Short Witnesses and Accepting Lassos in ω-automata[⋆]

Rüdiger Ehlers

Reactive Systems Group
Saarland University

Abstract. Emptiness checking of ω-automata is a fundamental part of the automata-theoretic toolbox and is frequently applied in many applications, most notably verification of reactive systems. In this particular application, the capability to extract accepted words or alternatively accepting runs in case of non-emptiness is particularly useful, as these have a diagnostic value. However, non-optimised such words or runs can become huge, limiting their usability in practice, thus solutions with a small representation should be preferred. In this paper, we review the known results on obtaining these and complete the complexity landscape for all commonly used automaton types. We also prove upper and lower bounds on the approximation hardness of these problems.

1 Introduction

In the last decades, model checking has emerged as an increasingly successful approach to the verification of complex systems [1, 2]. This development is witnessed by the existence of a significant number of industrial-scale model checkers and successful experiments on integrating the usage of model checkers into the development cycle of industrial products [3–5]. Compared to deductive verification approaches, model checking has the advantage of being a push-button technology: the designer of a system only has to state the desired properties and (a model of) the system implementation, but the proof of correctness/incorrectness of the system is done automatically. In case of an error in the implementation, the model checker usually constructs an example run of the system in which this error occurs, which in turn is useful for the system designer to correct the system. It has been observed that this makes model checking particularly useful in the early development stages of a complex system [4, 6], as the automatic generation of such *counter-examples* saves valuable time.

Finding good counter-examples in model checking is however a non-trivial task as the question which of the often infinitely many counter-examples is most useful for the designer heavily depends on the particular problem instance [6, 7].

[⋆] This work was supported by the German Research Foundation (DFG) within the program "Performance Guarantees for Computer Systems" and the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS).

Consequently, the length of a counter-example is the predominant quality metric that researchers in this area have agreed on [8].

As an example, in the context of model checking finite state machines against properties written in *linear time temporal logic* (LTL), the specification is usually negated and transformed into an equivalent non-deterministic Büchi automaton. Finding a witness for the non-satisfaction of the specification then amounts to finding an *accepting lasso* in the product of the finite state machine and the Büchi automaton. In this context, shorter lassos are preferred as they simplify analysing the cause of the problem. Nowadays, efficient polynomial algorithms for finding a shortest accepting lasso in such a setting exist [9, 7], allowing for the extraction of such shortest lassos. On the other hand, for systems that obey some *fairness* constraints, the problem of finding short counter-examples reduces to finding short accepting lassos in *generalized Büchi automata*. For this case, it is known that finding a shortest accepting lasso is NP-complete [10].

Independently of these complexity considerations, it has been argued that for debugging models, a shortest accepting lasso is not what the designer of a system is usually interested in [11]. In fact, some input to the system of the form uv^ω (for u and v being finite sequences) such that $|u| + |v|$ is minimal is likely to be more helpful for debugging as such a representation is independent of the actual automaton-encoding of the violated property. For this modified setting, Kupferman and Sheinfeld-Faragy proved that also for ordinary model checking without fairness, finding shortest such counter-examples (called witnesses in this case) is NP-complete, rendering the problem difficult.

From a more high-level view of these results, the existence of efficient algorithms for some of the cases just discussed on the one hand and the NP-completeness of the other cases leads to a natural question: where exactly is the borderline that separates the hard problems from the simple ones for finding short counter-examples in model checking? Furthermore, from a practical point of view, another question naturally arises: what is the approximation hardness of these problems? For example, while finding a shortest witness for the non-satisfaction of a specification might be NP-hard, finding a 2-approximate shortest witness might be doable in polynomial time. Obviously, such a result would have practical consequences. Nevertheless, to the best of our knowledge, this question has not been discussed in the literature yet.

In this paper, we give a thorough discussion of the complexity of finding short non-emptiness certificates for various types of ω-automata, which answers the question how hard obtaining short counter-examples in regular model checking (which reduces to Büchi automaton emptiness) and model checking under fairness (which reduces to generalized Büchi automaton emptiness) actually is. We discuss both types of certificates mentioned above: short accepting lassos and short witnesses. As finding short lassos and witnesses is also useful in other contexts in which automata-theoretic methods are applied, like *synthesis of closed systems* [12] or deciding the validity of formulas in logics such as S1S [13], we give a unified overview for all commonly used types of ω-automata, namely those with safety, Büchi, co-Büchi, parity, Rabin, Streett, generalized Büchi and Muller acceptance conditions. For all of these cases, we review the known complexity results for the exact minimization of the size of accepting lassos or witnesses and complete the complexity landscape for the cases not considered in the literature so far. This results in the first complete exposition of the borderline between the hard and simple problems in this context. We also examine the approximation hardness of the NP-complete problems of this landscape, which, from a practical point of view, is an important question to raise as approximate solutions often suffice for the good usability of a method in which finding short accepting lassos or witnesses is a sub-step. The results we obtain for the approximability of the problems considered are mostly negative: For example, we prove that approximating the minimal witness size within any polynomial is NP-complete even for the simple safety acceptance condition. We also give some positive results, e.g., a simple algorithm for approximating the minimal witness size within any (fixed) exponential function that runs in time polynomial in the number of states of some given ω-automaton. Table 1 contains a summary of the other results contained in this paper.

