Silvano Dal Zilio
dalzilio@laas.fr

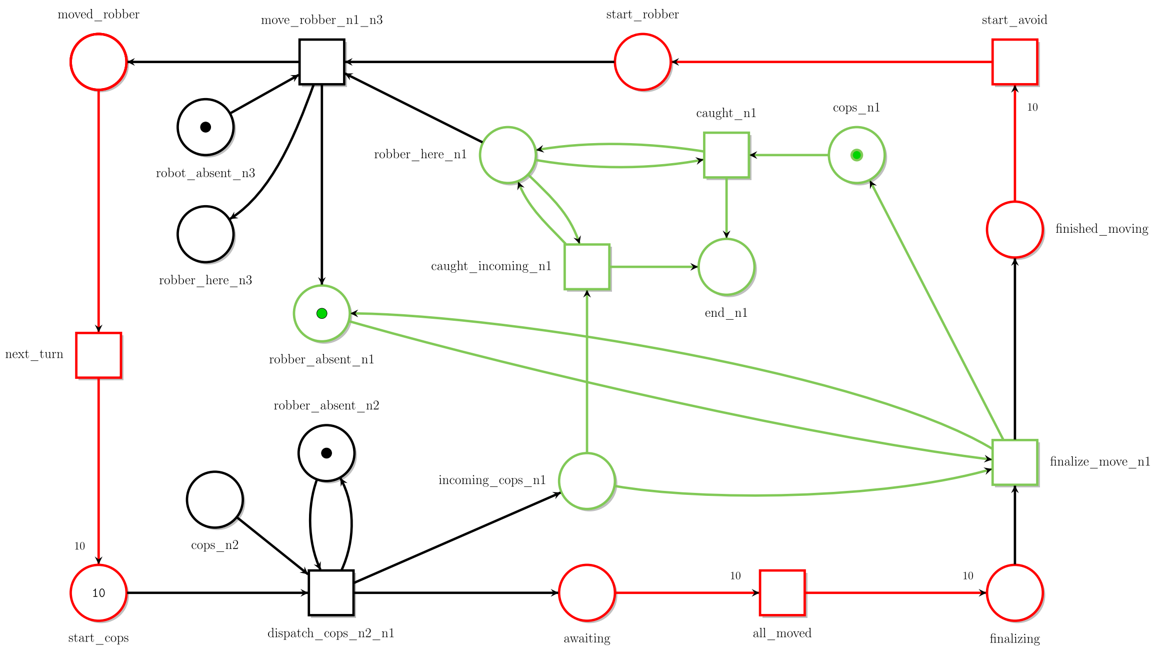
This form is a summary description of the model entitled "CopsAndRobbers" proposed for the Model Checking Contest @ Petri Nets. Models can be given in several instances parameterized by scaling parameters. Colored nets can be accompanied by one or many equivalent, unfolded P/T nets. Models are given together with property files (possibly, one per model instance) giving a set of properties to be checked on the model.

Description

Cops and Robbers [1] is a classic example of pursuit-evasion game played on graph structures, often studied in the field of game theory and artificial intelligence. It is notably connected with the result by Seymour and Thomas [2] which relates the tree-width of a graph with the number of cops necessary to have a winning strategy.

In our version of this game, a unique robber and one or several cops (a number determined by the value of parameter X), move freely along the edges of a fully connected, undirected graph of size L (another parameter). The goal for the cops is to capture the robber by moving to its position. We use replicated instances of the same component to model cops and robbers moving across nodes (indexed from `n1` to `nL`). Our model also uses several shared places to enforce an order on the different phases of our game. Specifically, all cops start moving first, and then the robber tries to escape. Each cop can move only once during its phase, but several cops can be in the same position. We take extra steps to ensure that if a cop share the same position as the robber, or attempts to move into it, then the model will eventually deadlock after a bounded number of transitions (which depends on the number of cops).

