

This form is a summary description of the model entitled “Railroad crossing” 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 model is the Petri nets semantics of an ABCD model of a railroad crossing system. It has three components: a gate sub-net (top of figure), a controller sub-net (split in 3 sub-figures, middle-right and bottom) and n tracks sub-nets that differ only by an identifier k in $\{0, \dots, n-1\}$ (middle-left of figure). These components communicate through shared places, some being low-level places to exchange signals, others being integer-valued places to exchange tracks identifiers. The controller also has a place to count the number of trains at a given time.

We list now all the places with there type, initial marking and properties:

- `green_one` (type $\{0, \dots, n-1\}$, marking $\{0, \dots, n-1\}$, colour-safe): a green light for each track;
- `green_all` (type $\{\bullet\}$, marking $\{\bullet\}$, 1-safe): a global flag to control all the tracks green lights;
- `down` (type $\{\bullet\}$, marking $\{\}$, 1-safe): a channel to command gates to go down;
- `up` (type $\{\bullet\}$, marking $\{\}$, 1-safe): a channel to command gates to go up;
- `enter` (type $\{\bullet\}$, marking $\{\}$, n -bounded): a channel to inform the controller that a train is approaching on any track;
- `leave` (type $\{0, \dots, n-1\}$, marking $\{\}$, colour-safe): a channel to inform the controller that a train is leaving on a particular track;
- `done` (type $\{0, 2\}$, marking $\{\}$, colour-safe): a channel to inform the controller that the gates have done moving (either up or down);
- `state` (type $\{0, 1, 2\}$, marking $\{2\}$, 1-safe): stores the current state of gates (closed, moving, opened);
- `crossingk` (type $\{\bullet\}$, marking $\{\}$, 1-safe): one place for each track k that holds a token when a train on this track is crossing the gates;
- `count` (type $\{0, \dots, n-1\}$, marking $\{0\}$, 1-safe): number of trains currently present.
- all the other places are control-flow places, and thus have type $\{\bullet\}$ and are 1-safe ($2+n$ control flow places are unmarked and isolated so they can be ignored and they are not depicted in the figure).

