Verification of Token-Scaling Models using an Under-Approximation

Torsten Liebke and Karsten Wolf

Universität Rostock, Institut für Informatik, Germany
{torsten.liebke,karsten.wolf}@uni-rostock.de

Abstract. In the model checking domain the state explosion problem is the core issue. The cause is usually the sheer size of the model or the cardinality of tokens in the initial state. For the latter, which we call *token-scaling models*, we propose an *under-approximation* for reachable states. The idea is to reduce the number of tokens in the initial state and thus reducing the state space. If in the reduced state space a witness path is found, then the witness path can also be executed in the original state space. This method preserves existential temporal properties (ECTL*) using a *simulation relation* between the reduced and the original state space. Since the cardinality of the initial marking varies from only a few tokens to multi-digit numbers of tokens, we apply heuristics to compute the number of tokens that should be removed. We implemented the new method in the explicit model checker LoLA 2. The experiments, done on the model checking contest benchmark, show that this method can speed up the model checking process and solve additional queries.

Keywords: Model Checking · Under-Approximation · Witness Path.

1 Introduction

State space explosion is the main issue in the model checking domain. The cause in place/transition Petri nets (P/T net) is usually either the model size itself, or the cardinality of the initial marking, i.e., the number of tokens on the initially marked places is large and thus resulting in a large state space. In this paper we are concerned with the latter one. P/T nets that have a large number of tokens on the initially marked places, are usually scaling over the number of tokens in the initial marking, while the net stays the same. We call such nets *token-scaling models.* Token-scaling models are widespread in a lot of different fields. E.g., in biochemistry the tokens represent molecules, or in scheduling problems they represent resources. Such a biochemistry example model would be the Angiogenesis model [2], which is part of the model checking contest (MCC) [1]. It consists of 39 places, 64 transitions and 185 arcs and with scaling parameter 25 it results in over $4.3 \cdot 10^{19}$ reachable markings. One example for a scheduling problem would be the RobotManipulation model [5] from the MCC. The structure is even smaller and consists of only 15 places, 11 transitions and 34 arcs and with scaling parameter 10000 this results in over $2.8 \cdot 10^{33}$ reachable markings.

To combat the state explosion problem on token-scaling models, we introduce an under-approximation for the verification of existentially quantified temporal properties (ECTL*). The idea is to reduce the number of tokens in the initial marking, on places which have more than one token. Due to this the state space is also reduced. If in the reduced state space a witness path is found, then the witness path can also be executed in the original state space. For preserving ECTL* formulas a simulation relation [6] between the original and the reduced transition system needs to be established. The cardinality of the initial marking depends on the model and its scaling parameter, which can vary from only a few to multi-digit numbers of tokens. we apply different heuristics to compute how many tokens should be removed from the initial marking.

The introduced method works with all specifications were a single path, either a witness path or a counterexample, is enough to verify the specification. This means next to positive ECTL* formulas with a witness path, we can verify ACTL* formulas (universal temporal properties) and LTL formulas (linear time logic) which have counterexamples. The downside of the token-scaling verification is, it is only a sufficient condition and can only provide answers if a witness path or a counterexample exists. If this is not the case, the original state space has to be checked. The upside is that this method uses in the end a normal model checking algorithm, meaning it can be combined with other reduction techniques like symmetries [8] or partial order reduction [3, 7, 9].

We implemented the introduced method in our award winning model checking tool LoLA 2 and tested it on the model checking contest (MCC) benchmark. We run the token-scaling verification in parallel to the actual model checking algorithm. The experiments show that the token-scaling method can produce results faster and solve additional queries, which couldn't be solved before due to the large state space.

The paper is organized as follows. We start in Section 2 with a motivational example. In Section 3 we give a brief introduction of the terminology of P/T nets, temporal logic and simulation. Then we introduce our new under-approximation for token-scaling models in Section 4. We continue in Section 5 with the introduction of two different heuristics to increase the performance. It follows an experimentally validation of our new approach in Section 6. And we conclude the work in Section 7.

