# Model Checking Contest 2018

## Rules

http://mcc.lip6.fr

## Contents

## Introduction

The Model Checking Contest is a yearly scientific event dedicated to the assessment of formal verification tools for concurrent systems.

The Model Checking Contest has two different parts: the Call for Models, which gathers Petri net models proposed by the scientific community, and the Call for Tools, which benchmarks verification tools developed within the scientific community.

Figure 1 illustrates, for your understanding, the workflow of a tool in the Model Checking Contest. The upper level depicts the behavior of a tool developer and the lower level show the way we handle the tools.

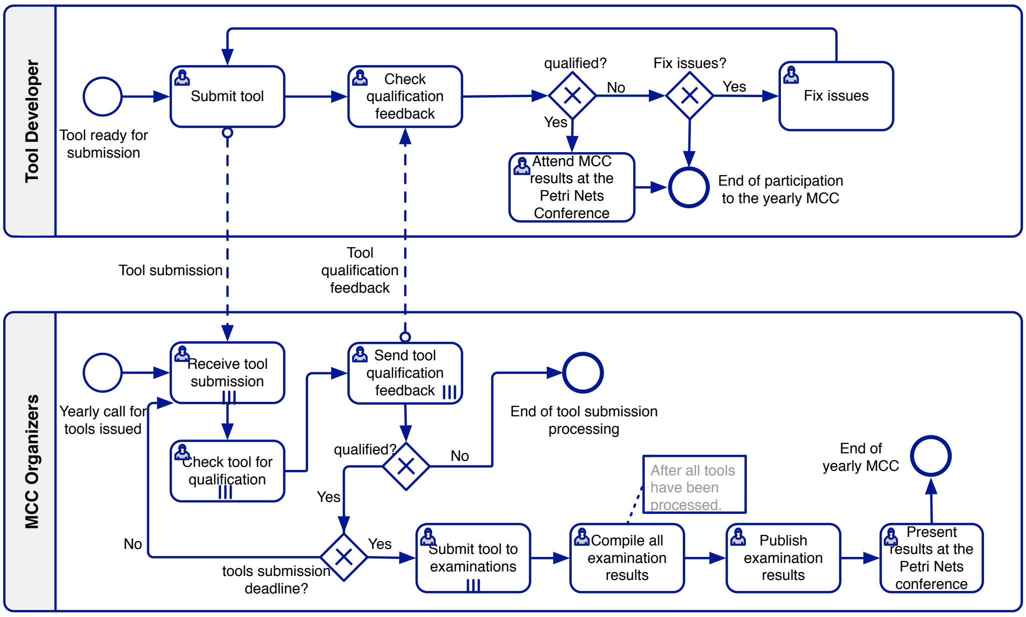
Tool Developer
Tool ready for submission
Submit tool
Check qualification feedback
qualified?
No
Yes
Fix issues?
Yes
Fix issues
Attend MCC results at the Petri Nets Conference
End of participation to the yearly MCC
Tool submission
Tool qualification feedback
MCC Organizers
Yearly call for tools issued
Receive tool submission
Check tool for qualification
Send tool qualification feedback
qualified?
No
Yes
End of tool submission processing
After all tools have been processed.
End of yearly MCC
No
tools submission deadline?
Yes
Submit tool to examinations
Compile all examination results
Publish examination results
Present results at the Petri Nets conference

Figure 1: BPMN description of the Tool submission workflow.

## I Model Submission and their Publication

**M-1.** The Organizers of the Call for Models of the 2018 edition of the Model Checking Contest are: Hubert Garavel, Lom Hillah, and Fabrice Kordon.

**M-2.** The detailed instructions for proposing a model are given in the **call for models** and **the model submission toolkit**.

**M-3.** Any model submitted should be in the public domain we agree on models whose origin has been hidden by changing the names of places and transitions). Any model that is proprietary, copyrighted, and/or confidential should not be submitted (unless it is under GPL, Creative Commons or any license allowing a wide distribution). Persons who submit a model hereby allow the organizers to freely use this model and publish it in the web if this model is selected.

**M-4.** If a submitted model is improper (e.g., invalid PNML file, missing or incomplete model description form, etc.), the organizers may contact the person(s) who submitted the model and ask for corrections. If corrections are not done or are unsatisfactory, the organizers can reject the model.

**M-5.** Among all submitted models, the Organizers will select some models according to criteria such as: originality, complexity, diversity with respect to other models, industrial or scientific relevance, etc.

