Solving E (ϕ U ψ) using the CEGAR Approach

Torsten Liebke and Karsten Wolf

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

Abstract. Petri nets are an established formal method for modelling and verifying asynchronous, concurrent and distributed systems. To verify a specification, given as a temporal logic formula, state space methods often encounter the state space explosion problem. We propose a verification technique to solve the CTL query E (ϕ U ψ) using the *Petri net state equation* with a *counterexample guided abstraction refinement* (CEGAR) approach. The algorithm tries to solve EF ψ, while keeping ϕ true. Especially in case the property does not hold, the technique often terminates quickly. As a side product we show that $(\text{EX})^k\phi$ formulas can be solved with the CEGAR approach as well.

Keywords: Petri Nets, Verification, Structural Analysis, CEGAR, ILP.

1 Introduction

Explicit model checking algorithms encounter the state space explosion problem. A different concept to verify the reachability problem was introduced in [7] and extended by [4, 3]. This concept is based on the structure of Petri nets and decreases the state space explosion problem significantly. It transforms the problem to an integer linear programming (ILP) problem, which runs iteratively based on counterexample guided abstraction refinement, proposed in [1].

Due to the fact that ILP problems can become infeasible, the CEGAR approach is especially good to verify negative results. This makes it a valuable complement to explicit model checking algorithms, which are in general good for verifying positive results, due to the on-the-fly effect.

In [5] it is shown that it is beneficial to use specialised routines for common formulas to increase the number of verifiable problems. We propose two techniques to solve the CTL queries E(ϕ U ψ) and $(\text{EX})^k\phi$ with the CEGAR approach for Petri nets. Using well known tautologies, also A(ϕ R ψ) and $(\text{AX})^k\phi$ are solvable with these techniques. [5] also shows that only 62.3 % of the E(ϕ U ψ)/A(ϕ R ψ) formulas from the model checking contest 2018 [2] are solved using the explicit model checker LoLA 2 [8]. This is due to the reason that the on-the-fly effect has no or very limited impact in some cases, e.g. when $\phi \wedge \neg\psi$ holds in the entire state space. For this case, the CEGAR approach we are introducing will terminate very quickly, stating that the ILP problem is infeasible.

One drawback is that termination of the introduced approach is not guaranteed, which makes the procedure incomplete [7]. Applying it in a portfolio approach with traditional algorithms, this drawback vanishes.

2 Basic Definitions

We consider place/transition Petri nets.

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

Transition t is enabled *in marking m if, for all $p \in P$, $m(p) \geq W([p, t])$. Firing an enabled transition in m yields the marking m' where, for all p, $m'(p) = m(p) - W([p, t]) + W([t, p])$. This is denoted $m \xrightarrow{t} m'$.*

Every Petri net defines a labeled transition system where the set of markings reachable from m_0 form the set of states, m_0 is the initial state, and the firing relation just defined forms the labeled transition relation. We restrict our considerations to Petri nets where the related transition system is finite.

The *incidence matrix* of a Petri net N is a matrix $C_N : P \times T \longrightarrow \mathbb{Z}$ where, for all $p \in P, t \in T$, $C_N(p, t) = W(t, p) - W(p, t)$. The incidence matrix is involved in important and well-known results of Petri net theory.

Definition 2 (Reachability problem). *Given is a tuple (N, m, m') consisting of a Petri net N and two markings m, m'. A marking m' is* reachable *from marking m in a Petri net N, if there exists a firing sequence $w \in T^*$ with $m \xrightarrow{w} m'$. The set of all reachable markings in N starting in m is written as $R_N(m)$. The question whether $m' \in R_N(m)$ is called the* reachability problem*.*

The feasibility of the Petri net state equation is a necessary condition for a positive answer to this question.

Proposition 1 (Petri net state equation). *Let $w \in T^*$ be a firing sequence of N, that is, the sequence of labels on a path from some marking m to a marking m' in the transition system corresponding to N. Then it holds*

$$m + C_N \cdot \wp(w) = m'$$

where $\wp(w)$ is a vector where $\wp(w)|t|$ is the number of occurrences of t in the sequence w.