2 Motivational example

Figure 2 shows the token-scaling P/T net RobotManipulation [5]. In the MCC the scaling parameter $n \in \mathbb{N}$ is between 1 and 10000. The initial marking is based on n. The three marked places *p_ i1*, *access* and *r_ stopped* get the following number of tokens: *p_ i1* $= 2 \cdot n + 1$ and *access* $=$ *r_ stopped* $= 2 \cdot n$. The model

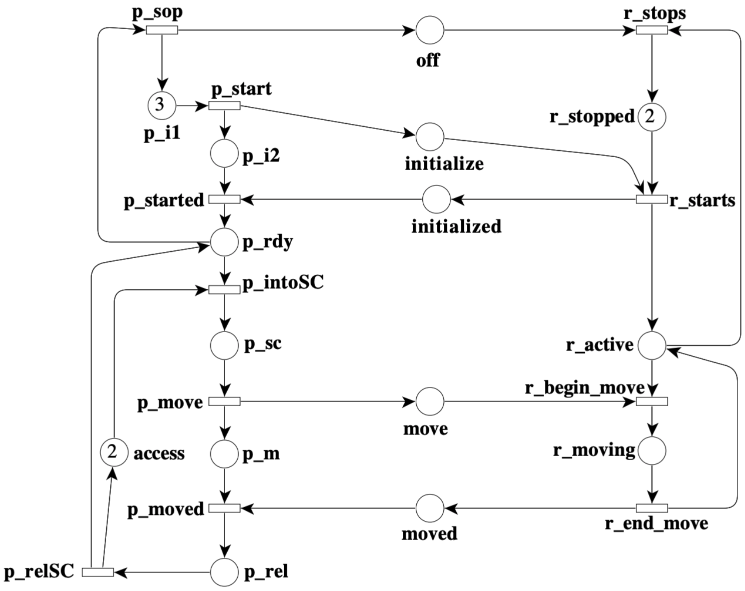

Fig. 1. RobotManipulation [5] model from the MCC with $n = 1$. The number of initial tokens is in place *p_ i1* $= 2 \cdot n + 1$ and in places *access* = *r_ stopped* $= 2 \cdot n$.

consists of 15 places, 11 transitions and 34 arcs. Despite its simple structure, the size of the state space for $n = 10000$ is rather large and results in $2.8 \cdot 10^{33}$ reachable markings.

We discovered that there are quite a few interesting temporal properties, which are not dependent on the actual number of tokens in the initial marking. The idea is now to reduce the number of tokens on the marked places in the initial marking and therefore reducing the state space. If we find a witness path in the model with a reduced initial marking, then we know the witness path can also be executed in the model with the original initial marking. It follows that if the property under investigation holds with the reduced initial marking, then the property also holds with the original initial marking.

For our example model with $n = 5000$ a simplified specification φ from the last edition of the MCC is concerned with the comparison of the cardinality of some places: $\varphi = \mathbf{EF}(p\_rel > p\_m \text{ AND } p\_m > p\_rdy)$, i.e., does a path exists, where in a marking finally it holds, that $(p\_rel > p\_m \text{ AND } p\_m > p\_rdy)$. With $n = 5000$ the places *access* and *r_ stopped* have 10000 tokens and the place *p_ i1* has 10001 tokens in the initial marking. We tested this with our model checking tool LoLA 2 [11]. Without partial order reduction LoLA 2 could not verify this property, even after computing the first one billion states. With partial order reduction enabled a witness path consisting of 195006 transitions was found, while the overall computed reachable markings where 250010. Using our new under-approximation we reduced the number of tokens in the initial marking of all marked places to 5 tokens. With this, LoLA 2 found a witness path consisting of only 112 transitions, while computing only 159 markings in total. This is with and without partial order reduction enabled. We see the possible speed-up, here three orders of magnitude, using the under-approximation.

3 Terminology

Definition 1 (Place/Transition Net). *A* place/transition net*, P/T net for short, consists of a finite set of* places P*, a finite set of* transitions T *where* $P \cap T = \emptyset$*, a set of* arcs $F \subseteq (P \times T) \cup (T \times P)$*, a* weight function $W : (P \times T) \cup (T \times P) \to \mathbb{N}$ *where* $W(x,y) = 0$ *if and only if* $(x,y) \notin F$*, and an* initial marking $m_0 : P \to \mathbb{N}$*, where* marking *is a mapping* $m : P \longrightarrow \mathbb{N}$*.*

The behaviour of a P/T net is defined by the transition rule.