**M-6.** The list of models selected for the 2018 edition of the Model Checking Contest will be kept confidential by the Organizers until all tools have been submitted, so that selected models are not known in advance by tool developers.

**M-7.** When tool evaluation starts, the selected "surprise" models will be published on the web site and the names of the authors of selected models will be mentioned on the same web site. The Organizers of the Call for Models will not submit tools themselves, and will ensure fair competition between all tool developers competing in the Call for Tools.

# II Tool Submission and their Publication

**T-1.** The Organizers of the Call for Tools of the 2018 edition of the Model Checking Contest are: Francis Hulin-Hubard, and Fabrice Kordon.

**T-2.** Developers of a tool (or a tool extension) can participate with their own work. Tools may also be submitted by others if they present the latest version available on the Web, and with the agreement of their authors.

**T-3.** All MCC models are provided in PNML, which is the standard exchange format for Petri nets (here, seen as a way to specify concurrent systems). In principle, each submitted tool should be able to read PNML files.

**T-4.** Participation is allowed for certain models only and/or certain examinations only (e.g., state space generation, some types of formulas, etc.). See the Submission Manual for details.

**T-5.** Submitted tools must honestly perform the computations required by the MCC examinations. Using precomputed results is not allowed and, if detected, will disqualify the tool submission.

**T-6.** Each examination must be computed independently, i.e. without using previous results of an earlier computation. However, for examination based on formulas, when a file contains several formulas, the tool may reuse the results of certain formulas to compute other formulas.

**T-7.** Participants authorize the publication of the MCC results.

**T-8.** Participants authorize the publication of the disk image containing the virtual machine running their tool in order to ease reproducibility of results.

**T-9.** By submitting a tool to the MCC, each participant thereby agrees to be bound to all the rules stated in this document.

# III Evaluation and Scoring of Tools

This section defines the way we intend to operate tools, evaluate tools, and compute scores.

## III.1 Description of the Examinations for 2018

*This section defines the examinations for 2018, as well as the outputs expected from tools for these examinations.*

**E-1.1.** There are five categories of examinations: **StateSpace**, **UpperBounds**, **Reachability** formulas, **CTL** formulas, and **LTL** formulas.

**E-1.2.** For the **StateSpace** category, each participating tool must provide the following results: number of states in the marking graph, number of transitions firing in the marking graph (including duplicates[1]), the maximum number of tokens per marking in the net, and the maximum number of tokens that can be found in a place.

**E-1.3.** For the **UpperBounds** category, each participating tool must provide the bound of places designated in a formula as an integer value, specifying the upper bound of the place. For each designated place, the provided value is expected to be its exact upper bound.

[1] This decision was taken based on a discussion between tool developers and the MCC organizers from June to October 2017.

For each examination of the **UpperBounds** category, 16 formulas (selected places) are proposed.

**E-1.4.** The **Reachability** formulas category contains three subcategories : **ReachabilityDeadlock**, **ReachabilityFireability**, and **ReachabilityCardinality**. Each tool must provide (when it answers):

- for **ReachabilityDeadlock**, a boolean TRUE or FALSE indicating the existence (or not) of at least one reachable deadlock state in the system,
- for **ReachabilityFireability** and **ReachabilityCardinality**, a sequence of booleans TRUE or FALSE, each stating whether a formula is satisfied.

The differences between the three subcategories lie in the atomic propositions of the formula. **ReachabilityDeadlock** corresponds to a simple formula stating the existence of a deadlock state. In **ReachabilityFireability**, atomic propositions are boolean combinations of propositions checking for the firability of transitions. In **ReachabilityCardinality** atomic propositions are boolean combinations of propositions comparing the number of tokens in places.

For each examination of the **ReachabilityFireability** and **ReachabilityCardinality** categories, 16 formulas are proposed (see rule E-1.9 for details on their elaboration).

**E-1.5.** The **CTL** formulas category contains two subcategories : **CTLFireability**, and **CTLCardinality**. Each tool must provide (when it answers), a sequence of booleans TRUE or FALSE stating whether a formula is satisfied.

The differences between the two subcategories lie in in the atomic propositions of the formula. In **CTLFireability**, atomic propositions are boolean combinations of propositions checking for the firability of transitions. In **CTLCardinality**, atomic propositions are boolean combinations of propositions comparing the number of tokens in places.

For each examination of the **CTL** category, 16 formulas are proposed (see rule E-1.9 for details on their elaboration).

**E-1.6.** The **LTL** formulas category contains two subcategories : **LTLFireability**, and **LTLCardinality**. Each tool must provide (when it answers), a sequence of booleans TRUE or FALSE stating whether a formula is satisfied.