In the sequel, we shall refer to $\wp(w)$ as the *Parikh vector* of w.

Definition 3 (T-invariant). *A Parikh vector $\wp(w)$ is called a T-*invariant *if $C_N \cdot \wp(w) = 0$. If the firing sequence w is executable, we call $\wp(w)$ realizable.*

A realizable T-invariant is a cycle in the state space and will not change the marking.

Definition 4 (Solution space). *The solution of the Petri net state equation $m + C_N \cdot \wp(w) = m'$ can be written as the sum of a base solution and a period vector, which is a linear combination of T-invariants: $\wp(w) = b + \sum_i n_i y_i$, where $b \in \mathbb{N}^T$ is the base solution and $n_i \in \mathbb{N}$ is the coefficient of the T-invariant $y_i \in \mathbb{N}^T$ [3, 7].*

3 Increasing and Decreasing Transitions

Consider a formal sum $s = k_1p_1 + \cdots + k_np_n$. Every marking m turns this sum into the integer number $v_s(m) = k_1m(p_1) + \cdots + k_nm(p_n)$. We can immediately derive from the firing rule of Petri nets:

Definition 5 (Delta). *Let s be a formal sum and t a transition, then $\Delta_{t,s}$ is defined as $\Delta_{t,s} = k_1C_N(p_1, t) + \cdots + k_nC_N(p_n, t)$.*

Lemma 1. *For all markings m, $m \xrightarrow{t} m'$ implies $v_s(m) + \Delta_{t,s} = v_s(m')$.*

Proof. Apply the Petri net state equation. □

As we assume the transition system to be finite, there is only a finite range of values that $v_s(m)$ can take. Call an integer number k a *lower bound* for formal sum s if, for any reachable marking m, $v_s(m) \geq k$, and *upper bound* for s if, for any reachable m, $v_s(m) \leq k$. There exist several approaches in Petri net theory for computing bounds. As an example, we can solve the following optimisation problem where s is the objective function (to be minimised or maximised) and the state equation serves as side condition. If the problem yields a solution with non-diverging value for the objective function, that value is a lower (resp. upper) bound for s.

Based on Lemma 1, we can identify increasing and decreasing transitions.

Definition 6 (Increasing, decreasing). *Given an atomic proposition of the form $s \leq k$. Let L be a lower bound and U an upper bound for s. We call transition t w.r.t. the formal sum s:*

1. weakly increasing *iff $\Delta_{t,s} < 0$*
2. weakly decreasing *iff $\Delta_{t,s} > 0$*
3. strongly increasing *iff there is an upper bound U for s where $\Delta_{t,s} \leq k - U$*
4. strongly decreasing *iff there is a lower bound L for s where $\Delta_{t,s} > k - L$.*

The terminology may sound strange at first glance. However, increasing transitions have the tendency to turn a false proposition into a true one while decreasing transitions help turning a true proposition into a false one.

Let $p \leq 0$ be an atomic proposition where p is the number of tokens on place p in a Petri net. Then all transitions in the preset of p are strongly decreasing.

Lemma 2. *Consider markings m and m', transition t with $m \xrightarrow{t} m'$ and atomic proposition $s \leq k$.*

1. *If $s \leq k$ is false in m and true in m' then t is weakly increasing w.r.t. s.*
2. *If $s \leq k$ is true in m and false in m' then t is weakly decreasing w.r.t. s.*
3. *If t is strongly increasing w.r.t. $s \leq k$ then $s \leq k$ is true in m'.*
4. *If t is strongly decreasing w.r.t. $s \leq k$ then $s \leq k$ is false in m'.*

Proof. Regarding 1, we have $v_s(m) > k$ and $v_s(m') \leq k$. By Lemma 1, we conclude $\Delta_{t,s} < 0$. Regarding 3, we have $v_s(m) \geq L$ (since L is a lower bound). Hence, $v_s(m') = v_s(m) + \Delta_{t,s} \leq L + \Delta_{t,s}$ and, according to Def. 6, $v_s(m') \leq k$. □

4 CEGAR approach for reachability analysis in Petri nets

Abstraction is a powerful method for verifying systems. It omits irrelevant details of the system behaviours, to simplify the analysis and verification. Finding the right abstraction is hard. If it is too coarse, the verification might fail and if it is too fine, the state space explosion problem might occur. A solution is to use some initial abstraction [1], which is an overapproximation of the original system and then iteratively refine the abstraction based on spurious counterexamples.

