Model and Objective Separation with Conditional Lower Bounds: Disjunction is Harder than Conjunction *

Krishnendu Chatterjee
IST Austria
krishnendu.chatterjee@ist.ac.at

Wolfgang Dvořák Monika Henzinger
Veronika Loitzenbauer
University of Vienna, Faculty of Computer Science,
Vienna, Austria
{wolfgang.dvorak, monika.henzinger,
veronika.loitzenbauer}@univie.ac.at

Abstract

Given a model of a system and an objective, the model-checking question asks whether the model satisfies the objective. We study polynomial-time problems in two classical models, graphs and Markov Decision Processes (MDPs), with respect to several fundamental ω-regular objectives, e.g., Rabin and Streett objectives. For many of these problems the best-known upper bounds are quadratic or cubic, yet no super-linear lower bounds are known. In this work our contributions are two-fold: First, we present several improved algorithms, and second, we present the first conditional super-linear lower bounds based on widely believed assumptions about the complexity of CNF-SAT and combinatorial Boolean matrix multiplication. A separation result for two models with respect to an objective means a conditional lower bound for one model that is strictly higher than the existing upper bound for the other model, and similarly for two objectives with respect to a model. Our results establish the following separation results: (1) A separation of models (graphs and MDPs) for disjunctive queries of reachability and Büchi objectives. (2) Two kinds of separations of objectives, both for graphs and MDPs, namely, (2a) the separation of dual objectives such as Streett/Rabin objectives, and (2b) the separation of conjunction and disjunction of multiple objectives of the same type such as safety, Büchi, and coBüchi. In summary, our results establish the first model and objective separation results for graphs and MDPs for various classical ω-regular objectives. Quite strikingly, we establish conditional lower bounds for the disjunction of objectives that are strictly higher than the existing upper bounds for the conjunction of the same objectives.

Categories and Subject Descriptors F [2]: 2—Computations on discrete structures

General Terms Algorithms, Verification

Keywords Conditional lower bounds; Graph algorithms; Markov Decision processes; Model checking;

* Full version available at http://arxiv.org/abs/1602.02670.

LICS '16, July 05 - 08, 2016, New York, NY, USA

ACM 978-1-4503-4391-6/16/07…$15.00
DOI: http://dx.doi.org/10.1145/2933575.2935304

1. Introduction

The fundamental problem in formal verification is the *model-checking* question that given a model of a system and a property asks whether the model satisfies the property. The model can be, for example, a standard graph, or a probabilistic extension of graphs, and the property describes the desired behaviors of the model. For several basic model-checking questions, though polynomial-time algorithms are known, the best-known existing upper bounds are quadratic or cubic, yet no super-linear lower bounds are known. In graph algorithmic problems unconditional super-linear lower bounds are very rare when polynomial-time solutions exist. However, recently there have been many interesting results that establish *conditional lower bounds* [1, 3, 5]. These are lower bounds based on the assumption that for some well-studied problem such as 3-SUM [23] or All-Pairs Shortest Paths [33, 36] no (polynomially[1]) faster algorithm exists (compared to the best known algorithm). The lower bounds in this work assume (A1) there is no combinatorial[2] algorithm with running time of $O(n^{3-\varepsilon})$ for any $\varepsilon > 0$ to multiply two $n \times n$ Boolean matrices; or (A2) for all $\varepsilon > 0$ there exists a k such that there is no algorithm for the k-CNF-SAT problem that runs in $2^{(1-\varepsilon)\cdot n} \cdot \mathrm{poly}(m)$ time, where n is the number of variables and m the number of clauses. These two assumptions have been used to establish lower bounds for several well-studied problems, such as dynamic graph algorithms [1, 5], measuring the similarity of strings [2, 4, 7, 9, 10], context-free grammar parsing [3, 31], and verifying first-order graph properties [32, 38]. No relation between conjectures (A1) and (A2) is known. In this work we present conditional lower bounds that are super-linear for fundamental model-checking problems.

Models. The two most classical models in formal verification are *standard graphs* and *Markov decision processes* (*MDPs*). MDPs are probabilistic extensions of graphs, and an MDP consists of a finite directed graph (V, E) with a partition of the vertex set V into player 1 vertices V_1 and random vertices V_R and a probabilistic transition function that specifies for vertices in V_R a probability distribution over their successor vertices. Let $n = |V|$ and $m = |E|$. An infinite path in an MDP is obtained by the following process. A token is placed on an initial vertex and the token is moved indefinitely as follows: At a vertex $v \in V_1$ a choice is made to move the token along one of the outedges of v, and at a vertex $v \in V_R$ the token is moved according to the probabilistic transition function. Note that if $V_R = \emptyset$, then we have a standard graph, and

[1] In particular improvements by polylogarithmic factors are not excluded.

[2] Combinatorial here means avoiding fast matrix multiplication [30], see also the discussion in [26].

if $V_1 = \emptyset$, then we have a Markov chain. Thus MDPs generalize standard graphs and Markov chains.

Objectives. Objectives (or properties) are subsets of infinite paths that specify the desired set of paths. Let $T \subseteq V$ be a set of target vertices. The most basic objective is *reachability* where an infinite path satisfies the objective if the path visits a vertex of T *at least once*. The dual objective to reachability is *safety* where an infinite path satisfies the objective if the path does *not* visit any vertex of T. The next extension of a reachability objective is the *Büchi* objective that requires the set T to be reached *infinitely often*. Its dual, the *coBüchi* objective, requires the set T to be reached only *finitely often*. A natural extension of single objectives are *conjunctive* and *disjunctive objectives* [16, 22, 40]. For two objectives ψ_1 and ψ_2 their conjunctive objective is equal to $\psi_1 \cap \psi_2$ and their disjunctive objective is equal to $\psi_1 \cup \psi_2$. The conjunction of reachability (resp. Büchi) objectives is known as *generalized reachability* (*resp. Büchi*) [22, 40]. A very central and canonical class of objectives in formal verification are *Streett* (strong fairness) objectives and their dual *Rabin* objectives [35]. A *one-pair Streett* objective for two sets of vertices L and U specifies that if the Büchi objective for target set L is satisfied, then also the Büchi objective for target set U has to be satisfied; in other words, a one-pair Streett objective is the disjunction of a coBüchi objective (with target set L) and a Büchi objective (with target set U). The dual *one-pair Rabin* objective for two vertex sets L and U is the conjunction of a Büchi objective with target set L and a coBüchi objective with target set U. A Streett objective is the conjunction of k one-pair Streett objectives and its dual Rabin objective is the disjunction of k one-pair Rabin objectives.

Algorithmic questions. Given a model and an objective, the algorithmic question (a) for standard graphs is whether there is a path that satisfies the objective and (b) for MDPs is whether there is a *strategy* that resolves the non-deterministic choices of outgoing edges for player 1 to ensure that the objective is satisfied with probability 1. Observe that if we consider the model-checking question for MDPs with $V_R = \emptyset$, then it exactly corresponds to the model-checking question for standard graphs. Given k objectives, the *conjunctive query* question asks whether there is a strategy for player 1 to ensure that *all* the objectives are satisfied with probability 1, and the *disjunctive query* question asks whether there is a strategy for player 1 to ensure that *one* of the objectives is satisfied with probability 1. Conjunctive queries coincide with conjunctive objectives on graphs and MDPs, while disjunctive queries coincide with disjunctive objectives on graphs but not on MDPs.

Significance of model and objectives. Standard graphs are the model for non-deterministic systems, and provide the framework to model hardware and software systems [20, 27], as well as many basic logic-related questions such as automata emptiness. MDPs model systems with both non-deterministic and probabilistic behavior; and provide the framework for a wide range of applications from randomized communication and security protocols, to stochastic distributed systems, to biological systems [8, 29]. In verification, reachability objectives are the most basic objectives for safety-critical systems. In general all properties that arise in verification (such as liveness, fairness) are ω-regular languages (ω-regular languages extend regular languages to infinite words), and every ω-regular language can be expressed as a Streett objective (or a Rabin objective). Important special cases of Streett (resp. Rabin) objectives are Büchi and coBüchi objectives [14]. Thus the algorithmic questions we consider are the most basic questions in formal verification.