The structure of our presentation is as follows: In the next section, we state the preliminaries. Sections 3 and 4 contain the precise definitions of the problems of finding shortest accepting lassos and witnesses and present hardness results and algorithms for them. Section 5 concludes the findings and sketches the open problems.

2 Preliminaries

An ω-automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ is a five-tuple consisting of some finite state set Q, some finite alphabet Σ, some initial state $q_0 \in Q$, some transition function $\delta : Q \times \Sigma \rightarrow 2^Q$ and some *acceptance component* $\mathcal{F}$ (to be defined later). We say that an automaton is deterministic if for every $q \in Q$ and $x \in \Sigma$, $|\delta(q, x)| \leq 1$.

Given an infinite word $w = w_1 w_2 \ldots \in \Sigma^\omega$ and an ω-automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$, we say that some sequence $\pi = \pi_0 \pi_1 \ldots$ is a run for w if $\pi_0 = q_0$ and for all $i \in \{1, 2, \ldots\}$, $\pi_i \in \delta(\pi_{i-1}, w_i)$. We say that π is accepting if for $\inf(\pi) = \{q \in Q \mid \exists^\infty j \in \mathbb{N} : \pi_j = q\}$, $\inf(\pi)$ is accepted by $\mathcal{F}$. The acceptance of π by $\mathcal{A}$ is defined with respect to the type of $\mathcal{F}$, for which many have been proposed in the literature [14].

- For a *safety winning condition*, all infinite runs are accepting. In this case, the $\mathcal{F}$-symbol can also be omitted from the automaton definition.
- For a *Büchi acceptance condition* $\mathcal{F} \subseteq Q$, π is accepting if $\inf(\pi) \cap \mathcal{F} \neq \emptyset$.
- For a *co-Büchi acceptance condition* $\mathcal{F} \subseteq Q$, π is accepting if $\inf(\pi) \cap \mathcal{F} = \emptyset$.
- For a *generalized Büchi acceptance condition* $\mathcal{F} \subseteq 2^Q$, π is accepting if for all $F \in \mathcal{F}$, $\inf(\pi) \cap F \neq \emptyset$.
- For a *Rabin acceptance condition* $\mathcal{F} \subseteq 2^Q \times 2^Q$, π is accepting if for $\mathcal{F} = \{(F_1, G_1), \ldots, (F_n, G_n)\}$, there exists some $1 \leq i \leq n$ such that $\inf(\pi) \subseteq F_i$ and $\inf(\pi) \cap G_i \neq \emptyset$.

- For a *parity acceptance condition*, $\mathcal{F} : Q \rightarrow \mathbb{N}$ and π is accepting in the case that $\max\{\mathcal{F}(v) \mid v \in \inf(\pi)\}$ is even.
- For a *Streett acceptance condition* $\mathcal{F} \subseteq 2^Q \times 2^Q$, π is accepting if for $\mathcal{F} = \{(F_1, G_1), \ldots, (F_n, G_n)\}$ and for all $1 \leq i \leq n$, we have $\inf(\pi) \nsubseteq F_i$ or $\inf(\pi) \cap G_i = \emptyset$.
- For a *Muller acceptance condition* $\mathcal{F} \subseteq 2^Q$, π is accepting if $\inf(\pi) \in \mathcal{F}$.

The language of $\mathcal{A}$ is defined as the set of words for which there exists a run that is accepting with respect to the type of the acceptance condition. We also call automata with a t-type acceptance condition t-automata (for $t \in \{$safety, Büchi, co-Büchi, generalized Büchi, parity, Rabin, Streett, Muller$\}$). For all acceptance condition types stated above, $|\mathcal{F}|$ is defined as the cardinality of $\mathcal{F}$ (for safety automata we set $|\mathcal{F}| = 0$).[1] We define the size of $\mathcal{A}$, written as $|\mathcal{A}|$ to be $|Q| + |\Sigma| + |\delta| + |\mathcal{F}|$ for $|\delta| = |\{(q, q', e) \in Q \times Q \times \Sigma \mid q' \in \delta(q, x)\}|$.

