Thomas Hujsa
thomas.hujsa@uni-oldenburg.de

*This form is a summary description of the model entitled "JoinFreeModules" 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

Liveness and boundedness are well-known Petri net properties that are fundamental for many real-world applications, including embedded systems, and are hard to check. So as to alleviate this difficulty, particular subclasses are often considered. The weighted join-free class has a limited expressiveness, since it forbids synchronizations. However, even with this strong structural limitation, this class remains hard to analyze and appears as a fundamental module of more complex classes, such as weighted asymmetric-choice nets whose behavior is strongly related to its join-free modules [1].

The model provided is inspired from a paper of T. Hujsa and R. Devillers [1] which investigates the relationship between liveness and structural boundedness (meaning boundedness for every initial marking) in some subclasses of weighted Petri nets, notably join-free nets. The paper also studies the monotonicity of liveness, meaning its preservation upon any increase of the live marking considered.

The model is based on one parameter $k$, which denotes:

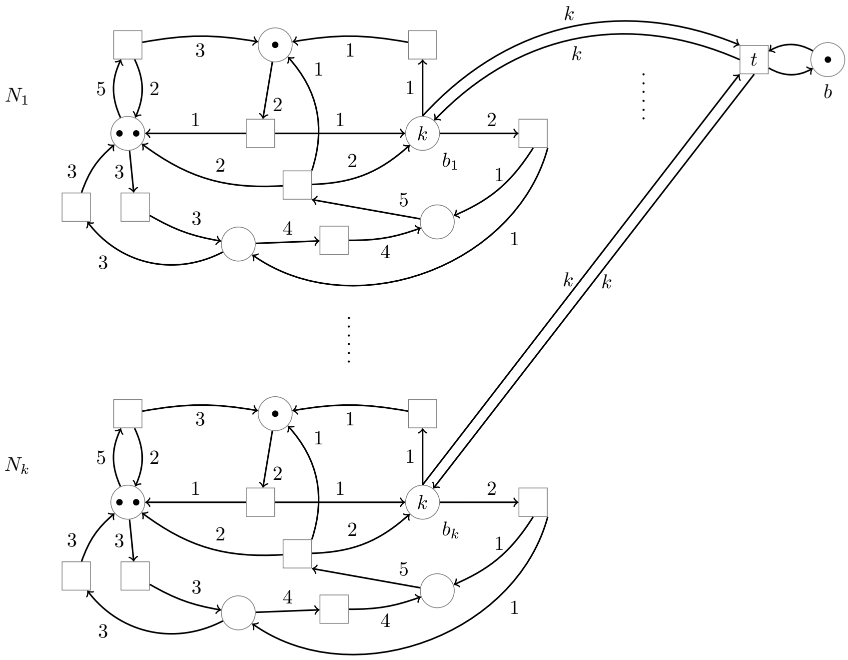
- the number of join-free modules,
- the initial number of tokens in each buffer (place) $b_1, \ldots, b_k$
- the amount of tokens read and written (as a self-loop) through one firing of $t$ (outside the modules) in each $b_1, \ldots, b_k$.

Each module $N_i$, $i \in \{1, \ldots, k\}$, is a structurally live and bounded join-free net containing the buffer $b_i$. A single transition $t$ checks the existence of $i$ tokens in each buffer $b_i$. Each $N_i$ can represent a sub-program with asynchronous iterations to be executed concurrently on different processors. Each new iteration in $N_i$ reads some data items (tokens) in its buffer $b_i$.

The sub-program $t$ reads $k$ data items in each buffer $b_i$, computes a function on them and updates the buffer $b$ with the result. The purpose of this operation is to analyse the current progress of each module and to gather this updated information in $b$. Due to this synchronisation $t$, the global system is not join-free.

Each join-free module $N_i$ is live for $k = 3$, becomes non-live for $k = 4, 5, 6$ (proving that liveness is not monotonic in the structurally bounded join-free class) and becomes live again for all $k \geq 7$ (this stems partly from a result of [2]). Each $N_i$ is live and reversible for all $k \geq 8$, implying that the global system is live and reversible for every $k \geq 8$, at least."

 *generated on January 2, 2024*




*For each $i \in \{1, \dots, k\}$, the subnet $N_i$ contains a place $b_i$ with $k$ initial tokens. The transition $t$ can fire when at least $k$ tokens are present in each $b_i$.*

## References

[1] Thomas Hujsa and Raymond Devillers. *On Liveness and Deadlockability in Subclasses of Weighted Petri Nets.* Proceedings of the 38th International Conference on Application and Theory of Petri Nets and Concurrency (Petri Nets'17), 2017.

[2] Jean-Marc Delosme, Thomas Hujsa, and Alix Munier-Kordon. *Polynomial sufficient conditions of well-behavedness for weighted Join-Free and Choice-Free systems.* In Proceedings of the 13th International Conference on Application of Concurrency to System Design (ACSD'13). pages 90–99, 2013.