Model separation and objective separation questions. In this work our results (upper and conditional lower bounds) aim to establish the following two fundamental separations:

- *Model separation.* Consider an objective where the algorithmic question for both graphs and MDPs can be solved in polynomial time, and establish a conditional lower bound for MDPs that is strictly higher than the best-known upper bound for graphs.
- *Objective separation.* Consider a model (either graphs or MDPs) with two different objectives and show that, though the algorithmic question for both objectives can be solved in polynomial time, there is a conditional lower bound for one objective that is strictly higher than the best-known upper bound for the other objective.

To the best of our knowledge, there is no previous work that establish any model separation or objective separation result in the literature.

Our results. In this work we present improved algorithms as well as the first conditional lower bounds that are super-linear for algorithmic problems in model checking that can be solved in polynomial time, and together they establish both model separation and objective separation results. An overview of the results for the different objectives is given in Table 1, where our results are highlighted in boldface. We use MEC to refer to the time to compute the maximal end-component decomposition of an MDP. An end-component is a strongly connected sub-MDP for which random vertices have no edges out of the component. We have $\text{MEC} = O(\min(n^2, m^{1.5}))$ [14]. Moreover, we use k to denote the number of combined objectives in the case of conjunction or disjunction of multiple objectives and b to denote the total number of elements in all the target sets that specify the objectives. We first describe Table 1 and our main results and then discuss the significance of our results for model and objective separation.

1. *Conjunctive and Disjunctive Reachability* (*and Büchi*) *Problems.* First, we consider conjunctive and disjunctive reachability objectives and queries. Recall that conjunctive objectives and queries in general and disjunctive objectives and queries on graphs coincide. For reachability further the disjunctive objective can be reduced to a single objective. The following results are known: the algorithmic question for conjunctive reachability objectives is NP-complete for graphs [18], and PSPACE-complete for MDPs [22]; and the disjunctive objective can be solved in linear time for graphs and in $O(\min(n^2, m^{1.5}) + b)$ time in MDPs [14, 15]. We present three results for disjunctive reachability queries in MDPs: (i) We present an $O(km+\text{MEC})$-time algorithm. (ii) We show that under assumption (A1) there does not exist a combinatorial $O(k \cdot n^{2-\varepsilon})$ algorithm for any $\varepsilon > 0$. (iii) We show that for $k = \Omega(m)$ there does not exist an $O(m^{2-\varepsilon})$ time algorithm for any $\varepsilon > 0$ under assumption (A2). Hence we establish an upper bound and matching conditional lower bounds based on (A1) and (A2).

   Disjunctive Büchi objectives (on graphs and MDPs) can be reduced in linear time to disjunctive reachability objectives and vice versa, therefore the same results apply to disjunctive Büchi problems. The basic algorithm for conjunctive Büchi objectives runs in time $O(m + b)$ on graphs and in time $O(\text{MEC} + b)$ on MDPs.

2. *Conjunctive and Disjunctive Safety Problems.* Second, we consider conjunctive and disjunctive safety objectives and queries. The following results are known: the conjunctive problem can be reduced to a single objective and can be solved in linear time, both in graphs and MDPs (see e.g. [17]); disjunctive queries for MDPs can be solved in $O(k \cdot m)$ time; and disjunctive objectives for MDPs are PSPACE-complete [22]. We present two results: (i) We show that for the disjunctive problem in graphs under assumption (A1) there does not exist a combinatorial $O(k \cdot n^{2-\varepsilon})$ algorithm for any $\varepsilon > 0$. This implies the same conditional lower bound for disjunctive queries and objectives in MDPs and

Table 1. Upper and lower bounds. Our results are boldface and respective results are referred.

		Graphs		MDPs	
		upper bound	lower bound*	upper bound	lower bound*
Reach	Conj.	NP-c [18]		PSPACE-c [22]	
	Disj. Obj.	$\Theta(m+b)$		$O(\text{MEC}+b)$ [14, 15]	
	Disj. Qu.			$\mathbf{O(k \cdot m + MEC)}$ [Th. 4.1]	$\mathbf{k \cdot n^{2-o(1)}}$ [Th. 3.1], $\mathbf{m^{2-o(1)}}$ [Th. 3.4]
Safety	Conj.	$\Theta(m+b)$		$\Theta(m+b)$	
	Disj. Obj.	$O(k \cdot m)$	$\mathbf{k \cdot n^{2-o(1)}}$ [Th. 3.7]	PSPACE-c [22]	
	Disj. Qu.			$O(k \cdot m)$	$\mathbf{k \cdot n^{2-o(1)}}$ [Th. 3.7], $\mathbf{m^{2-o(1)}}$ [Th. 3.10]
Büchi	Conj.	$\Theta(m+b)$		$O(\text{MEC}+b)$	
	Disj. Obj.	$\Theta(m+b)$		$O(\text{MEC}+b)$ [14, 15]	
	Disj. Qu.			$\mathbf{O(k \cdot m + MEC)}$ [Th. 4.1]	$\mathbf{k \cdot n^{2-o(1)}}$ [Cor. 3.13], $\mathbf{m^{2-o(1)}}$ [Cor. 3.14]
coBüchi	Conj.	$\Theta(m+b)$		$O(\text{MEC}+b)$	
	Disj. Obj.	$O(k \cdot m)$	$\mathbf{k \cdot n^{2-o(1)}}$ [Cor. 3.13]	$\mathbf{O(k \cdot m + MEC)}$ [Th. 4.3]	$\mathbf{k \cdot n^{2-o(1)}}$ [Cor. 3.13], $\mathbf{m^{2-o(1)}}$ [Cor. 3.14]
	Disj. Qu.			$\mathbf{O(k \cdot m + MEC)}$ [Th. 4.3]	$\mathbf{k \cdot n^{2-o(1)}}$ [Cor. 3.13], $\mathbf{m^{2-o(1)}}$ [Cor. 3.14]
Singleton	Disj. Obj.	$\mathbf{\Theta(m)}$ [Th. 4.4]			$\mathbf{m^{2-o(1)}}$ [Cor. 3.14]
	Disj. Qu.				$\mathbf{m^{2-o(1)}}$ [Cor. 3.14]
Streett		$O(\min(n^2, m\sqrt{m \log n}, km) + b \log n)$ [19, 25]		$\mathbf{O(min(n^2, m\sqrt{m \log n}) + b \log n)}$ [Th. 4.2]	
Rabin		$O(k \cdot m)$	$\mathbf{k \cdot n^{2-o(1)}}$ [Cor. 3.13]	$O(k \cdot \text{MEC})$	$\mathbf{k \cdot n^{2-o(1)}}$ [Cor. 3.13], $\mathbf{m^{2-o(1)}}$ [Cor. 3.14]

* $\mathbf{k \cdot n^{2-o(1)}}$ lower bounds are based on the combinatorial Boolean Matrix Multiplication Conjecture / Strong Triangle Conjecture (A1)
$\mathbf{m^{2-o(1)}}$ lower bounds are based on the Orthogonal Vectors Conjecture / Strong ETH (A2)

matches the upper bound for graphs and disjunctive queries in MDPs. (ii) We present, for $k = \Omega(m)$, an $\Omega(m^{2-o(1)})$ lower bound for disjunctive objectives and queries in MDPs under assumption (A2). Again this lower bound matches the upper bound of $O(k \cdot m)$ for disjunctive queries.

3. *Conjunctive and Disjunctive coBüchi Problems.* For coBüchi, a conjunctive objective can be reduced to a single objective. For single objectives the basic algorithm runs in time $O(\text{MEC} + b)$ on MDPs and in time $O(m+b)$ on graphs. Our conditional lower bounds for disjunctive safety objectives and queries also hold for coBüchi objectives. Here the running times and the conditional lower bounds are matching for both disjunctive queries and disjunctive objectives. For the conditional lower bound based on assumption (A2) only *singleton coBüchi* objectives, i.e., coBüchi objectives with target sets of cardinality one, are needed, therefore the bound already holds for this case. We additionally present two results: (i) We present $O(km + \text{MEC})$-time algorithms for disjunctive queries and objectives in MDPs. (ii) We present a linear time algorithm for disjunctive singleton coBüchi objectives in graphs.
4. *Rabin and Streett objectives.* Finally, we consider Rabin and Streett objectives. The basic algorithm for Rabin objectives runs in time $O(k \cdot m)$ on graphs and in time $O(k \cdot \text{MEC})$ on MDPs. As disjunctive coBüchi objectives are a special case of Rabin objectives, the conditional lower bounds for coBüchi objectives of $\Omega(k \cdot n^{2-o(1)})$ on graphs and additionally $\Omega(m^{2-o(1)})$ on MDPs extend to Rabin objectives. The conditional lower bound for graphs is matching (for combinatorial algorithms). Furthermore, we extend the results of [19, 25] from graphs to MDPs to show that MDPs with Streett objectives can be solved in $O(\min(m\sqrt{m \log n}, n^2) + b \log n)$ time.