We say that an algorithm approximates the minimal lasso/witness within some function $f(n)$ if for every problem instance with a shortest accepting lasso/witness having some size $n \in \mathbb{N}$, it always finds a solution of size not more than $f(n)$. An algorithm is said to approximate within a constant factor/within a polynomial if there exists some $c \in \mathbb{N}$/some polynomial function $p(n)$ such that it approximates within $f(n) = c \cdot n / f(n) = p(n)$, respectively. For the hardness and non-approximability results, we assume that P$\neq$NP (otherwise all problems discussed here are solvable in polynomial time).

An automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ can also be thought of as a graph $\langle V, E \rangle$ with vertices $V = Q$ and edges $E \subseteq V \times V$ such that for all $v_1, v_2 \in V$, $(v_1, v_2) \in E$ if there exists some $a \in \Sigma$ such that $v_2 \in \delta(v_1, a)$. A path in $\langle V, E \rangle$ going from v to v' is a sequence $\pi = v_1 \ldots v_n$ with $v_1 = v$ and $v_n = v'$ such that for all $i \in \{1, \ldots, n\}$, $v_i \in V$ and for all $i \in \{1, \ldots, n-1\}$, $(v_i, v_{i+1}) \in E$. A *strongly connected subset* of $\mathcal{A}$ is a set of states $Q' \subseteq Q$ such that there exist paths in $\langle Q', E|_{Q'} \rangle$ between all pairs of states in Q'.

For all acceptance condition types given above, the emptiness of the language of an automaton (i.e., whether there exists no accepted word) can be decided in time polynomial in the size of the automaton. For Rabin and Muller automata, this follows from standard automata constructions (see, e.g., the folk theorems in [15]). For Streett automata, this follows from the existence of efficient emptiness checking constructions [16]. Likewise, checking if a word uv^ω is accepted by some automaton $\mathcal{A}$ can also be performed in time polynomial in $|uv|$ and $|\mathcal{A}|$ for all of these acceptance condition types.

3 Finding Shortest Accepting Lassos

In this section, we deal with finding shortest accepting lassos in ω-regular automata. Given an ω-automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$, we formally define lassos as pairs (l, l') such that:

[1] As in this paper, we are only interested in the borderline between NP-complete problems and those that are in P (assuming P$\neq$NP), we can safely ignore the fact that an explicit encoding of $\mathcal{F}$ might actually be slightly bigger.

- $l = l_0 \ldots l_n \in Q^n$ for some $n \in \mathbb{N}_0$
- $l' = l'_0 \ldots l'_{n'} \in Q^{n'}$ for some $n' \in \mathbb{N}_{>0}$
- $l_0 = q_0$, $l_n = l'_0 = l'_{n'}$
- For all $i \in \{0, \ldots, n-1\}$, $\exists x \in \Sigma$ such that $\delta(l_i, x) = l_{i+1}$
- For all $i \in \{0, \ldots, n'-1\}$, $\exists x \in \Sigma$ such that $\delta(l'_i, x) = l'_{i+1}$

The length of such a lasso is defined to be $n + n'$. Given a lasso (l, l'), we call l' the *lasso cycle* of (l, l').

3.1 The Rabin Acceptance Condition and its Special Cases

First of all, we consider safety, Büchi, co-Büchi, parity and Rabin acceptance conditions and show that finding shortest accepting lassos for all of these acceptance condition types is doable in time (and thus, space) polynomial in the input size. Note that conversions from safety, Büchi, co-Büchi or parity acceptance components to equivalent Rabin acceptance components can easily be done with only polynomial blow-up (see, e.g., [17]).

For Büchi automata (and thus also safety automata as a special case), efficient algorithms for finding shortest accepting lassos are known, requiring roughly $O(|Q||\delta|)$ time (see [8, 9, 6] for entry points to the literature).

For the remaining cases, we show that finding shortest accepting lassos is solvable in polynomial time for Rabin automata, leading to the same result also for co-Büchi and parity automata. Without loss of generality, we can assume that a Rabin automaton has only one acceptance pair, i.e., $\mathcal{F} = \{(F, G)\}$ for some $F, G \subseteq Q$ as a word is accepted by a Rabin automaton $(Q, \Sigma, q_0, \delta, \mathcal{F})$ if and only if there exists an acceptance pair $(F, G) \in \mathcal{F}$ such that $\mathcal{A}' = (Q, \Sigma, q_0, \delta, \{(F, G)\})$ accepts the word. Therefore, by iterating over all elements in $\mathcal{F}$ and taking the shortest lasso found, we can extend a polynomial algorithm for a single acceptance pair to a polynomial algorithm for general Rabin automata.

Note that for a lasso (l, l') with $l' = l'_0 \ldots l'_{n'}$ to be accepting for $(Q, \Sigma, q_0, \delta, \{(F, G)\})$, we must have $\{q \in Q \mid \exists i : l'_i = q\} \setminus F = \emptyset$. So, states in $Q \setminus F$ may not occur on the cycle-part of the lasso. For each state $q \in F$, we can apply one of the basic shortest-lasso algorithms for Büchi automata on $(F, \Sigma, q, \delta|_F, G)$ and compute a shortest accepting lasso in it. Let the lasso length for each starting state $q \in F$ be called $c(q)$.