In our case, the Petri net state equation is the initial abstraction for the reachability problem. Solving the state equation is a non-negative integer linear programming problem. The objective function for the ILP problem is the shortest firing sequence of the Parikh vector $f(w) = \sum_{t \in T} \wp(w)|t|$ leading from the initial marking m to the final marking m'.

The feasibility of this linear system is a necessary condition for reachability, but not a sufficient one. We distinguish between three different situations:

- If the linear system is infeasible, the necessary condition is violated and the final marking is not reachable.
- If the linear system has a realizable solution, then the final marking is reachable.
- If the linear system has an unrealizable solution, which is a counterexample, then the abstraction has to be refined.

If we have an unrealizable solution, then there exists at least one $t \in T$ which fired less than $\wp(w)|t|$ times. To produces a new solution which avoids the spurious one, we build a refined abstraction using inequalities for the ILP problem.

Definition 7 (Constraints). *We define two types of constraints, both being linear inequalities over transitions* [7].

- *Jump constraints have the form* $|t_i| < n$, *with* $n \in \mathbb{N}$ *and* $t_i \in T$ *where* $|t_i|$ *represents the firing count of transition* t. *Using the fact that base solutions are pairwise incomparable, jump constraints intend to generate a new base solution.*
- *Increment constraints have the form* $\sum_{i=1}^{k} n_i|t_i| \geq n$ *with* $n_i \in \mathbb{Z}$, $n \in \mathbb{N}$, *and* $t_i \in T$. *Increment constraints are used to get a new non-base solution, i.e., T-invariants are added, since their interleaving with another sequence* w *may turn* w *from unrealizable to realizable.*

Adding the two types of constrains to existing solutions we can traverse through the solution space and check whether the unrealizable solution of our linear system becomes realizable or whether the ILP problem becomes infeasible.

Definition 8 (Partial solutions). *Let* $N = (P, T, F, W, m)$ *be a Petri net and* $m' \in R_N(m)$ *a reachability problem. A* partial solution *is a tuple* $ps = (\Gamma, \wp(w), \sigma, r)$ *with:*

- *Γ is the set of jump and increment constraints. Together with the state equation they form the ILP problem.*
- *$\wp(w)$ is the minimal solution fulfilling the ILP problem.*
- *σ is a firing sequence with $m \xrightarrow{\sigma}$ and $\wp(\sigma) \leq \wp(w)$.*
- *r is the remainder with $r = \wp(w) - \wp(\sigma)$ and $\forall t \in T : (r(t) > 0 \Longrightarrow \neg m \xrightarrow{\sigma t})$.*

Partial solutions are produced during the examination of the solution $\wp(w)$ of the ILP problem by exploring the state space of N. For this an explicit model checking algorithm with reachability preserving stubborn sets [6] can be used to build a tree of reachable markings, such that for all transitions $t \in T$ it holds that they only occur $\wp(w)|t|$ times. Each path to a leaf represents a maximal firing sequence of a new partial solution. If a partial solution has an empty remainder $r = 0$, it is a full solution and the reachability problem is satisfied. If no full solution exists, $\wp(w)$ might be realizable by another firing sequence σ', or by adding a jump constraint to get to a new base solution, or by adding an increment constraint to get additional tokens for transitions with $r(t) > 0$. If all possible partial solutions are explored and no full solution is found, the reachability problem can not be satisfied.

Theorem 1 (Reachability of solutions). *If the reachability problem has a solution, a realizable solution of the state equation can be reached by constantly appending the minimal solution with constraints* [7].