Significance of our results. We now describe the model and objective separation results that are obtained from the results we established.

Table 2. Model Separation.

	upper bound Graphs	lower bounds MDPs
Reach/Büchi Disj. Qu.	$m + nk$	$\mathbf{k \cdot n^{2-o(1)}}, \mathbf{m^{2-o(1)}}$
coBüchi Singl. Disj.	$\mathbf{m}$	$\mathbf{m^{2-o(1)}}$

Table 3. Dual Objective Separation for Graphs.

	upper bound	lower bound	
Reach Disj.	$m + nk$	$\mathbf{k \cdot n^{2-o(1)}}$	Safety Disj.
Büchi Disj.	$m + nk$	$\mathbf{k \cdot n^{2-o(1)}}$	coBüchi Disj.
Büchi Conj.	$m + nk$	$\mathbf{k \cdot n^{2-o(1)}}$	coBüchi Disj.
Streett	$n^2 + nk \log n$	$\mathbf{k \cdot n^{2-o(1)}}$	Rabin

Table 4. Dual Objective Separation for MDPs.

	upper bound	lower bound	
Büchi Disj. O.	$\min(n^2, m^{1.5}) + nk$	$\mathbf{k \cdot n^{2-o(1)}}, \mathbf{m^{2-o(1)}}$	coB. Disj. O.
Büchi Conj.	$\min(n^2, m^{1.5}) + nk$	$\mathbf{k \cdot n^{2-o(1)}}, \mathbf{m^{2-o(1)}}$	coB. Disj. O.
Streett	$\mathbf{\widetilde{O}(min(n^2, m^{1.5}))}$	$\mathbf{k \cdot n^{2-o(1)}}, \mathbf{m^{2-o(1)}}$	Rabin

1. *Model Separation.* Table 2 shows our results that separate graphs and MDPs regarding their complexity for certain objectives and queries under assumptions (A1) and (A2). First, for reachability and Büchi objectives disjunction in graphs is in linear time while in MDPs we have $\Omega(kn^{2-o(1)})$ and $\Omega(m^{2-o(1)})$ conditional lower bounds for disjunctive queries. Second, in graphs the disjunction of coBüchi objectives where each target set is a singleton is in linear time while we establish an $\Omega(m^{2-o(1)})$ conditional lower bound for MDPs for both disjunctive objectives and queries.

2. *Objective Separation.* We consider two aspects, separations between dual objectives like Büchi and coBüchi (Tables 3 and 4), and separations between conjunction and disjunction of objectives (Table 5). We compare dual objectives in two ways: (i) we show that single objectives that are dual to each other behave differently when we consider disjunction for each of them and (ii) we compare conjunctive objectives and their dual disjunctive objectives. For (ii) we have that conjunctive Büchi objectives are dual to disjunctive coBüchi objectives, and Streett objectives are dual to Rabin objectives.

   (a) *Separating Dual Objectives in Graphs.* In graphs we have that for reachability objectives disjunction is in linear time while for disjunctive safety objectives we establish an $\Omega(kn^{2-o(1)})$ lower bound under assumption (A1). Analogous results hold for Büchi and coBüchi objectives. Further, conjunctive Büchi objectives are in linear time and thus can be separated from their dual objective, the disjunctive coBüchi objectives. Finally, for Streett objectives in graphs with $b = O(n^2/\log n)$ we have an $O(n^2)$ algorithm while we establish an $\Omega(n^{3-o(1)})$ lower bound for Rabin objectives when $k = \Theta(n)$.

   (b) *Separating Dual Objectives in MDPs.* On MDPs disjunctive Büchi objectives are in time $O(\mathrm{MEC} + b)$, which is in $O(\min(n^2, m^{1.5}) + nk)$, while for coBüchi objectives we show $\Omega(kn^{2-o(1)})$ and $\Omega(m^{2-o(1)})$ conditional lower bounds for both disjunctive queries and disjunctive objectives. This separates the two objectives for both sparse and dense graphs. Further conjunctive Büchi objectives can be solved in $O(\mathrm{MEC} + b)$ time and thus there is also a separation between disjunctive coBüchi objectives and their dual. Finally, for Streett objectives in MDPs with $b = O(\min(n^2, m^{1.5})/\log n)$ we show both an $O(n^2)$-time and an $O(m^{1.5})$-time algorithm while we establish $\Omega(n^{3-o(1)})$ and $\Omega(m^{2-o(1)})$ conditional lower bounds for Rabin objectives when $k = \Theta(n)$.

   (c) *Separating Conjunction and Disjunction in Graphs and MDPs.* Except for reachability, i.e., in particular for all considered polynomial-time problems, we observe that the disjunction of objectives is computationally harder than the conjunction of these objectives (under assumptions (A1), (A2)). First, for safety objectives conjunction is in linear time even for MDPs while for disjunctive queries (disjunctive objectives are PSPACE-complete) we present $\Omega(kn^{2-o(1)})$ and $\Omega(m^{2-o(1)})$ conditional lower bounds, where the first bound also holds for graphs. Second, for Büchi and coBüchi objectives conjunction is in $O(\mathrm{MEC}+b)$ on MDPs (and $O(m+b)$ on graphs) while we show $\Omega(kn^{2-o(1)})$ and $\Omega(m^{2-o(1)})$ conditional lower bounds for disjunctive coBüchi objectives and disjunctive Büchi / coBüchi queries on MDPs. Further, for coBüchi objectives our $\Omega(kn^{2-o(1)})$ bound also holds on graphs, which separates conjunction and disjunction also in this setting. Third, corollaries of our results are that for each of one-pair Streett objectives and one-pair Rabin objectives their disjunction is harder than their conjunction.

Remark about Streett and Rabin objective separation. One remarkable aspect of our objective separation result is that we achieve it for Rabin and Streett objectives (both in graphs and MDPs), which are dual. In more general models such as games on graphs, Rabin objectives are NP-complete and Streett objectives are coNP-complete [21]. In graphs and MDPs, both Rabin and Streett objectives can be solved in polynomial time. Since Rabin and Streett objectives are dual, and they belong to the complementary complexity classes (either both in P, or one is NP-complete, other coNP-complete), they were considered to be equivalent for algorithmic purposes for graphs and MDPs. Quite surprisingly we show that under some widely believed assumptions, both for MDPs and graphs, Rabin objectives are algorithmically harder than Streett objectives.

Table 5. Separating Conjunction and Disjunction.

		Conjunction	Disjunction
Safety	Graphs	$m + nk$	$\mathbf{k \cdot n^{2-o(1)}}$
	MDP Qu.	$m + nk$	$\mathbf{k \cdot n^{2-o(1)}}, \mathbf{m^{2-o(1)}}$
Büchi	MDPs Qu.	$\min(n^2, m^{1.5}) + nk$	$\mathbf{k \cdot n^{2-o(1)}}, \mathbf{m^{2-o(1)}}$
coBüchi	Graphs	$m + nk$	$\mathbf{k \cdot n^{2-o(1)}}$
	MDPs	$\min(n^2, m^{1.5}) + nk$	$\mathbf{k \cdot n^{2-o(1)}}, \mathbf{m^{2-o(1)}}$
1-pair Streett	Graphs	$n^2 + nk \log n$	$\mathbf{k \cdot n^{2-o(1)}}$
	MDPs	$^*\mathbf{min(n^2, m^{1.5}) + nk}$	$\mathbf{k \cdot n^{2-o(1)}}, \mathbf{m^{2-o(1)}}$
1-pair Rabin	Graphs	$m + nk$	$\mathbf{k \cdot n^{2-o(1)}}$
	MDPs	$\min(n^2, m^{1.5}) + nk$	$\mathbf{k \cdot n^{2-o(1)}}, \mathbf{m^{2-o(1)}}$

* $\log n$ factor omitted

Remark about separating conjunction and disjunction. In logic disjunction and conjunction are dual and for polynomial-time problems to the best of our knowledge there have not been any results which show that one is harder than the other. For reachability objectives the conjunctive problems are harder (NP-complete for graphs, PSPACE-complete for MDPs) compared to the disjunctive problems, which are in polynomial time. In terms of strategy complexity, again conjunctive objectives are harder than disjunctive objectives: While for disjunctive Büchi objectives memoryless strategies suffice, for conjunctive Büchi objectives strategies require memory; and while for Rabin objectives memoryless strategies suffice, for Streett objectives strategies require memory (even exponential memory in games). Given the existing results, there was no evidence to expect that when polynomial-time algorithms exist that disjunction is harder than conjunction. On the contrary, existing results show that some aspects of conjunctive objectives are harder than the disjunctive counterpart. Surprisingly, our results indicate that from an algorithmic point of view several polynomial-time problems are harder for the disjunctive problems than for their conjunctive counterparts.