Definition 2 (Transition rule of a P/T net). *Let* $N = [P, T, F, W, m_0]$ *be a P/T net. Transition* $t \in T$ *is* enabled *in marking* m *if, for all* $p \in P$*,* $W(p,t) \leq m(p)$*. If* t *is enabled in* m*,* t *can* fire*, producing a new marking* m' *where, for all* $p \in P$*,* $m'(p) = m(p) - W(p,t) + W(t,p)$*. This firing relation is denoted as* $m \xrightarrow{t} m'$*. It can be extended to a* marking sequence *by the following inductive scheme:* $m \xrightarrow{\varepsilon} m$ *(for the empty sequence* ε*), and* $m \xrightarrow{\omega} m' \wedge m' \xrightarrow{t} m'' \implies m \xrightarrow{\omega t} m''$ *(for a sequence* $\omega \in T^*$ *and a transition* $t \in T$*).*

Using the transition rule, a P/T net induces a transition system, which is also called the *reachability graph* or the *state space* of the P/T net.

Definition 3 (Labeled Transition System, Reachability Graph). *A* transition system *consists of a set* S *of* states*, an initial state* $s_0 \in S$*, and a labelled transition relation* $E : S \times L \times S$*, for some label set* L *of transitions.*
The reachability graph *of a P/T net is a transition system, where the set of markings, transitively reachable from the initial marking* m_0 *using the transition rule of the P/T net, is the set of states, and the transition rule defines the set of vertices.* m_0 *serves as initial state. A marking* m' *is* reachable *from a marking* m *in a P/T net if there is a marking sequence* $\omega \in T^*$ *with* $m \xrightarrow{\omega} m'$*.*

We continue with the introduction of the syntax and semantics of the temporal logic CTL^*. We start with the presentation of atomic propositions.

Definition 4 (Atomic proposition). *Let* $N = [P, T, F, W, m_0]$ *be a P/T net. An* atomic proposition *is one of the constants* true *and* false*, or an expression of the shape* $k_1 p_1 + \dots + k_n p_n \leq k$*, for some* $n \in \mathbb{N}$*,* $k_1, \dots, k_n, k \in \mathbb{Z}$*, and* $p_1, \dots, p_n \in P$*. For a marking* m *of* N*,* m *satisfies proposition* $k_1 p_1 + \dots + k_n p_n \leq k$ *if and only if the term* $\sum_{i=1}^{n} k_i m(p_i)$ *actually evaluates to a number less or equal to* k*. The fact that* m *satisfies atomic proposition* φ *is denoted by* $m \models \varphi$*.*

Definition 5 (Syntax of CTL^*). *For a given set of atomic proposition* AP*, the temporal logic* CTL^* *is inductively defined as follows:*

- *Every $\varphi \in AP$ is a state formula;*
- *If φ and ψ are state formulas, so are $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, and $\neg\varphi$;*
- *Every state formula is a path formula;*
- *If φ and ψ are path formulas, so are $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $\neg\varphi$, $\mathbf{X}\,\varphi$, $\mathbf{F}\,\varphi$, $\mathbf{G}\,\varphi$, $(\varphi\,\mathbf{U}\,\psi)$, and $(\varphi\,\mathbf{R}\,\psi)$;*
- *If φ is a path formula then $\mathbf{E}\,\varphi$ and $\mathbf{A}\,\varphi$ are state formulas.*

We now consider the semantics of the logic CTL* with respect to a Kripke structure. A Kripke structure requires every state to have a successor state. Hence, the reachability graph of a P/T net with deadlocks is not a Kripke structure, but can be turned into one, by adding a silent transition from every deadlock state to itself.