The differences between the two subcategories lie in in the atomic propositions of the formula. In **LTLFireability**, atomic propositions are boolean combinations of propositions checking for the firability of transitions. In **LTLCardinality**, atomic propositions are boolean combinations of propositions comparing the number of tokens in places.

For each examination of the **LTL** category, 16 formulas are proposed (see rule E-1.9 for details on their elaboration).

**E-1.7.** If a tool participates in an examination in a category, its answers must be returned using a dedicated keyword (see the submission manual for details on the answering protocol). If the tool does not participate, it must state it explicitly. If the tool cannot compute a result, it must state it explicitly with a dedicated keyword. When the analysis script does not find any appropriately formated answer, it will assume that the tool could not compute anything.

**E-1.7.** All instructions with regards to outputs are presented in the submission manual. These instructions must be respected. Mistakes in the automatic evaluation of results that are due to not complying with these instructions are not the responsibility of the MCC organizers.

**E-1.8.** For all examinations involving formulas, the reference input is the XML representation of these formulas, the textual format is just provided to ease the readability.

**E-1.9.** For examinations involving formulas, new sets of formulas are computed every year and published once all tools have been submitted and the qualification phase is completed[2]. If the grammar for formula

[2] This decision was taken based on a discussion between tool developers and the MCC organizers from June to October 2017.

changes, a significative set of rules involving these changes will be published before the contest, otherwise, formulas from 2015, 2016 and 2017 are available (the syntax has not changed since 2015).

**E-1.10.** When answering an examination, tools state the techniques that were actually activated for this examination. When an examination contains several questions (e.g. several formulas), the activated techniques should be the stated specifically for each formula (this could allow to detect more precisely what technique were activated to solve a given problem).

## III.2 Evaluation of Tools Confidence in 2018

*This section defines the notion of tool Confidence, to be computed during a preliminary pass when compiling examination results, once all the tools have been submitted to all the examinations (see Figure 1).*

**E-2.1.** The Confidence of a tool is stated by a value $C_{tool} \in [0, 1]$. Where 1 means the tool always finds the commonly agreed results within a set of trusted values (see rule E-2.2 and section III.3). The value 0 means that the tool never finds the commonly agreed results within the same set of trusted values.

**E-2.2.** The evaluation of the participating tools Confidence is performed during a preliminary pass that checks for all the values provided by tools. A value is a piece of result for a given examination. For example, if an examination requires $N$ values, then, each of these will be considered separately to evaluate tools Confidence. Typically, the state space examination requires tools to compute 4 values that are to be considered separately when evaluating the tools Confidence.

From the full list of values provided by tools for the Model Checking Contest, let $V$ be the subset of trusted values that are defined in the following way:

- A majority of tools agree on the same value,
- and at least 3 tools agree on this value.

Let $V_{tool} \subseteq V$ be the set of values provided by a given tool within the set of the trusted values $V$ (not computed results are of course not considered), that also agree with those of $V$. Then, $C_{tool}$, the confidence of a tool is computed as follows :

$$C_{tool} = \frac{|V_{tool}|}{|V|}$$

where $|set|$ represents the cardinality of $set$.

## III.3 Computing the Trusted Values in 2018

*This section deals with how we compute the reference values that are used to evaluate the results provided by tools. The objective is to determine the correct result that is a priori unknown in most cases. We must consider two cases: when the expected result is a value and when the expected result is an interval.*

**E-3.1.** For a given examination where exact values are expected, all the values provided by the participating tools (e.g those that answered something else than "I cannot compute" or "I do not compete") are examined to determine the trusted value. This is evaluated according to the majority of the participating tools. In that process, each tool is weighted by its Confidence rate $C_{tool}$. Several situations are considered:

- All tools agree, then the reference value is a global agreement that is considered to be the trusted value,
- There is a weighted majority of tools that agree on one value that is considered to be the trusted value.
- Only two tools ($t_1$ and $t_2$) provide different values. If $C_{t_1} < C_{t_2}$, the value provided by $t_2$ is considered to be the trusted value. If $C_{t_2} < C_{t_1}$, the value provided by $t_1$ is considered to be the trusted value. Otherwise, the reference value is declared to be unknown.

- Only one tool provides a value. Then, if $C_{tool} \geqslant 0.97$, the value is considered to be the a correct value. Otherwise, the value is declared to be unknown.
- No tool provides any value, then the trusted value is declared to be unknown.