Technical contributions.

Algorithms. (1) We show that given the MEC-decomposition of an MDP, the *almost-sure reachability problem* can be solved in linear time on the MDP where each MEC is contracted to a player 1 vertex. This yields to the improved algorithms for disjunctive queries of reachability and Büchi objectives on MDPs. (2) For *MDPs* with *disjunctive coBüchi* objectives and disjunctive queries of coBüchi objectives we use the MEC-decomposition in a different way; namely, we show that it is sufficient to do a linear-time computation in each MEC per coBüchi objective to solve both disjunctive questions. (3) Further we show that for *graphs* with a *disjunctive coBüchi* objective for which the target set of each of the single coBüchi objectives has *cardinality one* the problem can be solved with a breadth-first search like algorithm in linear time. (4) Finally, we provide faster algorithms for *MDPs with Streett objectives*. The straight-forward algorithm repeatedly computes MEC-decompositions in a black-box manner; we show that one can open this black-box and combine the current best algorithms for MEC-decomposition [14] and graphs with Streett objectives [19, 25] to achieve almost the same running time for MDPs with Streett objectives as for graphs.

Conditional Lower Bounds. (a) Conjecture (A1) is equivalent to the conjecture that there is no combinatorial $O(n^{3-\varepsilon})$ time algorithm to detect whether an n-vertex graph contains a triangle [36]. We show that triangle-detection in graphs can be linear-time reduced to *disjunctive queries of almost-sure reachability* in MDPs and thus that the latter is hard assuming (A1). (b) For the hardness under (A2) we consider the intermediate problem Orthogonal Vectors, which is known to be hard under (A2) [37], and linear-time reduce it to *disjunctive queries of almost-sure reachability* in MDPs. (c) For *disjunctive safety problems* we give a linear-time reduction from triangle-detection that only requires player 1 vertices and thus hardness also holds in graphs when assuming (A1). (d) However, the reduction we give from Orthogonal Vectors to *disjunctive safety problems* requires random vertices and thus hardness under (A2) only holds on MDPs. (e) We then exploit reductions between the different types of objectives to obtain the hardness results for Büchi, coBüchi, and Rabin.

Outline. In Section 2 we provide formal definitions and state the conjectures on which the conditional lower bounds are based. In Section 3 we first describe the conditional lower bounds for disjunctive reachability queries in MDPs and the disjunction of safety objectives in graphs and MDPs and then outline how other conditional lower bounds follow from these results. In Section 4 we summarize our algorithmic results. All details can be found in the full version.

2. Preliminaries

Markov Decision Processes (MDPs) and Graphs. An MDP $P = ((V, E), (V_1, V_R), \delta)$ consists of a finite directed graph with vertices V and edges E with a partition of the vertices into *player 1 vertices* V_1 and *random vertices* V_R and a probabilistic transition function δ. We call an edge (u, v) with $u \in V_1$ *player 1 edge* and an edge (v, w) with $v \in V_R$ a *random edge*. The probabilistic transition function is a function from V_R to $\mathcal{D}(V)$, where $\mathcal{D}(V)$ is the set of probability distributions over V and a random edge $(v, w) \in E$ if and only if $\delta(v)[w] > 0$. Graphs are a special case of MDPs with $V_R = \emptyset$.

Plays and Strategies. A *play* or infinite path in P is an infinite sequence $\omega = \langle v_0, v_1, v_2, \ldots\rangle$ such that $(v_i, v_{i+1}) \in E$ for all $i \in \mathbb{N}$; we denote by Ω the set of all plays. A player 1 *strategy* $\sigma : V^* \cdot V_1 \to V$ is a function that assigns to every finite prefix $\omega \in V^* \cdot V_1$ of a play that ends in a player 1 vertex v a successor vertex $\sigma(\omega) \in V$ such that there exists an edge $(v, \sigma(\omega)) \in E$; we denote by Σ the set of all player 1 strategies. A strategy is *memoryless* if we have $\sigma(\omega) = \sigma(\omega')$ for any $\omega, \omega' \in V^* \cdot V_1$ that end in the same vertex $v \in V_1$.

Objectives and Almost-Sure Winning Sets. An *objective* ψ is a subset of Ω said to be winning for player 1. We say that a play $\omega \in \Omega$ *satisfies the objective* if $\omega \in \psi$. For any measurable set of plays $A \subseteq \Omega$ we denote by $\Pr_v^\sigma(A)$ the probability that a play starting at $v \in V$ belongs to A when player 1 plays strategy σ. A strategy σ is *almost-sure winning* from a vertex $v \in V$ for an objective ψ if $\Pr_v^\sigma(\psi) = 1$. In graphs the existence of an almost-sure winning strategy corresponds to the existence of a play in the objective. The *almost-sure winning set* $\langle\!\langle 1 \rangle\!\rangle_{as}(P, \psi)$ of player 1 is the set of vertices for which player 1 has an almost-sure winning strategy. Let $\mathrm{Inf}(\omega)$ for $\omega \in \Omega$ denote the set of vertices that occurs infinitely often in ω.

Reachability For a vertex set $T \subseteq V$ the reachability objective is the set of infinite paths that contain a vertex of T, i.e., $\mathrm{Reach}(T) = \{\langle v_0, v_1, v_2, \ldots\rangle \in \Omega \mid \exists j \geq 0 : v_j \in T\}$.

Safety For a vertex set $T \subseteq V$ the safety objective is the set of infinite paths that *do not contain* any vertex of T, i.e., $\mathrm{Safety}(T) = \{\langle v_0, v_1, v_2, \ldots\rangle \in \Omega \mid \forall j \geq 0 : v_j \notin T\}$.

Büchi For a vertex set $T \subseteq V$ the Büchi objective is the set of infinite paths in which a vertex of T occurs *infinitely often*, i.e., $\text{Büchi}(T) = \{\omega \in \Omega \mid \mathrm{Inf}(\omega) \cap T \neq \emptyset\}$.

coBüchi For a vertex set $T \subseteq V$ the coBüchi objective is the set of infinite paths for which *no* vertex of T occurs *infinitely often*, i.e., $\text{coBüchi}(T) = \{\omega \in \Omega \mid \mathrm{Inf}(\omega) \cap T = \emptyset\}$.

Streett Given a set SP of k pairs (L_i, U_i) of vertex sets $L_i, U_i \subseteq V$ with $1 \leq i \leq k$, the Streett objective is the set of infinite paths for which it holds *for each* $1 \leq i \leq k$ that whenever a vertex of L_i occurs infinitely often, then a vertex of U_i occurs infinitely often, i.e., $\mathrm{Streett}(\mathrm{SP}) = \{\omega \in \Omega \mid L_i \cap \mathrm{Inf}(\omega) = \emptyset \text{ or } U_i \cap \mathrm{Inf}(\omega) \neq \emptyset \text{ for all } 1 \leq i \leq k\}$.

Rabin Given a set RP of k pairs (L_i, U_i) of vertex sets $L_i, U_i \subseteq V$ with $1 \leq i \leq k$, the Rabin objective is the set of infinite paths for which there *exists* an i, $1 \leq i \leq k$, such that a vertex of L_i occurs infinitely often but no vertex of U_i occurs infinitely often, i.e., $\mathrm{Rabin}(\mathrm{RP}) = \{\omega \in \Omega \mid L_i \cap \mathrm{Inf}(\omega) \neq \emptyset \text{ and } U_i \cap \mathrm{Inf}(\omega) = \emptyset \text{ for some } 1 \leq i \leq k\}$.