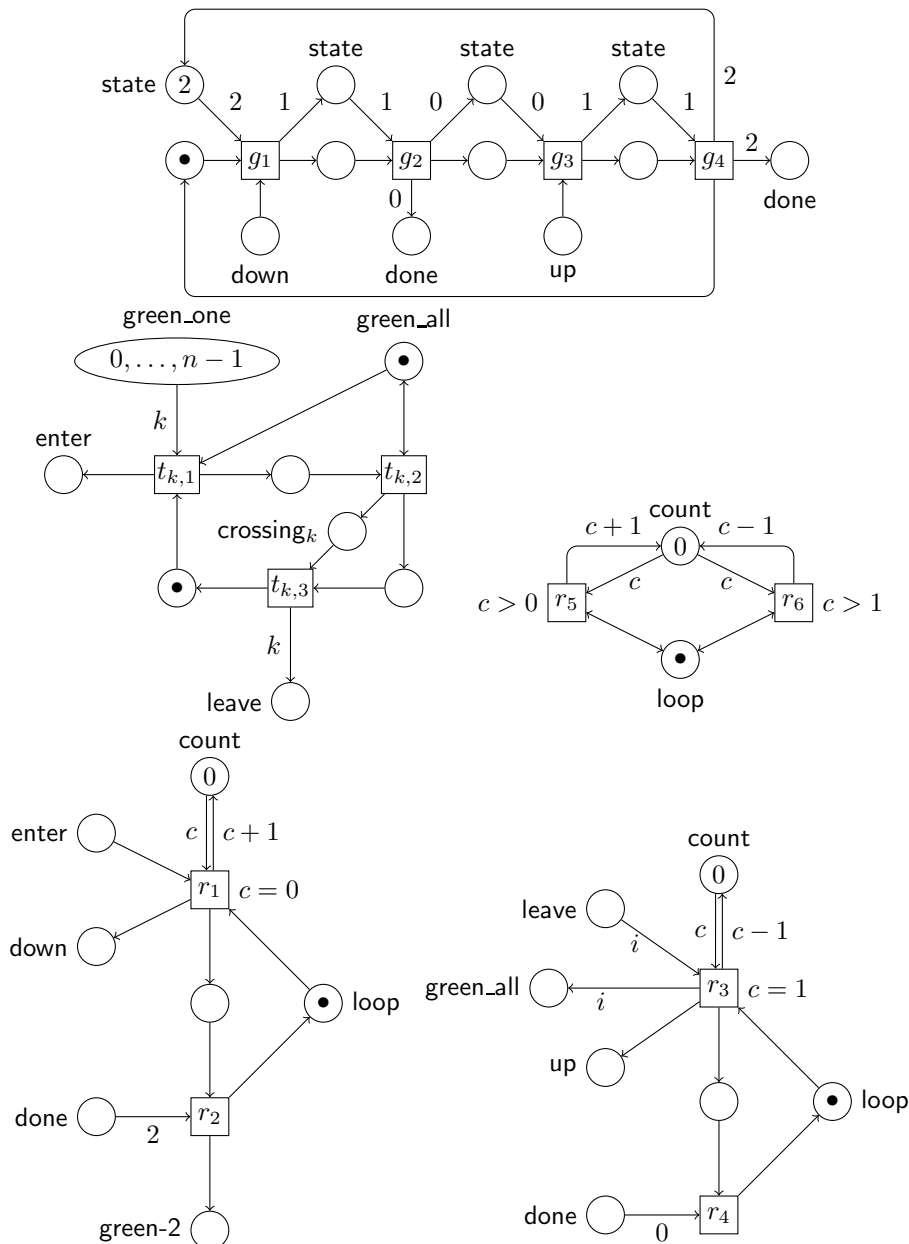
For the sake of readability, some places in the figure have been depicted several times, in which case they are labelled with a name. Moreover, unlabelled places have type $\{\bullet\}$ and unlabelled arcs carry a single token \bullet .

The transitions are as follows:

- g_1 (resp. g_3) models the receiving by the gates of a signal to go down (resp. up);
- g_2 (resp. g_4) models the full closing (resp. opening) of the gates;
- $t_{k,1}$, $t_{k,2}$ and $t_{k,3}$ respectively model for track k , the arrival of a train, the entering of a train in between the gates, and the leaving of a train;
- r_1 models the receiving by the controller of the signal that a first train arrived;
- r_5 models the receiving by the controller of the signal that one more train arrived;
- r_2 models the receiving by the controller of the signal that the gates finished to close;
- r_3 models the receiving by the controller of the signal that the last train exited;

- r_6 models the receiving by the controller of the signal that one train exited while there are still trains present;
- r_4 models the receiving by the controller of the signal that the gates finished to open.

r_1, r_3, r_5 and r_6 are the only guarded transitions (*i.e.*, all the other guards are true).



Scaling parameter

| Parameter name | Parameter description | Chosen parameter values |
|----------------|-----------------------|-------------------------|
| n | number of tracks | 5, 10, 20, 50, 100 |

Size of the model

| Parameter | Number of places | Number of transitions | Number of arcs |
|-----------|------------------|-----------------------|----------------|
| $n = 5$ | 68 | 56 | 313 |
| $n = 10$ | 118 | 156 | 898 |
| $n = 20$ | 218 | 506 | 2968 |
| $n = 50$ | 518 | 2756 | 16378 |
| $n = 100$ | 1018 | 10506 | 62728 |

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) |
| deadlock — there exists a reachable marking from which no transition can be fired | ✗ ^(p) |
| reversible — from every reachable marking, there is a transition path going back to the initial marking | ? ^(q) |
| quasi-live — for every transition t , there exists a reachable marking in which t can fire | ✗ ^(a) |
| live — for every transition t , from every reachable marking, one can reach a marking in which t can fire | ✗ ^(r) |