### III.4 Scoring for 2018

*This section provides information on how we compute a score from a value provided by a tool.*

**E-4.1.** Let us define the following constants that are used to compute scores in the model checking contest:

- $ScoreValue$: this is the number of points a tool gets when it computes a value correctly in an examination. The corresponding number depends on the examination, the total number of points a tool can receive from an examination being 16 (e.g. when 16 formulas are provided, each correctly computed formula is rewarded with 1 point).
- $PenaltyValue$: this is the number of points a tool looses when a value is considered to be wrong. It is worth twice the points when a tool has provided the correct value.

**E-4.2.** When a tool provides a value corresponding to the trusted value for an examination, it gets $ScoreValue$ points. The tool gets 0 points for non-computed values. For each wrong value provided, the tool gets the (penalty) score of $PenaltyValue$ points. The score for the examination is a sum of the scores obtained for the values.

**E-4.3.** Each time a tool computes all trusted results for a given examination (*i.e.* the 4 values for **StateSpace**, the value for **ReachabilityDeadlock**, or the 16 values for other formula examinations), computation time and memory consumption are measured. This information is reported separately but no score is associated[3].

**E-4.4.** There are two types of models in the competition[4]:

1. "known" models are the models gathered from the community since the beginning of the contest. These models are provided in the PNML standard and cannot be changed. However, they are already available for testing at `https://pnrepository.lip6.fr`.
2. "surprise" models are models gathered for 2018 from the community. They are provided to the community only once tools have been submitted and the qualification phase is completed.

There are score multipliers depending on the types of models: for "known" models, it is ×1, and for "surprise" models, it is ×4.

Some models are parameterized so that several *instances* can be deduced with different complexity based on the parameters while some other models provide only one *instance*. To avoid too much disparity, scores are normalized so that the maximum score of the model (*i.e.* the sum of the score for all its instances) with the largest number of instances cannot weight more than ×2 the score of a model providing only one instance.

Models may contain some structural information when it is known. So far, we provide informations about the 1-boundedness[5] of models that can be useful for some optimizations.

**E-4.5.** There are five podiums in the contest:

- **StateSpace** : it ranks tools for the `StateSpace` examination,
- **UpperBounds** : it ranks tools for the `UpperBounds` examination,
- **Reachability Formulas** : it ranks tools for the `ReachabilityCardinality`, `ReachabilityDeadlock`, and `ReachabilityFireability` examinations,

[3] This decision was taken based on a discussion between tool developers and the MCC organizers from June to October 2017.

[4] This rule aggregates decisions taken based on a discussion between tool developers and the MCC organizers from June to October 2017.

[5] The "safe" property in model forms.

- **CTL Formulas** : it ranks tools for the CTLCardinality and CTLFireability examination,
- **LTL Formulas** : it ranks tools for the LTLCardinality and LTLFireability examination,

For each podium, a "road-runner award" is provided for he fastest tool and the "camel award" is provided for the tool requiring the smallest amount of memory. Such awards are computed, for each tools, based on the examinations for which all the required values have been provided.

**E-4.6.** For each category, and for each tool, a total score is computed as the sum of all the examination scores of the tool in the examinations enclosed in this category (see rule E-4.5). Then, tools are ranked in this category with respect to this sum. If a tool is submitted with several variants, then, the best variant for the current category is considered for the podium only (the score of other variants are displayed for information).

## III.5 Execution Conditions for Tools in 2018

*This section defines the execution conditions for all tools.*

**E-5.1.** For a given examination, each tool is executed in a virtual machine. This machine is booted from a reference disk image, then the calculus is performed. Results are retrieved, and the machine is halted so that no information can be saved from one execution to another one.

For a given model, all examinations on all instances are processed for all tools on the same physical machine so that results can be compared fairly in terms of performances (CPU, memory).

For each examination, there is a CPU time limit of one hour. Once this time limit is reached, the virtual machine is halted and non provided values are considered as "cannot compute".