Definition 6 (Semantics of CTL^*). *Let AP be a set of atomic propositions and N a P/T net with its reachability graph TS extended to a Kripke structure. The semantics of CTL^* is defined by satisfaction relations of a state formula φ at a state (i.e., a marking) s, denoted $s \models \varphi$, and of a path formula φ by a path π, denoted $\pi \models \varphi$.*

- *for $\varphi \in AP$, let $s \models \varphi$ according to Def. 4;*
- *for a state formula φ, $\pi \models \varphi$ if $\pi = s_1 s_2 \ldots$ and $s_1 \models \varphi$;*
- *$\pi \models (\varphi \wedge \psi)$ if $\pi \models \varphi$ and $\pi \models \psi$; $s \models (\varphi \wedge \psi)$ if $s \models \varphi$ and $s \models \psi$;*
- *$\pi \models \neg\varphi$ if $\pi \not\models \varphi$; $s \models \neg\varphi$ if $s \not\models \varphi$;*
- *$\pi \models \mathbf{X}\,\varphi$ for $\pi = s_1 s_2 \ldots$ if $\pi' = s_2 s_3 \ldots$ and $\pi' \models \varphi$;*
- *$\pi \models (\varphi\,\mathbf{U}\,\psi)$ for $\pi = s_1 s_2 \ldots$ if, for some $i \geq 1$ and $\pi_i = s_i s_{i+1} \ldots$, $\pi_i \models \psi$, and for all j $(1 \leq j < i)$, $\pi_j = s_j s_{j+1}$ and $\pi_j \models \varphi$;*
- *$s \models \mathbf{E}\,\varphi$ if, for some path π starting in s, $\pi \models \varphi$.*

Let further $\varphi \vee \psi$ be equivalent to $\neg(\neg\varphi \wedge \neg\psi)$, $\mathbf{F}\,\varphi$ to true $\mathbf{U}\,\varphi$, *$\mathbf{G}\,\varphi$ to $\neg\,\mathbf{F}\,\neg\varphi$, $\varphi\,\mathbf{R}\,\psi$ to $\neg(\neg\varphi\,\mathbf{U}\,\neg\psi)$, and $\mathbf{A}\,\varphi$ to $\neg\,\mathbf{E}\,\neg\varphi$.*

A Kripke structure satisfies a state formula if its initial state does. It satisfies a path formula if all paths starting in the initial state do. For CTL^*, several fragments are frequently studied.

Definition 7 (Fragments of CTL^*). *CTL^* formula φ is in*

- *ACTL* if φ does neither contain $\mathbf{E}$ nor $\neg$;*
- *ECTL* if φ does neither contain $\mathbf{A}$ nor $\neg$;*
- *CTL if every occurrence of $\mathbf{X}, \mathbf{F}, \mathbf{G}, \mathbf{R}, \mathbf{U}$ is immediately preceded by an occurrence of $\mathbf{A}$ or $\mathbf{E}$;*
- *LTL $\subset$ ACTL* if φ does neither contain $\mathbf{E}$ nor $\mathbf{A}$;*

And since we have all the equivalences we need to push negations to the level of atomic propositions. Further we know that ECTL* properties are preserved if transition systems are related by a *simulation* relation.

Definition 8 (Abstraction, Simulation [6]). *Let $TS_1 = [S_1, E_1, s_{01}]$ and $TS_2 = [S_2, E_2, s_{02}]$ be Kripke structures. A relation $\sigma \subseteq S_1 \times S_2$ is an* abstraction relation *if, for all atomic propositions φ, $(s, s') \in \sigma$ and $s \models \varphi$ implies $s' \models \varphi$. Abstraction relation σ is a* simulation *if $(s_{01}, s_{02}) \in \sigma$ and, for all $(s_1, s_2) \in \sigma$ and $(s_1, t, s_1') \in E_1$ implies the existence of a state s_2' where (s_2, t', s_2') for some label t', and $(s_1', s_2') \in E_2$.*

Simulation preserves certain temporal logic fragments.

Proposition 1 (Simulation preserves ACTL*, [4]). *Let $TS_1 = [S_1, E_1, s_{01}]$ and $TS_2 = [S_2, E_2, s_{02}]$ be Kripke structures. If TS_1 simulates TS_2, then*

- *TS_1, $s_{01} \models \varphi$ implies $TS_2, s_{02} \models \varphi$, for any ACTL* formula φ;*
- *TS_2, $s_{02} \models \varphi$ implies $TS_1, s_{01} \models \varphi$, for any ECTL* formula φ.*

Using this, we can find a counterexample for ACTL* using an equivalent ECTL* formula.