(a) stated by [CÆSAR.BDD](#) version 1.7 on all 5 instances (5, 10, 20, 50, and 100).
 (b) stated by [CÆSAR.BDD](#) version 2.6 on all 5 instances (5, 10, 20, 50, and 100).
 (c) stated by [CÆSAR.BDD](#) version 1.7 on all 5 instances (5, 10, 20, 50, and 100).
 (d) stated by [CÆSAR.BDD](#) version 1.7 on all 5 instances (5, 10, 20, 50, and 100).
 (e) stated by [CÆSAR.BDD](#) version 1.7 on all 5 instances (5, 10, 20, 50, and 100).
 (f) the net is not connected and, thus, not strongly connected.
 (g) stated by [CÆSAR.BDD](#) version 1.7 on all 5 instances (5, 10, 20, 50, and 100).
 (h) stated by [CÆSAR.BDD](#) version 1.7 on all 5 instances (5, 10, 20, 50, and 100).
 (i) stated by [CÆSAR.BDD](#) version 1.7 on all 5 instances (5, 10, 20, 50, and 100).
 (j) stated by [CÆSAR.BDD](#) version 1.7 on all 5 instances (5, 10, 20, 50, and 100).
 (k) stated by [CÆSAR.BDD](#) version 1.7 on all 5 instances (5, 10, 20, 50, and 100).
 (l) stated by [CÆSAR.BDD](#) version 1.7 on all 5 instances (5, 10, 20, 50, and 100).
 (m) stated by [CÆSAR.BDD](#) version 1.7 on all 5 instances (5, 10, 20, 50, and 100).
 (n) the definition of Nested-Unit Petri Nets (NUPN) is available from <http://mcc.lip6.fr/nupn.php>
 (o) stated by [CÆSAR.BDD](#) version 2.0 to be true on 2 instance(s) out of 5, and unknown on the remaining 3 instance(s).
 (p) stated by [CÆSAR.BDD](#) version 2.0 to be false on 2 instance(s) out of 5, and unknown on the remaining 3 instance(s); confirmed at MCC'2014 by Lola and Tapaal on 2 instances ($N = 5$ and $N = 10$).
 (q) stated by [CÆSAR.BDD](#) version 2.0 on all 5 instances (5, 10, 20, 50, and 100).
 (r) the net is not quasi-live and, thus, not live.

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 |
|-----------|--|------------------------------------|---------------------------------|-----------------------------------|
| $n = 5$ | 1 838 ^(s) | 7699 ^(t) | 1 ^(u) | 16 ^(v) |
| $n = 10$ | 2.038×10^6 ^(w) | 1.632×10^7 ^(x) | 1 ^(y) | 26 ^(z) |
| $n = 20$ | 2.195×10^{12} ^(aa) | 1 ^(ab) | ? | 46 ^(ac) |
| $n = 50$ | ? | ? | ? | ≥ 105 |
| $n = 100$ | ? | ? | ? | ≥ 205 |

^(s) computed at MCC'2013 by Alpina, ITS-Tools, Marcie, Neco, and PNXDD; confirmed by [CESAR.BDD](#) version 1.8; confirmed at MCC'2014 by GreatSPN, Marcie, PNMC, PNXDD, Stratagem, and Tapaal.

^(t) computed at MCC'2014 by Marcie.

^(u) computed at MCC'2014 by GreatSPN, Marcie, PNMC, and Tapaal.

^(v) computed at MCC'2014 by GreatSPN, Marcie, PNMC, and Tapaal.

^(w) computed at MCC'2013 by ITS-Tools, Marcie, Neco, and PNXDD; confirmed by [CESAR.BDD](#) version 1.8; confirmed at MCC'2014 Marcie, PNMC, PNXDD, and Stratagem.

^(x) computed at MCC'2014 by Marcie.

^(y) computed at MCC'2014 by Marcie and PNMC.

^(z) computed at MCC'2014 by Marcie and PNMC.

^(aa) computed at MCC'2014 by PNMC and PNXDD.

^(ab) computed at MCC'2014 by PNMC.

^(ac) computed at MCC'2014 by PNMC.