For actually obtaining a shortest accepting lasso over $(Q, \Sigma, q_0, \delta, \{(F, G)\})$, we can apply a standard shortest-path algorithm by interpreting $\mathcal{A}$ as a graph, adding a goal vertex to it, adding edges from each state $q \in Q$ where $c(q)$ is defined to this goal vertex with cost $c(q)$, and taking q_0 as the starting vertex. The remaining transitions have cost 1. By taking the shortest path up to the point where an added edge is taken and then replacing it by the corresponding lasso computed in the previous step, we easily obtain a shortest accepting lasso for $(Q, \Sigma, q_0, \delta, \{(F, G)\})$.

The overall complexity of this procedure is clearly polynomial in $|\mathcal{A}|$.

3.2 Generalized Büchi and Streett Automata

Rabin automata and their special cases have a certain property: On every shortest accepting lasso, no state can occur twice. This property does not hold for generalized Büchi and Streett acceptance conditions. Intuitively, this can make finding short accepting lassos significantly harder as the corresponding search space is larger. Indeed, the length of a shortest accepting lasso cannot be approximated within any constant in polynomial time if P≠NP. We prove this fact by reducing the Ek-Vertex-Cover problem [18] onto finding short accepting lassos.

Problem 1. A k-uniform hypergraph is a 2-tuple $G = \langle V, E \rangle$ such that V is a finite set and $E \subseteq 2^Q$ such that all elements in E are of cardinality k. Given a k-uniform hypergraph $H = \langle V, E \rangle$, the Ek-Vertex-Cover problem is to find a subset $V' \subseteq V$ of minimal cardinality such that for all $e \in E$: $e \cap V' \neq \emptyset$. It has been proven that approximating the minimal size of such a subset within a factor of $(k - 1 - \epsilon)$ for some $\epsilon > 0$ is NP-hard [19].

Consider a k-uniform hypergraph $G = \langle V, E \rangle$ for some arbitrary $k \in \mathbb{N}$. We can easily reduce the problem of finding a small Ek-Vertex-Cover to finding short accepting lassos in a generalized Büchi automaton over a one-element alphabet $\Sigma = \{\cdot\}$. We define $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ with $Q = V$, $\delta(q, \cdot) = Q$ for all $q \in Q$ (so we have a complete graph) and $\mathcal{F} = E$. Furthermore q_0 is set to some arbitrary element of Q. Given some vertex cover $V' \subseteq V$, it is clear from the definition of $\mathcal{A}$ that for $V' = \{v_1, \ldots, v_m\}$, the lasso (l, l') with $l = q_0 v_1$ and $l' = v_1 v_2 \ldots v_m v_1$ is accepting. On the other hand, an accepting lasso (l, l') with $l = q_0 v_1$ and $l' = v_1 v_2 \ldots v_m v_1$ induces a vertex cover $V' \subseteq V$ with $V' = \{v_1, \ldots, v_m\}$. Therefore, this reduction preserves the quality of the solutions up to a possible deviation of 1 (for the initial state of the lasso).

As the Ek-Vertex-Cover problem is reducible to finding short lassos (up to a deviation of 1) and is NP-hard to approximate within a factor of $(k - 1 - \epsilon)$ for all $k \in \mathbb{N}$ and $\epsilon > 0$, we obtain the following result:

Theorem 2. *Approximating the length of a shortest accepting lasso in generalized Büchi automata is NP-hard within any constant factor.*

As generalized Büchi automata have a simple translation to Streett automata, the same result holds for Streett automata as well. Note that these problems are also in NP as verifying the validity of an accepting lasso is simple and the length of a shortest accepting lasso in $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ is bounded by $|Q|^2$.

Thus, NP-completeness of these problems follows. Note that this line of reasoning also holds for the Muller acceptance condition to be discussed next.

3.3 Muller Automata

For finding short accepting lassos in Muller automata, we can use the same scheme as for Rabin automata: Given a Muller automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$

with $\mathcal{F} = \{F_1, \ldots, F_m\}$, we can search for short accepting lassos in each of the automata $(Q, \Sigma, q_0, \delta, F_1)$, …, $(Q, \Sigma, q_0, \delta, F_m)$ and take the shortest accepting lasso we find in these automata as a shortest lasso for $\mathcal{A}$. Thus, assuming that we have a $f(n)$-approximation algorithm for a Muller automaton with a single acceptance set running in polynomial time, this immediately gives rise to a polynomial $f(n)$-approximation algorithm for general Muller automata.