Given c objectives $\psi_1, \ldots, \psi_c$, the *conjunctive objective* $\psi = \psi_1 \cap \ldots \cap \psi_c$ is given by the intersection of the c objectives, and the *disjunctive objective* $\psi = \psi_1 \cup \ldots \cup \psi_c = \bigvee_{i=1}^{c} \psi_i$ is given by the union of the c objectives. For the *conjunctive query* of c objectives $\psi_1, \ldots, \psi_c$ we define the (almost-sure) winning set to be the set of vertices that have one strategy that is (almost-sure) winning for *each of* the objectives $\psi_1, \ldots, \psi_c$. Analogously, a vertex is in the (almost-sure) winning set $\bigvee_{i=1}^{c} \langle\!\langle 1 \rangle\!\rangle_{as}(P, \psi_i)$ for the *disjunctive query* of the c objectives if it is in a (almost-sure) winning set for *at least one* of the c objectives (i.e. we take the union of the winning sets).

2.1 Conjectured Lower Bounds

While classical complexity results are based on standard complexity-theoretical assumptions, e.g., $\mathsf{P} \neq \mathsf{NP}$, polynomial lower bounds are often based on widely believed, conjectured lower bounds about well studied algorithmic problems. Our lower bounds will be conditioned on the popular conjectures discussed below.

First, we consider conjectures on Boolean matrix multiplication [1, 36] and triangle detection [1] in graphs, which build the basis for our lower bounds on dense graphs. A triangle in a graph is a triple x, y, z of vertices such that $(x, y), (y, z), (z, x) \in E$.

Conjecture 2.1 (Combinatorial Boolean Matrix Multiplication Conjecture (BMM))**.** *There is no $O(n^{3-\varepsilon})$ time combinatorial algorithm for computing the boolean product of two $n \times n$ matrices for any $\varepsilon > 0$.*

Conjecture 2.2 (Strong Triangle Conjecture (STC))**.** *There is no $O(n^{3-\varepsilon})$ time combinatorial algorithm that can detect whether a graph contains a triangle for any $\varepsilon > 0$.*

BMM is equivalent to STC [36]. A weaker assumption, without the restriction to combinatorial algorithms, is that detecting a triangle in a graph takes super-linear time.

Second, we consider the Strong Exponential Time Hypothesis [11, 28] and the Orthogonal Vectors Conjecture [6], the former dealing with satisfiability in propositional logic and the latter with the *Orthogonal Vectors Problem*.

The Orthogonal Vectors Problem (OV). Given two sets S_1, S_2 of d-bit vectors with $|S_i| \leq N$, $d \in \Theta(\log N)$, are there $u \in S_1$ and $v \in S_2$ such that $\sum_{i=1}^{d} u_i \cdot v_i = 0$?

Conjecture 2.3 (Strong Exponential Time Hypothesis (SETH))**.** *For each $\varepsilon > 0$ there is a k such that k-CNF-SAT on n variables and m clauses cannot be solved in $O(2^{(1-\varepsilon)n}\,\mathrm{poly}(m))$ time.*

Conjecture 2.4 (Orthogonal Vectors Conjecture (OVC))**.** *There is no $O(N^{2-\varepsilon})$ time algorithm for OV for any $\varepsilon > 0$.*

SETH implies OVC [37], i.e., whenever a problem is hard assuming OVC, it is also hard when assuming SETH. Hence, it is preferable to use OVC for proving lower bounds. Finally, to

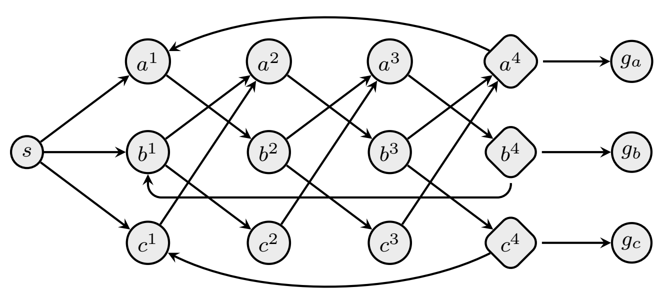

Figure 1. Illustration of Reduction 3.2, with $G = (\{a, b, c\}, \{(a, b), (b, a), (b, c), (c, a)\})$. Vertices drawn as cycle are owned by player 1, vertices drawn as diamond are random vertices.

the best of our knowledge, no relations between the former two conjectures and the latter two conjectures are known.

Remark 2.5. *The conjectures that no* polynomial *improvements over the best known running times are possible do not exclude improvements by sub-polynomial factors such as poly-logarithmic factors or factors of, e.g., $2^{\sqrt{\log n}}$ as in [39].*

3. Conditional Lower Bounds

In this section we present all conditional lower bounds.

3.1 Disjunctive Reachability in MDPs

We first present our lower bound for dense MDPs based on STC.

Theorem 3.1. *There is no combinatorial $O(n^{3-\epsilon})$ or $O((k \cdot n^2)^{1-\epsilon})$ algorithm (for any $\epsilon > 0$) for disjunctive reachability queries in MDPs under Conjecture 2.2 (i.e., unless STC and BMM fail). The bounds hold for dense MDPs with $m = \Theta(n^2)$.*

The above theorem is by the following reduction from the triangle detection problem.

Reduction 3.2. *Given an instance of triangle detection, i.e., a graph $G = (V, E)$, we build the following MDP P.*

- *The vertices V' of P are given by four copies V^1, V^2, V^3, V^4 of V, a start vertex s, and absorbing vertices $F = \{g_v \mid v \in V\}$. The edges E' of P are defined as follows: There is an edge from s to the first copy $v^1 \in V^1$ of every $v \in V$ and the last copy $v^4 \in V^4$ of every $v \in V$ is connected to its first copy v^1 and its corresponding absorbing vertex $g_v \in F$; further for $1 \leq i \leq 3$ there is an edge from v^i to u^{i+1} iff $(v, u) \in E$.*
- *The set of vertices V' is partitioned into player 1 vertices $V'_1 = \{s\} \cup V^1 \cup V^2 \cup V^3 \cup F$ and random vertices $V'_R = V^4$. Moreover, the probabilistic transition function for each vertex $v \in V'_R$ chooses among v's successors with equal probability $1/2$ each.*

The reduction is illustrated in Figure 1. Next we prove that Reduction 3.2 is indeed a valid reduction from triangle detection to disjunctive reachability queries in MDPs.

Lemma 3.3. *A graph G has a triangle iff s is contained in $\bigvee_{v \in V} \langle\langle 1 \rangle\rangle_{as} (P, Reach\,(T_v))$, where P is the MDP given by Reduction 3.2 and $T_v = \{g_v\}$ for $v \in V$.*

Proof. For the only if part assume that G has a triangle with vertices a, b, c and let a^i,b^i,c^i be the copies of a, b, c in V^i. Now a strategy for player 1 in the MDP P to reach g_a with probability 1 is as follows: When in s, go to a^1; when in a^1, go to b^2; when in b^2, go to c^3; when in c^3, go to a^4. As a, b, c form a triangle, all the edges required by the above strategy exist. When player 1 starts in s and follows the above strategy the only random vertex he encounters is a^4. The random choice sends him to the target vertex g_a and to vertex a^1 with probability $1/2$ each. In the former case he is done, in the latter case he continues playing his strategy and will reach a^4 again after three steps. The probability that player 1 has reached g_a after $3q + 1$ steps is $1 - (1/2)^q$ which converges to 1 with q going to infinity. Thus we have found a strategy to reach g_a with probability 1.

