</td>
<td>8421</td>
<td>33684</td>
</tr>
<tr>
<td>( N = 30 )</td>
<td>961</td>
<td>27931</td>
<td>111724</td>
</tr>
<tr>
<td>( N = 40 )</td>
<td>1681</td>
<td>65641</td>
<td>262564</td>
</tr>
<tr>
<td>( N = 50 )</td>
<td>2601</td>
<td>127551</td>
<td>510204</td>
</tr>
</tbody>
</table>

Structural properties

- ordinary — all arcs have multiplicity one
- simple free choice — all (different) transitions with a shared input place have no other input place
- 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 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

(a) stated by CÆSAR.BDD version 1.7 on all 7 instances (5, 10, 15, 20, 30, 40, and 50).
(b) stated by CÆSAR.BDD version 1.7 on all 7 instances (5, 10, 15, 20, 30, 40, and 50).
(c) stated by CÆSAR.BDD version 1.7 on all 7 instances (5, 10, 15, 20, 30, 40, and 50).
(d) stated by CÆSAR.BDD version 1.7 on all 7 instances (5, 10, 15, 20, 30, 40, and 50).
(e) stated by CÆSAR.BDD version 1.7 on all 7 instances (5, 10, 15, 20, 30, 40, and 50).
(f) stated by CÆSAR.BDD version 1.7 on all 7 instances (5, 10, 15, 20, 30, 40, and 50).
(g) stated by CÆSAR.BDD version 1.7 on all 7 instances (5, 10, 15, 20, 30, 40, and 50).
(h) stated by CÆSAR.BDD version 1.7 on all 7 instances (5, 10, 15, 20, 30, 40, and 50).
(i) stated by CÆSAR.BDD version 1.7 on all 7 instances (5, 10, 15, 20, 30, 40, and 50).
(j) stated by CÆSAR.BDD version 1.7 on all 7 instances (5, 10, 15, 20, 30, 40, and 50).
(k) stated by CÆSAR.BDD version 1.7 on all 7 instances (5, 10, 15, 20, 30, 40, and 50).
(l) stated by CÆSAR.BDD version 1.7 on all 7 instances (5, 10, 15, 20, 30, 40, and 50).
Nested units — places are structured into hierarchically nested sequential units \((m)\).

**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.

**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>(N = 5)</td>
<td>166 ((r))</td>
<td>365 ((r))</td>
<td>1 ((o))</td>
<td>6 ((o))</td>
</tr>
<tr>
<td>(N = 10)</td>
<td>58,905 ((v))</td>
<td>294,050 ((w))</td>
<td>1 ((s))</td>
<td>11 ((s))</td>
</tr>
<tr>
<td>(N = 15)</td>
<td>3.5358E+7 ((z))</td>
<td>2.6518E+8 ((aa))</td>
<td>1 ((ab))</td>
<td>16 ((ac))</td>
</tr>
<tr>
<td>(N = 20)</td>
<td>2.44663E+10 ((ad))</td>
<td>?</td>
<td>1 ((ac))</td>
<td>21 ((af))</td>
</tr>
<tr>
<td>(N = 30)</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>(\geq 31 ((ag))</td>
</tr>
<tr>
<td>(N = 40)</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>(\geq 41)</td>
</tr>
<tr>
<td>(N = 50)</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>(\geq 51)</td>
</tr>
</tbody>
</table>

\((m)\) the definition of Nested-Unit Petri Nets (NUPN) is available from [http://mcc.lip6.fr/nupn.php](http://mcc.lip6.fr/nupn.php).

\((r)\) stated by CÆSAR.BDD version 2.0 to be true on 2 instance(s) out of 7, and unknown on the remaining 5 instance(s).

\((v)\) computed at MCC’2013 by GreatSPN, ITS-Tools, Marcie, PNXDD; confirmed by CÆSAR.BDD version 1.8; confirmed by Helena, GreatSPN, Marcie, PNXDD, Stratagem, and Tapaal at MCC’2014.

\((w)\) computed by Helena and Marcie at MCC’2014.

\((z)\) computed by GreatSPN, PNMC, Marcie, and Tapaal at MCC’2014.

\((ad)\) computed by PNMC at MCC’2014.

\((ag)\) lower bound given by the number of initial tokens.