We consider three main categories of graphs, parametrized by their length / number of nodes (L): `Circular` is a simple cycle of size L, whereas `Heawood` and `Kneser` (N, K) are families of graphs with small degree but large girth. Initially, the cops and robbers are randomly dispatched to their starting position, except for instances built from a Kneser graph, usually much larger, where they have fixed, evenly distributed initial positions. This is reflected in the name of the models using the suffixes `Random` and `Fixed`.



Part of the model corresponding to node `n1`*. Places and transitions whose name is not suffixed by a node (in red) are shared. One can follow the lifecycle of a chase by looking at the sequence of transitions on the perimeter of this net.*

References

1. A. Quilliot. *"A short note about pursuit games played on a graph with a given genus."* Journal of combinatorial theory, Series B 38.1 (1985).

Silvano Dal Zilio
dalzilio@laas.fr

2. Paul D. Seymour, and Robin Thomas. *"Graph searching and a min-max theorem for tree-width."* Journal of Combinatorial Theory, Series B 58.1 (1993).

Scaling parameter

Parameter name	Parameter description	Chosen parameter values
Circular (L, X) or Heawood (L, X) or Kneser (L, N, K, X)	L is the number of nodes in the underlying graph, and X is the number of cops. For Heawood graphs, the number of nodes is equal to $3\,N + 1$, for some $N \geq 5$. For Kneser graphs, the value of L is $\binom{N}{K}$, with $2\,K < N$.	Circular $(5, 1)$, $(15, 5)$, $(23, 4)$, $(25, 4)$, Heawood $(14, 2)$, $(17, 5)$, $(20, 6)$, $(23, 6)$, Kneser $(10, 5, 2, 1)$, $(28, 8, 2, 5)$, $(36, 9, 2, 3)$, $(84, 9, 3, 2)$

Size of the model

Parameter	Number of places	Number of transitions	Number of arcs
Circular (L, X)	$10 + 5\,L$	$4 + 7\,L$	$9 + 33\,L$
Heawood (L, X)	$10 + 5\,L$	?	?
Kneser (L, N, K, X)	$6 + 5\,L$	?	?

Structural properties

ordinary — *all arcs have multiplicity one* ..... ? [a]
simple free choice — *all transitions sharing a common input place have no other input place* ..... no [b]
extended free choice — *all transitions sharing a common input place have the same input places* ..... no [c]
state machine — *every transition has exactly one input place and exactly one output place* ..... no [d]
marked graph — *every place has exactly one input transition and exactly one output transition* ..... no [e]
connected — *there is an undirected path between every two nodes (places or transitions)* ..... yes [f]
strongly connected — *there is a directed path between every two nodes (places or transitions)* ..... no [g]
source place(s) — *one or more places have no input transitions* ..... ? [h]
sink place(s) — *one or more places have no output transitions* ..... yes [i]
source transition(s) — *one or more transitions have no input places* ..... no [j]
sink transitions(s) — *one or more transitions have no output places* ..... no [k]
loop-free — *no transition has an input place that is also an output place* ..... no [l]
conservative — *for each transition, the number of input arcs equals the number of output arcs* ..... ? [m]
subconservative — *for each transition, the number of input arcs equals or exceeds the number of output arcs* ..... ? [n]
nested units — *places are structured into hierarchically nested sequential units* [o] ..... no