For the if part assume that $s \in \bigvee_{v \in V} \langle\langle 1 \rangle\rangle_{as} (P, \text{Reach}\,(T_v))$. That is, there is an $a \in V$ such that $s \in \langle\langle 1 \rangle\rangle_{as} (P, \text{Reach}\,(T_a))$. Let us consider a corresponding strategy for reaching $T_a = \{g_a\}$. First, assume that the strategy would visit a vertex v^4 for $v \in V \setminus \{a\}$. Then with probability $1/2$ player 1 would end up in the vertex g_v which has no path to g_a, a contradiction to $s \in \langle\langle 1 \rangle\rangle_{as} (P, \text{Reach}\,(T_a))$. Thus the strategy has to avoid visiting vertices v^4 for $v \in V \setminus \{a\}$. Second, as the only way to reach g_a is a^4, the strategy has to choose a^4. But then with probability $1/2$ it will be send to a^1 and there must be a path from a^1 to g_a that doesn't not cross $V^4 \setminus \{a^4\}$. By the latter this path must be of the form a^1, b^2, c^3, a^4, g_a for some $b, c \in V$. Now by the construction of G' in the MDP P the vertices a, b, c form a triangle in the original graph G. □

The size and the construction time of the MDP P, constructed by Reduction 3.2, is linear in the size of the original graph G and we have $k = \Theta(n)$ target sets. Thus if we would have a combinatorial $O(n^{3-\epsilon})$ or $O((k \cdot n^2)^{1-\epsilon})$ algorithm for disjunctive queries of reachability objectives in MDPs for any $\epsilon > 0$, we would immediately get a combinatorial $O(n^{3-\epsilon})$ algorithm for triangle detection, which contradicts STC and BMM.

Next we present a lower bound for sparse MDPs based on OVC and SETH.

Theorem 3.4. *There is no $O(m^{2-\epsilon})$ or $O((k \cdot m)^{1-\epsilon})$ algorithm (for any $\epsilon > 0$) for disjunctive reachability queries in MDPs under Conjecture 2.4 (i.e., unless OVC and SETH fail).*

To prove the above we give a reduction from OVC to disjunctive reachability queries in MDPs.

Reduction 3.5. *Given two sets S_1, S_2 of d-dimensional vectors, we build the following MDP P.*

- *The vertices V of the MDP P are given by a start vertex s, vertices S_1 and S_2 representing the sets of vectors, vertices $\mathcal{C} = \{c_i \mid 1 \leq i \leq d\}$ representing the coordinates, and absorbing vertices $F = \{g_v \mid v \in S_2\}$. The edges E of P are defined as follows: the start vertex s has an edge to every vertex of S_1 and every vertex $v \in S_2$ has an edge to s and to its corresponding absorbing vertex $g_v \in F$; further for each $x \in S_1$ there is an edge* to *$c_i \in \mathcal{C}$ iff $x_i = 1$ and for each $y \in S_2$ there is an edge* from *$c_i \in \mathcal{C}$ iff $y_i = 0$.*
- *The set of vertices V is partitioned into player 1 vertices $V_1 = \{s\} \cup \mathcal{C} \cup F$ and random vertices $V_R = S_1 \cup S_2$. The probabilistic transition function for each vertex $v \in V_R$ chooses among v's successors uniformly at random.*

The reduction is illustrated on an example in Figure 2.

Lemma 3.6. *There exist orthogonal vectors $x \in S_1$, $y \in S_2$ iff $s \in \bigvee_{v \in V} \langle\langle 1 \rangle\rangle_{as} (P, Reach\,(T_v))$ where P is the MDP given by Reduction 3.5 and $T_v = \{g_v\}$ for $v \in V$.*

Proof. If one of the sets S_1 and S_2 contains the all-zero vector, an orthogonal pair of vectors exists trivially and this can be detected in

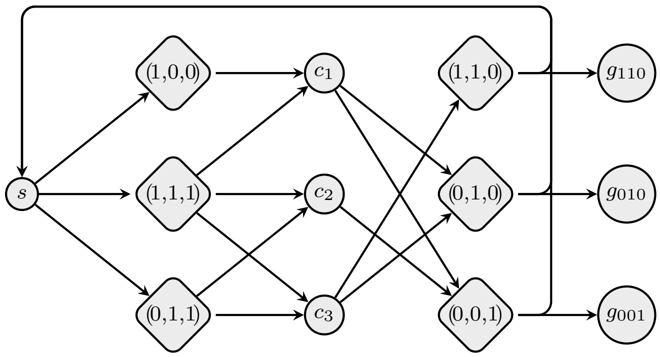

Figure 2. Illustration of Reduction 3.5 for $S_1 = \{(1,0,0), (1,1,1), (0,1,1)\}$ and $S_2 = \{(1,1,0), (0,1,0), (0,0,1)\}$.

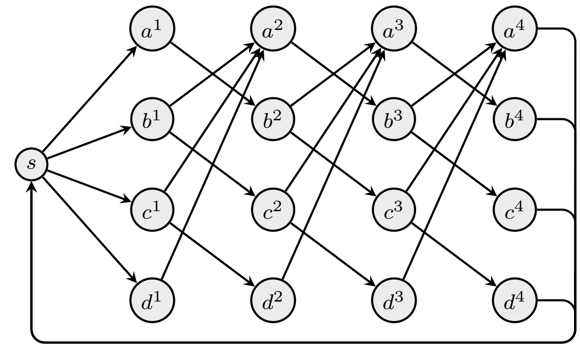

Figure 3. Illustration of Reduction 3.8, with $G = (\{a,b,c,d\}, \{(a,b),(b,a),(b,c),(c,a),(c,d),(d,a)\})$. The target sets for disjunctive safety are $T_a = \{b^1, c^1, d^1, b^4, c^4, d^4\}$, $T_b = \{a^1, c^1, d^1, a^4, c^4, d^4\}$, $T_c = \{a^1, b^1, d^1, a^4, b^4, d^4\}$, and $T_d = \{a^1, b^1, c^1, a^4, b^4, c^4\}$.

linear time; thus we assume w.l.o.g. that none of the sets contains the all-zero vector.

For the only if part assume that there are orthogonal vectors $x \in S_1$, $y \in S_2$. Now a strategy for player 1 in the MDP P to reach g_y with probability 1 is as follows: When in s, go to x; when in some $c \in \mathcal{C}$, go to y. As x and y are orthogonal, each $c_i \in \mathcal{C}$ reachable from x has an edge to y, i.e., for $x_i = 1$ it must be that $y_i = 0$. When player 1 starts in s and follows the above strategy, he reaches y after three steps. There the random choice sends him to the target vertex g_y and back to vertex y with probability $1/2$ each. In the former case he is done, in the latter case he continues playing his strategy and will reach y again after three steps. The probability that player 1 has reached g_y after $3q$ steps is $1 - (1/2)^q$, which converges to 1 with q going to infinity. Thus we have found a strategy to reach g_y with probability 1.

For the if part assume that $s \in \bigvee_{v \in V} \langle\!\langle 1 \rangle\!\rangle_{as}(P, \text{Reach}(T_v))$. That is, there is an $y \in S_2$ such that $s \in \langle\!\langle 1 \rangle\!\rangle_{as}(P, \text{Reach}(T_y))$. Let us consider a corresponding strategy for reaching $T_y = \{g_y\}$. First, assume that the strategy would visit a vertex $y' \in S_2$ for $y' \neq y$. Then with probability $1/2$ the player would end up in the vertex $g_{y'}$ which has no path to g_y, a contradiction to $s \in \langle\!\langle 1 \rangle\!\rangle_{as}(P, \text{Reach}(T_y))$. Thus the strategy has to avoid visiting vertices $S_2 \setminus \{y\}$. Second, as the only way to reach g_y is y, the strategy has to choose y. But then with probability $1/2$ it will be send to s and thus there must be a strategy to reach g_y from s with probability 1 that does not cross $S_2 \setminus \{y\}$. As y is the only predecessor of g_y, there must also be such a strategy to reach y. In other words, there must be an $x \in S_1$ such that for each successor $c_i \in \mathcal{C}$ there is an edge to y. By the construction of the MDP P this is equivalent to the existence of an $x \in S_1$ such that whenever $x_i = 1$ then $y_i = 0$, and thus x and y are orthogonal vectors. □

The number of vertices in P, constructed by Reduction 3.5, is $O(N)$ and the construction can be performed in $O(N \log N)$ time (recall that $d \in O(\log N)$). The number of edges m is $O(N \log N)$ (thus we consider P to be a sparse MDP) and the number of target sets $k \in \Theta(N) = \theta(m / \log N)$. Finally, if we would have an $O(m^{2-\epsilon})$ or $O((k \cdot m)^{1-\epsilon})$ algorithm for disjunctive reachability queries in MDPs for any $\epsilon > 0$, we would immediately get an $O(N^{2-\epsilon})$ algorithm for OV, which contradicts OVC (and thus SETH).