## Scaling parameter

| **Parameter name** | **Parameter description** | **Chosen parameter values** |
|---|---|---|
| $k$ | The number of initial tokens in each buffer $b_i$ (in the module $N_i$), the number of tokens consumed and produced by each firing of $t$ in the same buffers, and the number of modules | 3, 4, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000 |

Thomas Hujsa
thomas.hujsa@uni-oldenburg.de

## Size of the model

| Parameter | Number of places | Number of transitions | Number of arcs |
|---|---|---|---|
| $k$ | $5k+1$ | $8k+1$ | $23k+2$ |
| $k=3$ | 16 | 25 | 71 |
| $k=4$ | 21 | 33 | 94 |
| $k=5$ | 26 | 41 | 117 |
| $k=10$ | 51 | 81 | 232 |
| $k=20$ | 101 | 161 | 462 |
| $k=50$ | 251 | 401 | 1 152 |
| $k=100$ | 501 | 801 | 2 302 |
| $k=200$ | 1 001 | 1 601 | 4 602 |
| $k=500$ | 2 501 | 4 001 | 11 502 |
| $k=1000$ | 5 001 | 8 001 | 23 002 |
| $k=2000$ | 10 001 | 16 001 | 46 002 |
| $k=5000$ | 25 001 | 40 001 | 115 002 |

## 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)* ✓ [e]
**strongly connected** — *there is a directed path between every two nodes (places or transitions)* ✓ [f]
**source place(s)** — *one or more places have no input transitions* ✗ [g]
**sink place(s)** — *one or more places have no output transitions* ✗ [h]
**source transition(s)** — *one or more transitions have no input places* ✗ [i]
**sink transitions(s)** — *one or more transitions have no output places* ✗ [j]
**loop-free** — *no transition has an input place that is also an output place* ✗ [k]
**conservative** — *for each transition, the number of input arcs equals the number of output arcs* ✓ [l]
**subconservative** — *for each transition, the number of input arcs equals or exceeds the number of output arcs* ✓ [m]
**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* ✗ [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* ?

(a) the net is not ordinary.
(b) the net is not ordinary.
(c) the net is not ordinary.
(d) the net is not ordinary.
(e) stated by CÆSAR.BDD version 2.7 on all 12 instances (see all aforementioned parameter values).
(f) stated by CÆSAR.BDD version 2.7 on all 12 instances (see all aforementioned parameter values).
(g) stated by CÆSAR.BDD version 2.7 on all 12 instances (see all aforementioned parameter values).
(h) stated by CÆSAR.BDD version 2.7 on all 12 instances (see all aforementioned parameter values).
(i) stated by CÆSAR.BDD version 2.7 on all 12 instances (see all aforementioned parameter values).
(j) stated by CÆSAR.BDD version 2.7 on all 12 instances (see all aforementioned parameter values).
(k) stated by CÆSAR.BDD version 2.7 on all 12 instances (see all aforementioned parameter values).
(l) stated by PNML2NUPN 3.1.0 on all 12 instances (see all aforementioned parameter values).
(m) stated by PNML2NUPN 3.1.0 on all 12 instances (see all aforementioned parameter values).
(n) the definition of Nested-Unit Petri Nets (NUPN) is available from `http://mcc.lip6.fr/nupn.php`
(o) stated by CÆSAR.BDD version 2.7 on all 12 instances (see all aforementioned parameter values).

Thomas Hujsa
thomas.hujsa@uni-oldenburg.de

**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* ................?

## 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 |
|---|---|---|---|---|
| For any $k$: | – | – | $\leq k+3$ | $k \cdot (k+3)+1$ |
| $k=3$ | 35 937 | 225 450 | $\leq 6$ | 19 [(p)] |
| $k=4$ | 14 776 336 | 138 230 321 | $\leq 7$ | 29 [(q)] |
| $k=5$ | ? | ? | $\leq 8$ | 41 [(r)] |
| $k=10$ | ? | ? | $\leq 13$ | 131 [(s)] |
| $k=20$ | ? | ? | $\leq 23$ | 461 [(t)] |
| $k=50$ | ? | ? | $\leq 53$ | 2 651 [(u)] |
| $k=100$ | ? | ? | $\leq 103$ | 10 301 [(v)] |
| $k=200$ | ? | ? | $\leq 203$ | 40 601 [(w)] |
| $k=500$ | ? | ? | $\leq 503$ | 251 501 [(x)] |
| $k=1000$ | ? | ? | $\leq 1\,003$ | 1 003 001 [(y)] |
| $k=2000$ | ? | ? | $\leq 2\,003$ | 4 006 001 [(z)] |
| $k=5000$ | ? | ? | $\leq 5\,003$ | 2 5015 001 [(aa)] |

## Other properties

P — and T – invariants exist and cover the net. Denoting by $L$ the liveness property and by $R$ the reversibility property, there exist markings satisfying:

- $L$ and $R$
- $L$ and not $R$
- not $L$ and $R$
- not $L$ and not $R$

For the value $k = 4$, a deadlock is reachable.

[(p)] number of initial tokens, because the net is conservative.
[(q)] number of initial tokens, because the net is conservative.
[(r)] number of initial tokens, because the net is conservative.
[(s)] number of initial tokens, because the net is conservative.
[(t)] number of initial tokens, because the net is conservative.
[(u)] number of initial tokens, because the net is conservative.
[(v)] number of initial tokens, because the net is conservative.
[(w)] number of initial tokens, because the net is conservative.
[(x)] number of initial tokens, because the net is conservative.
[(y)] number of initial tokens, because the net is conservative.
[(z)] number of initial tokens, because the net is conservative.
[(aa)] number of initial tokens, because the net is conservative.