For a lasso (l, l') to be accepting for some Muller acceptance set F, all states in F must occur in l'. As we can furthermore assume that the states in $F \subseteq Q$ form a strongly connected subset in Q (as otherwise F cannot be precisely the set of states occurring infinitely often on a run), the problem of finding a short accepting lasso is related to the *asymmetric metric travelling salesman problem* (AMTSP), as we explain in the remainder of this section.

Problem 3. Given a set of cities C with $|C| = n$ and a distance function $d : C \times C \rightarrow \mathbb{N}_0$ such that $d(c, c) = 0$ for all $c \in C$ and for every $c_1, c_2, c_3 \in C$, $d(c_1, c_2) + d(c_2, c_3) \leq d(c_1, c_3)$, the AMTSP-problem is to find a cycle $c_0, \ldots, c_{n-1}$ such that the cost of the cycle (i.e., $\sum_{i=0}^{n-1} d(c_i, c_{(i+1) \bmod n})$) is as small as possible.

It has been proven that in a special case of the AMTSP problem in which the distance between two cities is either 1 or 2, the cost of the cheapest cycle cannot be approximated within a factor of $\frac{321}{320} - \epsilon$ for some $\epsilon > 0$ in polynomial time, unless P=NP [20]. A simple reduction shows that this is also the case for finding shortest accepting lassos in Muller automata:

Theorem 4. *Approximating the length of a shortest accepting lasso within a factor of $\frac{321}{320} - \epsilon$ for some $\epsilon > 0$ in a Muller automaton is NP-hard.*

Proof. Given an AMTSP-Problem $\langle C, d \rangle$ in which the distance between two different cities is always 1 or 2, we reduce finding the length of a shortest cycle to finding the shortest accepting lasso in a Muller automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ over $\Sigma = \{\cdot\}$ with $\mathcal{F} = \{F_1, F_2\}$ by defining $Q = C \uplus \{\Gamma\}$, $\delta(c, \cdot) = \{c' \in C \mid d(c, c') = 1\} \cup \{\Gamma\}$ (for all $c \in C$), $\delta(\Gamma, \cdot) = C$, $F_1 = C$, $F_2 = Q$ and set $q_0 = c$ for some arbitrary $c \in C$.

For every cycle of length j for some $j \in \mathbb{N}$, there exists an accepting lasso of the same length starting with q_0. Whenever an edge with cost 2 is taken, the lasso is routed through Γ, the other edges can be taken directly.

On the other hand, each lasso cycle in $\mathcal{A}$ induces a cycle in $\langle C, d \rangle$ with a cost equal to the length of the lasso by skipping over all visits to Γ. Without loss of generality, we can assume that such an accepting lasso (l, l') has $l = q_0$.

As this way, the cost of the cycle and the lasso length coincide and approximating the cost of a shortest cycle in $\langle C, d \rangle$ within $\frac{321}{320} - \epsilon$ is NP-complete for all $\epsilon > 0$, the claim follows.

Thus, also in the Muller automaton case, we cannot approximate the size of a shortest accepting lasso arbitrarily well. However, the close connection between the AMTSP problem and Muller automaton emptiness allows us to make use of a positive approximation result for the AMTSP problem:

Theorem 5. *Given a Muller automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ with $\mathcal{F} = \{F_1\}$, we can compute a lasso of length not more than $\ulcorner \log_2 |F_1| \urcorner$ times the length of a shortest one in polynomial time.*

Proof. The problem can be solved using a $\ulcorner \log_2 |n| \urcorner$-approximation algorithm for the AMTSP problem [21]. We construct an AMTSP instance $\langle C, d \rangle$ by taking $C = F_1$ and for each pair of cities $c_1, c_2 \in C$ with $c_1 \neq c_2$, we use a standard shortest-path finding algorithm for computing $d(c_1, c_2)$, i.e., the length of the shortest path through the graph of $\mathcal{A}$ restricted to F_1 from state c_1 to c_2. For every computed value, we store the corresponding path for later retrieval. Then, we apply the approximation algorithm on $\langle C, d \rangle$ and obtain a tour of length at most $\ulcorner \log_2 |F_1| \urcorner \cdot m$, where m is the length of the optimal tour. As we can assume that F_1 is a strongly connected subset in $\mathcal{A}$, taking the tour and stitching together the individual respective parts we stored in the previous step results in a lasso cycle with a length equal to the cost of the tour. By finding a shortest path in $\mathcal{A}$ from q_0 to one of the states in F_1 and adding this path as first part of the lasso, we obtain a complete accepting lasso. The approximation quality of the solution follows directly from the definition of $\langle C, d \rangle$ and the fact that the first part of the lasso is indeed as short as possible as all elements in F_1 have to occur on the cycle.