3.2 Safety Objectives

We first present a lower bound for disjunctive safety based on STC that even holds on graphs.

Theorem 3.7. *There is no combinatorial $O(n^{3-\epsilon})$ or $O((k \cdot n^2)^{1-\epsilon})$ algorithm (for any $\epsilon > 0$) for disjunctive safety (objectives or queries) in graphs under Conjecture 2.2 (i.e., unless STC and BMM fail).*

The above is by the linear time reduction from triangle detection to disjunctive safety in graphs provided below.

Reduction 3.8. *Given a graph $G = (V, E)$ (for triangle detection), we build a graph $G' = (V', E')$ (for disjunctive safety) as follows. As vertices V' we have four copies V^1, V^2, V^3, V^4 of V and a vertex s. A vertex $v^i \in V^i$ has an edge to a vertex $u^{i+1} \in V^{i+1}$ iff $(v, u) \in E$. Finally, s has an edge to all vertices in V^1 and all vertices in V^4 have an edge to s.*

Reduction 3.8 is illustrated in Figure 3.

Lemma 3.9. *Let G' be the graph given by Reduction 3.8 for a graph G and let $T_v = (V^1 \setminus \{v^1\}) \cup (V^4 \setminus \{v^4\})$. Then G has a triangle iff s is in the winning set of $(G', \bigvee_{v \in V} \text{Safety}(T_v))$.*

Proof. For the only if part assume that G has a triangle with vertices a, b, c and let a^i,b^i,c^i be the copies of a, b, c in V^i. Now a strategy for player 1 in G' to satisfy Safety (T_a) is as follows: When in s, go to a^1; when in a^1, go to b^2; when in b^2, go to c^3; when in c^3, go to a^4; and when in a^4, go to s. As a, b, c form a triangle, all the edges required by the above strategy exist. When player 1 starts in s and follows the above strategy, then he plays an infinite path that only uses vertices s, a^1, b^2, c^3, a^4 and thus satisfies Safety (T_a).

For the if part assume that there is a winning play starting in s and satisfying Safety (T_a). Starting from s, this play has to first go to a^1, as all other successors of s would violate the safety constraint. Then the play continues on some vertex $b^2 \in V^2$ and $c^3 \in V^3$ and then, again by the safety constraint, has to enter a^4. Now by construction of G' we know that there must be edges $(a,b), (b,c), (c,a)$ in the original graph G, i.e. there is a triangle in G. □

The size and the construction time of the graph G', constructed by Reduction 3.8, is linear in the size of the original graph G and we have $k = \Theta(n)$ target sets. Thus if we would have a combinatorial $O(n^{3-\epsilon})$ or $O((k \cdot n^2)^{1-\epsilon})$ algorithm for disjunctive safety objectives or queries in graphs, we would immediately get a combinatorial $O(n^{3-\epsilon})$ algorithm for triangle detection, which contradicts STC (and thus BMM).

The above reduction uses a linear number of safety constraints which are all of linear size. Thus, a natural question is whether smaller safety sets would make the problem any easier. Next we

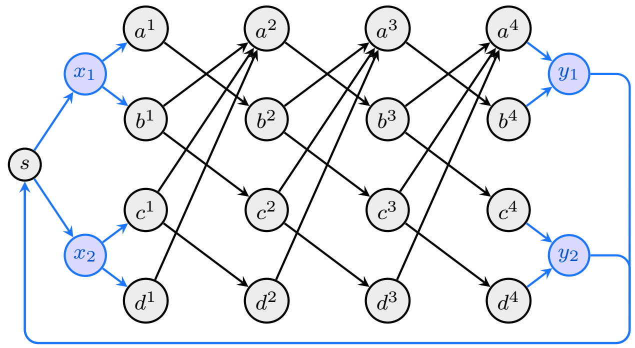

Figure 4. Illustration of how to reduce the number of entries in the target sets in Reduction 3.8 with two complete binary trees. Here $G = (\{a, b, c, d\}, \{(a, b), (b, a), (b, c), (c, a), (c, d), (d, a)\})$ and the target sets for disjunctive safety are $T_a = \{b^1, x_2, b^4, y_2\}$, $T_b = \{a^1, x_2, a^4, y_2\}$, $T_c = \{d^1, x_1, d^4, y_1\}$, and $T_d = \{c^1, x_1, c^4, y_1\}$.

argue that our result even holds for safety sets that are of logarithmic size. To this end we modify Reduction 3.8 as follows. We remove all edges incident to s and replace them by two complete binary trees. The first tree with s as root and the vertices V^1 as leaves is directed towards the leaves, the second tree with root s and leaves V^4 is directed towards s. Now for each pair v^1, v^4 one can select one vertex of each level of the trees (except for the root levels) for the set T_v such that the only safe path starting in s has to use v^1 and each safe path to s must pass v^4. As the depth of the trees is logarithmic in the number of leaf vertices, we get sets of logarithmic size. The construction with the binary trees is illustrated in Figure 4.

Next we present an $\Omega(m^{2-o(1)})$ lower bound for disjunctive objective/query safety in sparse MDPs.

Theorem 3.10. *There is no $O(m^{2-\epsilon})$ or $O((k \cdot m)^{1-\epsilon})$ algorithm (for any $\epsilon > 0$) for disjunctive safety objectives/queries in MDPs under Conjecture 2.4 (i.e., unless OVC & SETH fail).*

To prove the above, we give a linear time reduction from OV to disjunctive safety objectives/queries.

Reduction 3.11. *Given two sets S_1, S_2 of d-dimensional vectors, we build the following MDP P.*

- *The vertices V of the MDP P are given by a start vertex s, vertices S_1 and S_2 representing the sets of vectors, and vertices $\mathcal{C} = \{c_i \mid 1 \le i \le d\}$ representing the coordinates. The edges E of P are defined as follows: the start vertex s has an edge to every vertex of S_1 and every vertex $v \in S_2$ has an edge to s; further for each $x \in S_1$ there is an edge* to *$c_i \in \mathcal{C}$ iff $x_i = 1$ and for each $y \in S_2$ there is an edge* from *$c_i \in \mathcal{C}$ iff $y_i = 1$.*
- *The set of vertices V is partitioned into player 1 vertices $V_1 = \{s\} \cup S_2$ and random vertices $V_R = S_1 \cup \mathcal{C}$. Moreover, the probabilistic transition function for each vertex $v \in V_R$ chooses among v's successors uniformly at random.*

The reduction is illustrated on an example in Figure 5.

Lemma 3.12. *Given two sets S_1, S_2 of d-dimensional vectors, the corresponding MDP P given by Reduction 3.11 and $T_v = \{v\}$ for $v \in S_2$ the following statements are equivalent*

1. *There exist orthogonal vectors $x \in S_1$, $y \in S_2$.*
2. *$s \in \bigvee_{v \in S_2} \langle\langle 1 \rangle\rangle_{as} (P, \text{Safety}(T_v))$*
3. *$s \in \langle\langle 1 \rangle\rangle_{as} \left(P, \bigvee_{v \in S_2} \text{Safety}(T_v)\right)$*

Proof. W.l.o.g. we assume that the 1-vector, i.e., the vector with all coordinates being 1, is contained in S_2 (adding the 1-vector does not change the result of the OV instance). Then a play in the MDP P proceeds as follows. Starting from s, player 1 chooses a vertex $x \in S_1$; then a vertex $c \in \mathcal{C}$ and then a vertex $y \in S_2$ are picked randomly; then the play goes back to s, starting another cycle of the play.

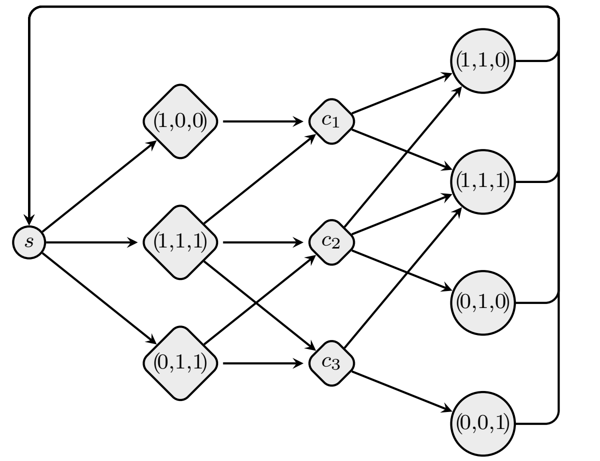

Figure 5. Illustration of Reduction 3.11, for $S_1 = \{(1, 0, 0), (1, 1, 1), (0, 1, 1)\}$ and $S_2 = \{(1, 1, 0), (1, 1, 1), (0, 1, 0), (0, 0, 1)\}$.