Proposition 2 (Counterexample for ACTL*). *For every ACTL* formula φ there exists an ECTL* formula ψ s.t. $\neg\varphi$ and ψ are equivalent CTL* formulas.*

4 Under-approximation for token-scaling models

For token-scaling models we propose an under-approximation to reduce the size of the state space. The idea is to reduce the number of tokens on certain places in the initial marking. I.e., the state space is reduced, since fewer tokens are available. The basic concept is to use a threshold of tokens $\lambda \in \mathbb{N}$. The number of tokens on places that contain more than λ tokens in the initial marking, will be reduced to λ tokens.

Definition 9 (Reduced initial marking m_{0r}). *Let $N = [P, T, F, W, m_0]$ be a P/T net and $\lambda \in \mathbb{N}$. In the* reduced initial marking *m_{0r} it holds for all $p \in P$ that $m_{0r}(p) = \lambda$ if, $m_0(p) \geq \lambda$ or $m_{0r} = m_0(p)$ else.*

We substitute the initial marking m_0 of the given P/T net N with the reduced initial marking m_{0r} and verify the property under investigation on the reduced net.

Definition 10 (Reduced net N_r). *Let $N = [P, T, F, W, m_0]$ be a P/T net. We call $N_r = N[m_0 \mapsto m_{0r}]$ the* reduced net*, where the initial marking m_0 is substituted with the reduced initial marking m_{0r}.*

With the reduced net, we can now introduce the under-approximation for token-scaling models.

Theorem 1 (N simulates N_r). *Let $N = [P, T, F, W, m_0]$ be a P/T net, with the induced transition system $TS = [S, E, s_0]$ and $N_r = [P, T, F, W, m_{0r}]$ the corresponding reduced net, with the induced transition system $TS_r = [S_r, E_r, s_{0r}]$. It holds that TS simulates TS_r and that $TS_r, s_{0r} \models \varphi$ implies $TS, s_0 \models \varphi$, for any ECTL* formula φ.*

Proof. The existence of the simulation, together with proposition 1 preserve ECTL*. Since the new system TS_r is an under-approximation, it holds that the original system TS is relative to the reduced system TS_r an over-approximation. For over-approximation, it is well known (proposition 1), that the simulation preserves ACTL*. And with the inversion, which is the under-approximation, the simulation preserves ECTL*. □

This approach is able to verify temporal properties in a transition system which have a witness path or a counterexample.

Proposition 3. *For every ECTL* formula φ, it holds, if φ is true in the reduced net N_r then φ is also true in the original net N. For every ACTL* formula φ and with this also for every LTL formula φ, it holds, if φ is false in the reduced net N_r then φ is also false in the original net N.*

5 Heuristics

The *reduced initial marking* has to balance between two objectives. On one hand a small state space is desired and therefore initially marked places should have as few tokens as possible. On the other hand it should have enough tokens to verify the property under investigation. The optimum would be that initially marked places have exactly the number of tokens on it, which are needed to produce the witness path. Since we don't know this number in advance, we propose two heuristics, to compute the *reduced initial marking*. We calculate a threshold for the marked places in the initial marking and cut the number of tokens on these places, to the calculated threshold.

Largest constant heuristic: Given is a multiplier $n \in \mathbb{N}$, we compute the threshold based on the n-times largest constant appearing in the net, meaning the arc weight and the formula.

Percentage heuristic: Given is a divisor $d \in \mathbb{N}$, the threshold for each place is $1/d$-times the original number of tokens. To avoid getting zero tokens on places, we always round up the division.

6 Experimental validation

We implemented the introduced method in our explicit model checker LoLA 2 [11]. For evaluating the methods, we used the benchmark provided by the model checking contest 2019 [1]. The benchmark consists of 94 models, which result in 1018 model-instances due to the scaling parameter. As specification we use the formulas provided in the MCC 2019. Although the introduced method works also for LTL and a lot of CTL formulas, we only present the experimental results for reachability queries. The results for LTL and CTL are similar. For each instance, there are 32 reachability formulas: 16 of them are concerned with the cardinality of tokens on places and the other 16 are concerned with the fireability of transitions.