4 Finding Shortest Witnesses

In this section, we consider finding shortest witnesses, i.e., given some ω-automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$, the task is to find a word uv^ω for $u, v \in \Sigma^*$ that is accepted by $\mathcal{A}$ with $|u| + |v|$ being as small as possible. We show that approximating the length of a shortest such word within any polynomial is NP-complete for all acceptance condition types considered in this paper, but we can approximate this length within any exponential function in polynomial time (for every fixed alphabet Σ). We start with the hardness result.

Theorem 6. *Given some polynomial function p, approximating the length of a minimal witness in some safety-type ω-automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta)$ within p over a ternary alphabet $\Sigma = \{0, 1, \#\}$ is NP-hard.*

Proof. The proof is based on a reduction from the satisfiability (SAT) problem, which is known to be NP-hard (see, e.g., [22] for details).

We define a conjunctive normal form SAT-instance to consist of a set of variables $V = \{v_1, \ldots, v_m\}$ and a set of clauses $C = \{c_1, \ldots, c_n\}$ (with $c_i : V \times \{0, 1\} \rightarrow \mathbb{B}$ for all $1 \leq i \leq n$) which are formally functions such that $c_i(v_k, 1) =$ **true** if and only if v_k is a literal in clause i and $c_i(v_k, 0) =$ **true** if $\neg v_k$ is a literal in clause i (for all $1 \leq k \leq m$, $1 \leq i \leq n$).

We reduce the problem of determining whether there exists some valuation of the variables that satisfies all clauses in C to finding some short witness in some safety automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta)$ over $\Sigma = \{0, 1, \#\}$ as follows:

- $Q = \{(i, j, k, b) \in \mathbb{N}^3 \times \mathbb{B} \mid 1 \leq i \leq n, 1 \leq j \leq p(m), 1 \leq k \leq m+1, b \Rightarrow k > 1\} \cup \{\bot\}$
- $q_0 = (1, 1, 1, \textbf{false})$
- For all $(i, j, k, b) \in Q$, $a \in \{0, 1, \#\}$, $\delta((i, j, k, b), a)$ is the union of:
  - $\{(i, j, k+1, b') \mid k \leq m, b' = (b \vee c_i(v_k, a))\}$
  - $\{(i, j+1, 1, \textbf{false}) \mid b = \textbf{true}, a = \#, j \leq p(m), k = m+1\}$
  - $\{(i+1, 1, 1, \textbf{false}) \mid b = \textbf{true}, j = p(m), k = m+1, a = \#, i \leq n\}$
  - $\{\bot\}$ if $b = \textbf{true}$, $j = p(m)$, $k = m+1$, $a = \#$, b and $i = n$
- $\delta(\bot, a) = \{\bot\}$ for all $a \in \{0, 1, \#\}$

Figure 1 gives an example of such an automaton for an example SAT-instance.

The key idea of this reduction is the following: The automaton built only accepts input words on which during the first $p(m)(m+1)n$ input letters, precisely every $(m+1)$th letter is a #. Furthermore, the letters in between represent valuations to the variables in the SAT instance. During the first $p(m)(m+1)$ input letters, it is checked that the solution given satisfies the first clause. Subsequent parts of the input words are then checked against the next clause (and so on). Now assume that a word uv^ω for which $|u|+|v| \leq p(m)(m+1)$ holds is accepted by the automaton. All parts in between two occurrences of # in the word represent variable valuations satisfying all clauses. On the other hand, if there exists some valuation for the variables satisfying all clauses, then there exists a simple word with $|u| = 0$ and $|v| = m+1$ such that uv^ω is accepted. Therefore, by using a p-approximation algorithm for finding the length of a shortest accepting witness, we can check if there exists a valuation of V satisfying C.

This non-approximability result for finding (or even determining the minimal size of) short witnesses is surprising. While finding short accepting lassos is doable in polynomial time even for the more complex Rabin condition, approximating the size of a shortest witness is NP-hard even for safety automata and thus considerably harder. For the other acceptance condition types, the same result holds as only the state $\bot$ can be visited infinitely often on any accepting run. It is trivial to build corresponding acceptance components for any of the other acceptance condition types defined in this paper. The hardness proof given above also holds for a binary alphabet with only a slight modification.

As in the case of finding short accepting lassos, the fact that the problem of finding a shortest witness is actually contained in NP is easy to show: for all automaton types considered, the problem of checking whether a word uv^ω is in the language of the automaton is solvable in polynomial time. Furthermore, if the language of the automaton is non-empty, then there exists some witness of length not more than the square of the automaton's number of states. By taking together these facts, membership in NP trivially follows.