5 Solving E (ϕ U ψ) with the CEGAR approach

Definition 9 (E(ϕ U ψ)). *Let $N = (P, T, F, W, m)$ be a Petri net and ϕ and ψ two propositions. $m \models \mathrm{E}(\phi \mathrm{~U~} \psi) \iff \exists w \in T^* : m \xrightarrow{w} m'$, with $\exists i \in \mathbb{N}\ \forall j < i : (m_j \models \phi) \wedge (m_i \models \psi)$. Which means that in every state along path w, ϕ is true until a state is reached where ψ is true.*

It is well known that EF ψ can be rewritten as E (true U ψ). To solve E(ϕ U ψ), where ϕ and ψ are atomic propositions, we solve EF ψ with the CEGAR approach. In addition to this we introduce additional (balance) constraints to keep ϕ true along the path. Furthermore we cut-off paths in the exploration of partial solutions, whenever states are reached where both ϕ and ψ are false.

Definition 10 (Balance constraints). *Given a Petri net $N = (P, T, F, W, m)$ and an atomic proposition $\phi = s_0 \leq k_0 \wedge s_1 \leq k_1 \wedge \cdots \wedge s_n \leq k_n$, where s_i is a formal sum, $0 \leq i \leq n$ and $i, k, n \in \mathbb{N}$. $T_{s_i} = \{t \in T | \Delta_{t,s_i} \neq 0\}$ is the set of transitions which can change the value of s_i. It contains all weakly/strongly increasing/decreasing transitions w.r.t. to s_i. We call $T_{s_i,\psi} \subseteq T_{s_i}$ the set of decreasing transitions w.r.t s_i, which are at the same time increasing w.r.t ψ: $T_{s_i,\psi} = \{t \in T_{s_i} | \Delta_{t,s_i} > 0 \wedge \Delta_{t,\phi} < 0\}$. We define variables δ_i, which are 0, if $T_{s_i,\psi} = \emptyset$ and otherwise are $MAX(\Delta_{t,s_i} | t \in T_{s_i,\psi})$. The δ_i-offset is the maximum arc weight of all transitions that can change the value of $s_i \leq k_i$ from true to false and ψ from false to true. Let $\theta_i = k_i - v_{s_i}(m)$ be the offset, which is the* *number of tokens that can be consumed from the initial marking and still leave the truth value of* $s_i \leq k_i$ *unchanged. We call* $\forall s_i : \sum_{t \in T_s} \Delta_{t,s} \leq \theta_i + \delta_i$ balance constraints *w.r.t.* s_i *and* m.

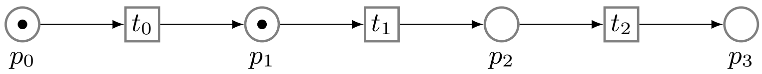

Fig. 1. The minimal solution for this Petri net and the formula E $(p_1 > 0)$ U $(p_3 > 0)$ is $t_1 t_2$. Since t_1 is weakly decreasing w.r.t. $p_1 > 0$, the balance constraint adds the weakly increasing transition t_0 to the solution.

As an example, consider Figure 1 and the formula E $(p_1 > 0)$ U $(p_3 > 0)$. Note that this formula and every other formula can be rewritten into the required $s \leq k$-format: E $(-p_1 \leq -1)$ U $(-p_3 \leq -1)$. To satisfy the formula, we check EF $p_3 > 0$, while keeping $p_1 > 0$ true along the path. The minimal solution to the ILP would be the firing vector (t_1, t_2), $m \xrightarrow{t_1 t_2} m'$, where m' satisfies $p_3 > 0$. But after firing the weakly decreasing transition t_1 w.r.t. $p_1 > 0$, a marking $m'' = (p_0, p_2)$ is reached that does neither satisfy $p_3 > 0$ nor $p_1 > 0$. To avoid this marking, the balance constraint would add the weakly increasing transition t_0 to the solution vector, $m \xrightarrow{t_0 t_1 t_2} m'$, to keep $p_1 > 0$ true.