For our purpose we only choose a subset of the models from the benchmark. We consider only P/T nets, since coloured nets usually scaling over the cardinality of places, transitions and arcs. In the MCC coloured nets are mostly safe and if they are not safe they contain only a few tokens on each place. This is also true for their place/transition counterparts, which we therefore also ignore. We also only consider nets which are not safe. Furthermore, we consider only nets, whose initial marking has tokens that can be removed. We also avoid one instance that has in the initial marking, on at least one place, more than 2^{32} tokens. All in all we consider 21 models with 214 instances with 32 formulas for each instance. The total number of checked formulas are 6848.

We executed the experiments on our machine Ebro, which has been used for executing parts of the actual MCC in recent years. It has 32 physical cores running at 2.7 GHz and 1 TB of RAM. The operation system used is CentOS Linux 7 (Core). We run our new method in parallel to the actual state space search and a structural method, namely a *counterexample guided abstraction refinement* (CEGAR) state equation [10]. Both methods are well developed and already pushed to the limits. From 6848 formulas 5152 formulas were solvable with a witness path or counterexample. The remaining formulas were either solvable directly in the initial marking or the entire state space had to be searched. There were 122 formulas that could not be solved. Our new method, with the threshold set to 5 tokens, and run in parallel to the other two methods, was able to solve 436 (8.5 %) of the remaining 5152 queries. No additional query from the 122 unsolved ones could be solved. The picture stays more or less the same if we change the threshold to anything between 1 and 10. The largest constant heuristic does not perform very well with multiplier 1 or 2, although increasing the multiplier is improving the performance. The percentage heuristic, on the other hand, performs really well with n set to anything between 3 and 10. With $n = 3$, the new method could solve 659 (12.8 %) queries. It was even able to solve 8 (6.5 %) queries from the 122 unsolved ones.

7 Conclusion

Token-scaling P/T nets tend to have large state spaces, since they have a lot of tokens on the initially marked places. However, there are a lot of interesting specifications, which can be verified with way fewer tokens in the initial marking. We introduced an under-approximation for token-scaling P/T nets. It is a sufficient quick check, that is based on the reduction of tokens on the marked places in the initial marking. The method is applicable whenever a witness path can be found, namely in the class of ECTL* formulas, including reachability, or whenever a counterexample can be found, namely in the class of ACTL* and LTL formulas. The experiments show that running the token-scaling method in parallel to a full model checking algorithm speeds up the verification process, which gives in reverse more time to other verification queries. Our new method could solve 9.6 % of all queries in the benchmark, while competing directly against well established methods like the actual state space search and the

CEGAR state equation method. An additional 6.5 % of the queries that could not be solved by any other method were solved by the new method. We also showed that using a good heuristic can increase the performance even more. In the future we are interested in better heuristics using the structural information of the P/T net and involving the formula.

References

1. Fabrice Kordon et al. Presentation of the 9th edition of the model checking contest. In *Proc. TACAS, LNCS 11429*, pages 50–68, 2019.
2. Lucia Napione et al. On the use of stochastic Petri nets in the analysis of signal transduction pathways for angiogenesis process. In *Proc. CMSB, LNCS 5688*, pages 281–295, 2009.
3. P. Godefroid and P. Wolper. A partial approach to model checking. *Inf. Comput.*, 110(2):305–326, 1994.
4. Orna Grumberg and David E. Long. Model checking and modular verification. In *Proc. CONCUR, LNCS 527*, pages 250–265, 1991.
5. F. Kordon. RobotManipulation. https://mcc.lip6.fr/pdf/RobotManipulation-form.pdf, 2017. Accessed: 2020-03-01.
6. R. Milner. *Communication and Concurrency*. Prentice Hall international series in computer science. Prentice Hall, New York, 1989.
7. D. A. Peled. All from one, one for all: on model checking using representatives. In *Proc. CAV, LNCS 697*, pages 409–423, 1993.
8. Karsten Schmidt. How to calculate symmetries of petri nets. *Acta Inf.*, 36(7):545–590, 2000.
9. A. Valmari. Stubborn sets for reduced state space generation. In *Advances in Petri Nets, LNCS 483*, pages 491–515, 1989.
10. H. Wimmel and K. Wolf. Applying CEGAR to the Petri net state equation. *Logical Methods in Computer Science*, 8(3), 2012.
11. K. Wolf. Petri net model checking with LoLA 2. In *Proc. PETRI NETS, LNCS 10877*, pages 351–362, 2018.