A natural question to ask at this point is which positive statements about the approximability of this problem can be given. In this paper, we show the following:

Theorem 7. *Let $c > 1$ and Σ be some fixed finite alphabet. Given some ω-automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ with any of the acceptance types considered in*

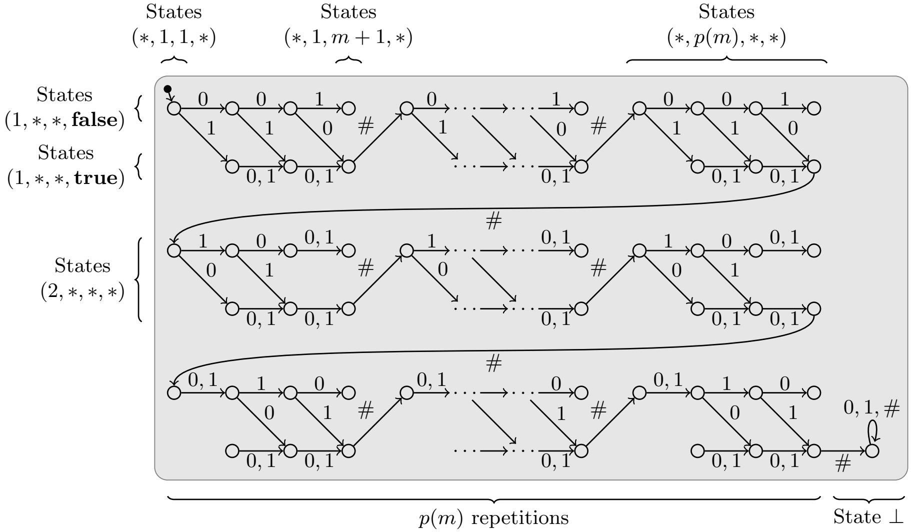

Fig. 1. Example automaton constructed from the SAT-instance $(v_1 \vee v_2 \vee \neg v_3) \wedge (\neg v_1 \vee v_2) \wedge (\neg v_2 \vee v_3)$ as described in the proof of Theorem 6. In this example, we have $m = 3$ and $n = 3$ with $V = \{v_1, v_2, v_3\}$. The labels next to the braces explain the structure of the automaton generated (with $*$ denoting that the states corresponding to any suitable value at this point in the tuple are contained in the state set).

this paper, computing a word uv^ω such that $|u| + |v|$ is not longer than c^n for n being the minimal witness length can be done in time polynomial in $|\mathcal{A}|$.

Proof. Note that for all acceptance types considered in this paper, checking whether a word uv^ω is accepted by $\mathcal{A}$ is possible in time polynomial in $|u| + |v|$ and $|\mathcal{A}|$.

Furthermore, for all acceptance condition types, emptiness checking and the extraction of an accepting lasso of size no longer than $|Q|^2$ can be performed in polynomial time. Therefore, we can iterate over all words uv^ω such that $|uv| \leq \ulcorner \log_c |Q|^2 \urcorner$ (which are only polynomially many) and check for each of them whether they are in the language of the automaton. A c^n-approximation algorithm can thus return the shortest such witness, if found. In all other cases, the simple accepting lasso of size not more than $|Q|^2$ can be converted to an accepting word by copying the edge labels. The fact that exponentially shorter words would have been found by the first step suffices for proving the approximation quality of this algorithm.

Taking the results obtained in this section together, we obtain a quite precise characterisation of the approximation hardness of finding short witnesses in ω-automata: Approximating the size within any polynomial is NP-complete, but the problem is approximable within any exponential function in polynomial time for every fixed alphabet.

Table 1. Summary of the approximability results on finding short accepting lassos and witnesses for the acceptance condition types considered in this paper. In all cases, it is assumed that P≠NP and only algorithms running in time polynomial in the size of the input are considered.

Acceptance cond. type	**Shortest accepting lassos**	**Shortest witnesses**
Safety, Büchi, co-Büchi, parity, Rabin	solvable precisely in polynomial time	not approximable within any polynomial, approximable within every exponential function for a fixed alphabet
Generalized Büchi, Streett	not approximable within any constant, approximable within every exponential function for a fixed alphabet	
Muller	not approximable within $\frac{321}{320}-\epsilon$, approximable within $\ulcorner \log_2 \lvert Q \rvert \urcorner$	

As a final note, the exponential-quality approximation algorithm presented in this section is also useful for finding short accepting lassos. Therefore, we obtain the same upper bound on the approximation hardness of that problem.

5 Conclusion

In this paper, we have examined the problem of finding short accepting lassos and witnesses for ω-automata of various acceptance condition types. We bounded the borderline between NP-complete approximation problems and those in P from both above and below (assuming that P≠NP) by giving NP-hardness proofs for numerous variations of the problem along with polynomial approximation algorithms of lower approximation quality. Table 1 summarises the details of the findings.

Additionally, for the case of short accepting lassos for Muller automata, we have established its connection to the travelling salesman problem by identifying it as special case of the asymmetric metric TSP.