Balance constraints in general ensure that the sum of all increasing and decreasing transitions w.r.t. a formal sum s is smaller than the offset, which is based on the initial marking and the maximal arc weight of all transitions $t \in T_{s_i,\psi}$. In case the offset θ_i is negative, ϕ is violated and E$(\phi$ U $\psi)$ has the value of ψ. We detect this case in the initial marking, before we compute the balance constraints and can return with a definitive answer directly in the beginning. Balance constraints make sure that ϕ is not violated and ψ is true in the final marking. The only transitions which are allowed to violate ϕ are in the set $T_{s_i,\psi}$ and they have also the effect to turn ψ to true. Due this effect, if such transitions exist, they tend to occur at the end of the firing sequence, but not exclusively. We add the balance constraints to our initial abstraction, the state equation and run the CEGAR algorithm for EF ψ.

Lemma 3. *Given a Petri net* $N = (P, T, F, W, m)$ *and formula* $\phi = s_0 \leq k_0 \wedge s_1 \leq k_1 \wedge \cdots \wedge s_n \leq k_n$*, where* s_i *is a formal sum and* $k \in \mathbb{N}$ *and* $m \models \phi$*. Adding to the ILP problem all balance constraints for* ϕ *and checking that* $\theta_i \geq 0$*, then it is guaranteed that after executing the entire firing sequence given as a solution* $\wp(w)$ *to the ILP problem that* ψ *is true. It also ensures that if a complete firing sequence exists,* ϕ *is true along the path and is only violated, if at all, in the final marking, where* ψ *holds.*

Proof. Regarding the second claim, we know, based on Definition 6, that only increasing/decreasing transitions affect $s_i \leq k_i$. The offset θ_i ensures that the truth value of $s_i \leq k_i$ stays unchanged. The balance constraint ensures that ϕ is not violated minus the δ_i-offset, which ensures the possibility of a firing sequence which does not violate ϕ along the path, until ψ holds.

If the set $T_{s_i,\psi}$ is not empty, the δ_i-offset based on the maximum of Δ_{t,s_i} ensures that transitions are not ignored in the balance constraint that violate ϕ but also turn ψ to true. The additional offset, which is the maximal arc weight of the transitions in the set, is enough to make sure that only one transition is allowed to fire, with the effect of making ϕ false and ψ true. We use the maximum, since an arc weight, which is not the maximum, will have a smaller effect and will not change the outcome. Transitions from the set $T_{s_i,\psi}$ can also fire, if they are in a different context, i.e. when they do not turn ϕ to false.

Theorem 1 ensures that if the complete solution $\wp(w)$ is fired, we get to the final marking m' which satisfies ψ. □

Lemma 3 only ensures that $m' \models \psi$, where m' is the final marking after firing the entire solution $\wp(w)$. But it does not guarantee that intermediate markings satisfy ϕ. This is due to the fact that also decreasing transitions w.r.t. ϕ are allowed to fire.

Lemma 4. *In the exploration of the solution space cutting off paths in markings m^*, with $m^* \models \neg\phi\wedge\neg\psi$ results in keeping only partial solutions which can become full solutions.*

Proof. Based on Definition 9, marking $m^* \models \neg\phi \wedge \neg\psi$ violates the property E(ϕ U ψ). All paths extending m^* are also violating E(ϕ U ψ) and no extension to the path can make the property true. □

Consider, for example, the Petri net in Figure 2 and the formula E ($p_1+p_2 > 0$) U ($p_3 > 0$). The minimal solution to the ILP is (t_0, t_1). After firing t_0, a marking $m' = (p_0, p_5)$ is reached that violates $p_1 + p_2 > 0$ and $p_3 > 0$. Lemma 4 ensures that this solution is cut off. There are also no increasing transitions we can add to this solution. Using the CEGAR approach, we jump to a new base solution, (t_2, t_3). But this solution is only a partial solution due to the fact that neither t_2 nor t_3 can fire. At this point, the CEGAR approach adds the T-invariant (t_4, t_5) from which tokens can be borrowed. Now we have a full solution and we get the path $m \xrightarrow{t_2 t_3 t_4 (t_5)} m'$ which satisfies $p_1 + p_2 > 0$ until $(p_3 > 0)$ is satisfied.