**E-5.2.** The CPU emulated in the virtual machine corresponds to the SandyBridge parameter in the -cpu option of the qemu-system command. It enables the following flags:

| flag | signification |
|---:|---|
| de | Debugging Extensions (CR4.DE) |
| pse | Page Size Extensions(4MB memory pages) |
| tsc | Time Stamp Counter (RDTSC) |
| msr | Model-Specific Registers (RDMSR) |
| pae | Physical Address Extensions (support for more than 4GB of RAM) |
| mce | Machine Check Exception |
| cx8 | CMPXCHG8 instruction (64-bit compare-and-swap) |
| apic | Onboard APIC |
| sep | SYSENTER/SYSEXIT |
| mtrr | Memory Type Range Registers |
| pge | Page Global Enable (global bit in PDEs and PTEs) |
| mca | Machine Check Architecture |
| cmov | CMOV instructions (conditional move) |
| pat | Page Attribute Table |
| pse36 | 36-bit PSEs (huge pages) |
| clflush | Cache Line Flush instruction |
| mmx | Multimedia Extensions |
| fxsr | FXSAVE/FXRSTOR (CR4.OSFXSR) |
| sse | Intel SSE vector instructions |
| sse2 | SSE2 |
| syscall | SYSCALL (Fast System Call) and SYSRET (Return From Fast System Call) |
| nx | Execute Disable |
| lm | Long Mode(x86-64: amd64 |

| flag | signification |
|---:|---|
| nopl | The NOPL (0F 1F) instructions |
| pni | SSE-3 ("Prescott New Instructions") |
| pclmulqdq | Perform a Carry-Less Multiplication of Quadword instruction accelerator for GCM) |
| ssse3 | Supplemental SSE-3 |
| cx16 | CMPXCHG16B |
| sse4_1 | SSE-4.1 |
| sse4_2 | SSE-4.2 |
| x2apic | x2APIC |
| popcnt | Return the Count of Number of Bits Set to 1, instruction Hamming weight |
| tsc_deadline_timer | Tsc deadline timer |
| aes | Advanced Encryption Standard (New Instructions) |
| xsave | XRSTOR Save Processor Extended States : also provides XGETBY |
| avx | Advanced Vector Extensions |
| hypervisor | Running on a hypervisor |
| lahf_lm | Load AH from Flags (LAHF) and Store AH into Flags (SAHF) in long mode |

Depending on the configuration of the execution machine, some other flags may be enabled too:

| flag | signification |
|---:|---|
| vme | Virtual 8086 mode enhancements |
| rdtscp | Read Time-Stamp Counter and Processor ID |
| constant_tsc | TSC ticks at a constant rate |
| rep_good | rep microcode works well |
| eagerfpu | Non lazy FPU restore |
| arat | Always Running APIC Timer |
| xsaveopt | Optimized XSAVE |
| xtopology | cpu topology enum extensions |
| vmmcall | prefer VMMCALL to VMCALL |

**IMPORTANT:** if non-compatible CPU-based optimizations are enabled in the provided executable files, this may lead to crashes (and bad results for the concerned tools). In that case, organizers of the MCC decline any responsibility in the crash-leading cause of the concerned tool.

**E-5.3.** For sequential tools, the virtual machine characteristics are defined as follows:

- 16 GB of memory,
- 1 core.

**E-5.4.** For concurrent tools, the virtual machine characteristics are defined as follows:

- 16 GB of memory,
- 4 cores.

**E-5.5.** To avoid bias due to the interaction between the way the operating system handles physical resources on the host machine and the virtual machine embedding the tool, we pin each virtual machine to either 1 physical core (sequential tools) or 4 physical cores (parallel tools)[6].

[6] This decision was taken based on a discussion between tool developers and the MCC organizers from June to October 2017.

**E-5.6.** A prototype VM running Linux is provided by the MCC organizers. It is composed of two disk images:

- `mcc2018.vmdk`, that contains the Operating System, the monitoring system, the interaction scripts for ***BenchKit***, and a location where you must install your tool. Tool developers may update this Disk image,
- `mcc2018-input.vmdk`, that contains the models. It is mounted in read-only mode at `/Home/mcc/BenchKit/INPUTS`. The proposed one will contain updated data based on the MCC'2017 edition of the Model Checking Contest. It will be replaced by the one containing models and formulas for 2018 after the qualification phase. It will be published as a standalone file with the results so that track of these models are easier to extract and exploit.

## IV Miscellaneous Provisions

**P-1.** In case of technical problems or litigious situations, participants should first discuss with the organizers of the Call for Models, or Call for Tools, respectively.

**P-2.** Any issue that cannot be solved by discussing with the organizers (call for models, call for tools) should be brought before Didier Buchs, who is one of the two General Chairs.

**P-3.** Withdrawal of a participating tool by the organizers at any moment is possible in case of not respecting the present rules.

**P-4.** A participating tool can resign at any moment, it will be completely removed from the record of the contest.

**P-5.** The Model Checking Contest is governed by French laws. Only the justice courts of Paris (France) are legally competent for settling disputes related to this contest.