We considered the automata types currently employed in model checking applications as well as those that currently mainly serve as models in theoretical works in order to fill the automata-theoretic toolbox for use cases which have not been discovered yet.

At a first glance, the non-approximability results for Büchi and generalized Büchi automata are discouraging: Assuming that P≠NP, the implementation of methods for extracting approximate shortest witnesses (or approximate shortest lassos in the case of fair systems) for the non-satisfaction of a specification in future model checkers appears not to be a fruitful idea. However, it should be noted that the identification of these problems as being hard helps preparing the field for the development of suitable heuristics. Also, the hardness results obtained may serve as justification for developing counter-example quality metrics which also base on other objectives than only their size.

References

1. Vardi, M.Y.: An automata-theoretic approach to linear temporal logic. In: Proceedings of the VIII Banff Higher order workshop conference on Logics for concurrency: structure versus automata, Springer-Verlag New York, Inc. (1996) 238–266
2. Baier, C., Katoen, J.P.: Principles of Model Checking. MIT Press (2008)
3. Grumberg, O., Veith, H., eds.: 25 Years of Model Checking - History, Achievements, Perspectives. Volume 5000 of LNCS. Springer (2008)
4. Mitra, R.S.: Strategies for mainstream usage of formal verification. In Fix, L., ed.: DAC, ACM (2008) 800–805
5. Woodcock, J., Larsen, P.G., Bicarregui, J., Fitzgerald, J.S.: Formal methods: Practice and experience. ACM Comput. Surv. **41**(4) (2009)
6. Hansen, H., Geldenhuys, J.: Cheap and small counterexamples. In Cerone, A., Gruner, S., eds.: SEFM, IEEE Computer Society (2008) 53–62
7. Groce, A., Visser, W.: What went wrong: Explaining counterexamples. In Ball, T., Rajamani, S.K., eds.: SPIN. Volume 2648 of LNCS., Springer (2003) 121–135
8. Gastin, P., Moro, P.: Minimal counterexample generation for SPIN. In Bosnacki, D., Edelkamp, S., eds.: SPIN. Volume 4595 of LNCS., Springer (2007) 24–38
9. Schwoon, S., Esparza, J.: A note on on-the-fly verification algorithms. In Halbwachs, N., Zuck, L.D., eds.: TACAS. Volume 3440 of LNCS. (2005) 174–190
10. Clarke, E.M., Grumberg, O., McMillan, K.L., Zhao, X.: Efficient generation of counterexamples and witnesses in symbolic model checking. In: DAC. (1995) 427–432
11. Kupferman, O., Sheinvald-Faragy, S.: Finding shortest witnesses to the nonemptiness of automata on infinite words. In Baier, C., Hermanns, H., eds.: CONCUR. Volume 4137 of LNCS., Springer (2006) 492–508
12. Clarke, E.M., Emerson, E.A.: Design and synthesis of synchronization skeletons using branching-time temporal logic. In Kozen, D., ed.: Logic of Programs. Volume 131 of LNCS., Springer (1981) 52–71
13. Büchi, J.R.: On a decision method in restricted second-order arithmetic. In: Proc. 1960 Int. Congr. for Logic, Methodology, and Philosophy of Science. (1962) 1–11
14. Grädel, E., Thomas, W., Wilke, T., eds.: Automata, Logics, and Infinite Games: A Guide to Current Research. Volume 2500 of LNCS. Springer (2002)
15. Safra, S.: Complexity of Automata on Infinite Objects. PhD thesis, Weizmann Institute of Science, Rehovot, Israel (March 1989)
16. Henzinger, M.R., Telle, J.A.: Faster algorithms for the nonemptiness of Streett automata and for communication protocol pruning. In Karlsson, R.G., Lingas, A., eds.: SWAT, Springer (1996) 16–27
17. Farwer, B.: ω-automata. [14] 3–20
18. Khot, S., Regev, O.: Vertex cover might be hard to approximate to within 2-ϵ. J. Comput. Syst. Sci. **74**(3) (2008) 335–349
19. Dinur, I., Guruswami, V., Khot, S., Regev, O.: A new multilayered PCP and the hardness of hypergraph vertex cover. SIAM J. Comput. **34**(5) (2005) 1129–1146
20. Engebretsen, L., Karpinski, M.: Approximation hardness of TSP with bounded metrics. In Orejas, F., Spirakis, P.G., van Leeuwen, J., eds.: ICALP. Volume 2076 of LNCS., Springer (2001) 201–212
21. Frieze, A.M., Galbiati, G., Maffioli, F.: On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. Networks **12**(1) (1982) 23–39
22. Wegener, I.: Complexity Theory. Exploring the Limits of Efficient Algorithms. Springer Verlag (2004)