Theorem 2. *Let $N = (P, T, F, W, m)$ be a Petri net and ϕ, ψ atomic propositions, with $\phi = s_0 \leq k_0 \wedge s_1 \leq k_1 \wedge \cdots \wedge s_n \leq k_n$, where s_i is a formal sum and $k, n \in \mathbb{N}$ and it holds that $m \models \phi$. If* E(ϕ U ψ) *has a realizable solution in the solution space, it can be reached by solving* EF ψ *using the* CEGAR *approach from* [7] *and by adding all* balance constraints *to the initial abstraction and* cutting-off *all paths starting in m^* in the exploration of the solution space, whenever such a marking $m^* \models \neg\phi \wedge \neg\psi$ is reached.*

Proof. In [7] EF ψ is proved. We constantly add jump and increment constraints to get to a full solution, such that the final marking of this solution satisfies

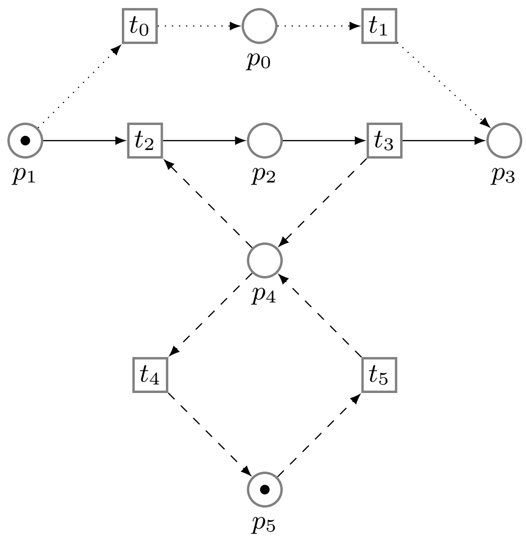

Fig. 2. For the given Petri net and the formula E $(p_1 + p_2 > 0)$ U $(p_3 > 0)$, the minimal solution (t_0, t_1) is cut off. With the CEGAR approach we jump to the next base solution (t_2, t_3), which is only a partial one. The T-invariant (t_4, t_5) is with the next CEGAR step and provides a full solution, $m \xrightarrow{t_2 t_3 t_4 (t_5)} m'$.

$m' \models \psi$. Lemma 3 ensures that we only get solutions, such that after firing the complete solution $\wp(w)$, ϕ holds. Lemma 4 makes sure that ϕ is not violated along the path. $\square$

6 Solving $(\mathrm{EX})^k\phi$ with the CEGAR approach

Definition 11 ($(\mathrm{EX})^k\phi$). *Given a Petri net $N = (P, T, F, W, m)$, a proposition ϕ and $k \in \mathbb{N} \setminus \{0\}$. $m \models (EX)^k\phi \iff \exists w \in T^k \wedge m \xrightarrow{w} m_k \wedge m_k \models \phi$. This means there exists a path $m \xrightarrow{w} m_k$ with $|w| = k$ transitions in it and $m_k \models \phi$.*

For example for $k = 2$ this means $(\mathrm{EX})^2\phi = \mathrm{EX\ EX}\ \phi \iff \exists t_1 t_2 \in T^2$: $m \xrightarrow{t_1 t_2} m_k \wedge m_k \models \phi$. To solve $(\mathrm{EX})^k\phi$, we solve EF ϕ. In addition to this we introduce an additional (length) constraint which ensures that the length of sequence w of the ILP problem solution $\wp(w)$ is equal to k.

Definition 12 (Length constraint). *Given a proposition of the form $(EX)^k\phi$ with $k \in \mathbb{N} \setminus \{0\}$ and an atomic proposition ϕ. We call $\sum_{t \in T} \wp(w)|t| = k$ a* length constraint*.*

The sum of the number of occurrences of all transitions in the Parikh-vector $\wp(w)$ should exactly be k. To make the proposition true, marking m_k, which is reached after firing k transitions, must satisfy ϕ.