(a) stated by CÆSAR.BDD version 3.7 to be true on 2 instance(s) out of 12, and false on the remaining 10 instance(s).
(b) stated by CÆSAR.BDD version 3.7 on all 12 instances (3 types of models (Circular, Heawood, Kneser) × 4 instances).
(c) stated by CÆSAR.BDD version 3.7 on all 12 instances (3 types of models (Circular, Heawood, Kneser) × 4 instances).
(d) stated by CÆSAR.BDD version 3.7 on all 12 instances (3 types of models (Circular, Heawood, Kneser) × 4 instances).
(e) stated by CÆSAR.BDD version 3.7 on all 12 instances (3 types of models (Circular, Heawood, Kneser) × 4 instances).
(f) stated by CÆSAR.BDD version 3.7 on all 12 instances (3 types of models (Circular, Heawood, Kneser) × 4 instances).
(g) stated by CÆSAR.BDD version 3.7 on all 12 instances (3 types of models (Circular, Heawood, Kneser) × 4 instances).
(h) stated by CÆSAR.BDD version 3.7 to be true on 8 instance(s) out of 12, and false on the remaining 4 instance(s).
(i) stated by CÆSAR.BDD version 3.7 on all 12 instances (3 types of models (Circular, Heawood, Kneser) × 4 instances).
(j) stated by CÆSAR.BDD version 3.7 on all 12 instances (3 types of models (Circular, Heawood, Kneser) × 4 instances).
(k) stated by CÆSAR.BDD version 3.7 on all 12 instances (3 types of models (Circular, Heawood, Kneser) × 4 instances).
(l) stated by CÆSAR.BDD version 3.7 on all 12 instances (3 types of models (Circular, Heawood, Kneser) × 4 instances).
(m) stated by CÆSAR.BDD version 3.7 to be true on 1 instance(s) out of 12, and false on the remaining 11 instance(s).
(n) stated by CÆSAR.BDD version 3.7 to be true on 1 instance(s) out of 12, and false on the remaining 11 instance(s).
(o) the definition of Nested-Unit Petri Nets (NUPN) is available from `http://mcc.lip6.fr/nupn.php`

Silvano Dal Zilio
dalzilio@laas.fr

Behavioural properties

safe — *in every reachable marking, there is no more than one token on a place* .... ? [p]
dead place(s) — *one or more places have no token in any reachable marking* .... no
dead transition(s) — *one or more transitions cannot fire from any reachable marking* .... no [q]
deadlock — *there exists a reachable marking from which no transition can be fired* .... yes [r]
reversible — *from every reachable marking, there is a transition path going back to the initial marking* .... no [s]
live — *for every transition t, from every reachable marking, one can reach a marking in which t can fire* .... no [t]

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
Circular $(5, 1)$	166 [u]	215 [v]	1 [w]	8 [x]
Circular $(15, 5)$	8 396 499	19 876 830	5	26
Circular $(23, 4)$	10 376 566	21 409 872	4	32
Circular $(25, 3)$	1 285 276	2 231 400	3	32
Heawood $(14, 2)$	17 158	31 556	2	19
Heawood $(17, 5)$	21 026 621	100 942 456	5	28
Heawood $(20, 6)$	?	?	6	≥ 27 [y]
Heawood $(23, 6)$	974 061 455	5 728 869 122	6	36
Kneser $(10, 5, 2, 1)$	620	980	1	12
Kneser $(28, 8, 2, 5)$	345 513 308	6 394 296 188	5	38
Kneser $(36, 9, 2, 3)$	5 473 188	83 869 344	3	42
Kneser $(84, 9, 3, 2)$	?	?	5	≥ 88 [z]

Other properties

A winning configuration in the cops and robbers game, depending on your perspective, is a marking in which one of the `end` place is marked, meaning that a cop was in the same node as the robber, or in the process of moving into it. By design, this should (eventually) lead to a deadlock. We can also try to find a scenario in which a robber could indefinitely avoid being caught. The verdict in this case may depend on the value of parameter X. This gives the following two interesting CTL formulas, where **end** is the sum of markings in the places `end_n1` to `end_nL`.

$$\text{EVASION} \;:\; \text{EG}\;(\mathbf{end} = 0)$$

$$\text{JAIL} \;:\; \text{AG}\;\left((\mathbf{end} \geq 1) \implies \text{AF dead}\right)$$

[p] stated by CÆSAR.BDD version 3.7 to be true on 2 instance(s) out of 12, and false on the remaining 10 instance(s).
[q] this is false unless the graph has a single node, which is never the case with our instances.
[r] by design.
[s] because of deadlocks.
[t] because of deadlocks.
[u] computed by TINA version 3.8.5 on February 2025.
[v] computed by TINA version 3.8.5 on February 2025.
[w] by construction it is equal to the number of cops.
[x] computed by TINA version 3.8.5 on February 2025.
[y] lower bound given by the number of initial tokens.
[z] lower bound given by the number of initial tokens.