Theorem 3. *Given a Petri net $N = (P, T, F, W, m)$ and proposition $(EX)^k\phi$ with $k \in \mathbb{N} \setminus \{0\}$. If* $(\mathrm{EX})^k\phi$ *has a realizable solution in the solution space, it can be reached by solving* EF ϕ *using the CEGAR approach from* [7] *and by adding the* length constraint *to the initial abstraction.*

Proof. Based on Definition 11, $m \models (\mathrm{EX})^k\phi \iff \exists w \in T^k \wedge m \xrightarrow{w} m' \wedge m' \models \phi$. The length constraint $\sum_{t\in T} \wp(w)|t| = k$ from Definition 12 ensures that only solutions $\wp(w)$ of the ILP problem are found, such that the length of the firing sequence is exactly k and results in the final marking $m_k \models \phi$. □

7 Conclusion and future work

We proposed two promising techniques to solve E(ϕ U ψ) and $(\mathrm{EX})^k\phi$ with the CEGAR approach for Petri nets.

To solve E(ϕ U ψ), we solve EF ψ and keep ϕ true in every state along the path. To keep ϕ true, we introduced the concept of balance constraints for the ILP problem to ensure that an atomic proposition is true after firing the entire solution vector. Furthermore we used a cut-off criterion to ensure that ϕ is also true in every state along the path. For solving $(\mathrm{EX})^k\phi$ we introduced the concept of a length constraint, which makes sure that we only get solutions of length k.

These techniques will be implemented in LoLA 2 [8]. LoLA 2 is an explicit model checker and is every year on the podium of the Model Checking Contest (MCC) for Petri nets. Once implemented we expect that the proposed approach will increase the verification performance for this formulas significantly. Especially in case of a negative result, the procedure will terminate quickly, due to the fact that the ILP problem will become infeasible. We expect a similar performance increase as it was the case for the CEGAR approach for reachability analysis, where the performance of LoLA 2 increased from solving under 80 % to over 90 % in the MCC.

References

1. Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement. In *Computer Aided Verification, 12th International Conference, CAV 2000, Chicago, IL, USA, July 15-19, 2000, Proceedings*, pages 154–169, 2000.
2. Fabrice Kordon et al. The model checking contest (2019). To appear in LNCS 11429.
3. Ákos Hajdu, András Vörös, and Tamás Bartha. New search strategies for the Petri net CEGAR approach. In *Application and Theory of Petri Nets and Concurrency - 36th International Conference, PETRI NETS 2015, Brussels, Belgium, June 21-26, 2015, Proceedings*, pages 309–328, 2015.
4. Ákos Hajdu, András Vörös, Tamás Bartha, and Zoltán Mártonka. Extensions to the CEGAR approach on Petri nets. *Acta Cybern.*, 21(3):401–417, 2014.
5. Torsten Liebke and Karsten Wolf. Relieving the CTL model checker in explicit verification. Accepted for Application and Theory of Petri Nets and Concurrency - 40th International Conference, PETRI NETS 2019, Aachen, Germany, June 23-28.

6. Karsten Schmidt. Stubborn sets for standard properties. In *Application and Theory of Petri Nets 1999, 20th International Conference, ICATPN '99, Williamsburg, Virginia, USA, June 21-25, 1999, Proceedings*, pages 46–65, 1999.
7. Harro Wimmel and Karsten Wolf. Applying CEGAR to the Petri net state equation. In *Tools and Algorithms for the Construction and Analysis of Systems - 17th International Conference, TACAS 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26-April 3, 2011. Proceedings*, pages 224–238, 2011.
8. Karsten Wolf. Petri net model checking with LoLA 2. In *Application and Theory of Petri Nets and Concurrency - 39th International Conference, PETRI NETS 2018, Bratislava, Slovakia, June 24-29, 2018, Proceedings*, pages 351–362, 2018.