(1)⇒(2): Assume there are orthogonal vectors $x \in S_1$, $y \in S_2$. Now player 1 can satisfy Safety (T_y) in the MDP P by simply going to x whenever the play is in s. The random player will then send it to some adjacent $c \in \mathcal{C}$ and then to some adjacent vertex in S_2, but as x and y are orthogonal, this c is not connected to y. Thus the play will never visit y.

(2)⇒(3): Assume $s \in \bigvee_{v \in S_2} \langle\langle 1 \rangle\rangle_{as} (P, \text{Safety}(T_v))$. Then there is a vertex $y \in S_2$ such that $s \in \langle\langle 1 \rangle\rangle_{as} (P, \text{Safety}(T_y))$. Now we can enlarge the objective to $\bigvee_{v \in S_2} \text{Safety}(T_v)$ and obtain $s \in \langle\langle 1 \rangle\rangle_{as} \left(P, \bigvee_{v \in S_2} \text{Safety}(T_v)\right)$.

(3)⇒(1): Assume $s \in \langle\langle 1 \rangle\rangle_{as} \left(P, \bigvee_{v \in S_2} \text{Safety}(T_v)\right)$ and consider a corresponding strategy σ. W.l.o.g. we can assume that this strategy is memoryless [34]. Thus whenever the play is in s, it picks a fixed $x \in S_1$ as the next vertex. Assume towards contradiction that there is no orthogonal vector $y \in S_2$ for x. Then for each $y \in S_2$ we have that there is a $c \in \mathcal{C}$ connecting x to y. In each cycle of the play one goes from s to x and then by random choice to some vertex in S_2. By the above, each of the vertices in S_2 has a non-zero probability to be reached in this cycle, which can, for each fixed n, be lower bounded by a constant p. Thus after n cycles in the play with probability at least $p^{|S_2|}$ all vertices in S_2 have been visited and thus none of the safety objectives is satisfied, a contradiction to the assumption that with probability 1 at least one safety objective is satisfied. Thus there must exist a vector $y \in S_2$ orthogonal to x. □

The number of vertices in the MDP P, constructed by Reduction 3.11, is $O(N)$, the number of edges m is $O(N \log N)$ (recall that $d \in O(\log N)$), we have $k \in \Theta(N)$ target sets, and the construction can be performed in $O(N \log N)$ time. Thus, if we would have an $O(m^{2-\epsilon})$ or $O((k \cdot m)^{1-\epsilon})$ algorithm for disjunctive queries or disjunctive objectives of safety objectives for any $\epsilon > 0$, we would immediately get an $O(N^{2-\epsilon})$ algorithm for OV, which contradicts OVC (and thus SETH).

3.3 Discussion on other Conditional Lower Bounds

With the following observations the conditional lower bounds for Rabin objectives and several disjunctive questions for Büchi and coBüchi objectives (see Table 1) follow as corollaries from the results presented for reachability and safety objectives. First there exists a linear time reduction from reachability to Büchi objectives in MDPs [12, Remark 2.3] that also holds for disjunctive queries (see Observation 2.6 in the full version). Second the conditional lower bounds for disjunctive safety objectives and queries actually already hold for the question whether the (almost-sure) winning set is empty. Further we have that the winning set for disjunctive safety objectives (resp. queries) is empty if and only if the winning set for disjunctive coBüchi objectives (resp. queries) with the same target sets is empty. Finally it is easy to see that disjunctive coBüchi objectives are special instances of Rabin objectives.

Corollary 3.13. *Assuming BMM or STC, there is no combinatorial $O(n^{3-\epsilon})$ or $O((k \cdot n^2)^{1-\epsilon})$ algorithm for each of*

1. *disjunctive Büchi queries in MDPs,*
2. *disjunctive coBüchi queries or objectives in graphs, and*
3. *Rabin objectives in graphs.*

Corollary 3.14. *Assuming SETH or OVC, there is no $O(m^{2-\epsilon})$ or $O((k \cdot m)^{1-\epsilon})$ algorithm for each of*

1. *disjunctive Büchi queries in MDPs,*
2. *disjunctive coBüchi objectives or queries in MDPs (even if all target sets have cardinality one), and*
3. *Rabin objectives in MDPs.*

4. Improved Algorithms

Disjunctive Reachability and Büchi Queries in MDPs. The improved algorithm for disjunctive reachability queries in MDPs is based on the following three ideas:

1. In an MDP without end-components, almost-sure reachability can be solved in linear time.
2. For almost-sure reachability either all vertices in a MEC are winning or none.
3. When each MEC of an MDP is contracted to a single vertex, then the resulting MDP has no end-components (other than self-loops).

Thus, given the MEC decomposition of an MDP, disjunctive reachability queries can be answered in linear time per target set. Finally, Büchi objectives can be reduced to reachability objectives in linear time [12, Remark 2.3].

Theorem 4.1. *For an MDP and target sets T_i for $1 \leq i \leq k$ the almost-sure winning set for disjunctive reachability (or Büchi) queries can be computed in $O(km + \mathrm{MEC})$ time.*

MDPs with Streett Objectives. On graphs, the algorithms for Streett objectives with Streett pairs $\mathrm{SP} = \{(L_i, U_i) \mid 1 \leq i \leq k\}$ are based on finding "good" strongly connected subgraphs and then determining which vertices can reach these "good components". A *good component* is a strongly connected subgraph induced by vertices C such that for each $1 \leq i \leq k$ we have $L_i \cap C = \emptyset$ or $U_i \cap C \neq \emptyset$, i.e., if an infinite path traverses exactly the vertices of the good component infinitely often, then this path is winning.

(a) First we show (compare [8]) that we can use a similar approach for MDPs: finding *good end-components*, i.e., end-components X with $L_i \cap X = \emptyset$ or $U_i \cap X \neq \emptyset$ for each $1 \leq i \leq k$, and then compute almost-sure reachability on the union of all good end-components to determine the almost-sure winning set. In a good end-component player 1 has an almost-sure winning strategy that visits with probability 1 all the vertices in the good end-component infinitely often.

(b) The basic algorithm to find good end-components, starting with the MEC-decomposition, repeatedly removes vertices that cannot be in a good end-component, namely sets L_i for which no vertex of U_i is in the same end-component, and then recomputes the MEC-decomposition of the remaining MDP. This algorithm maintains a set of end-components as candidates for good end-components.

(c) To improve upon the basic algorithm, we do not maintain end-components but only sets of vertices without outgoing random edges for which we can guarantee that each good end-component is contained in one candidate. This relaxed invariant allows us, after initializing with the MEC-decomposition, to make progress in each iteration only with computing maximal strongly connected components (SCCs) and removing certain vertices. This algorithm can be seen as "interleaving" the removal of vertices that cannot be in a good end-component and the recomputations of MEC-decompositions.

(d) We then extend the two techniques that lead to the best asymptotic running times on graphs [13], one for dense graphs and one for sparse graphs, to MDPs. Both techniques basically enable to check for strong connectivity faster after the removal of vertices. For dense graphs we use a sparsification technique called *Hierarchical Graph Decomposition* that was introduced by [24] for undirected graphs and extended to directed graphs and game graphs by [14]. For sparse graphs we use "parallel local searches" similar to [25].

Theorem 4.2. *For an MDP with Streett objectives defined by Streett pairs $\mathrm{SP} = \{(L_i, U_i) \mid 1 \leq i \leq k\}$ with $b = \sum_{i=1}^{k}(|L_i| + |U_i|)$ the almost-sure winning set can be computed in $O(\min(n^2, m\sqrt{m \log n}) + b \log n)$ time.*[3]

MDPs with Disjunctive coBüchi Objectives and Queries. For disjunctive coBüchi we want to know whether there is a strategy that visits one of the target sets only finitely often. For this we can test each of the target sets by removing vertices from the MDP and computing MECs and almost-sure reachability in the remaining graph. The improved algorithm is based on the observation that, instead of computing a MEC-decomposition for each target set, we can compute a MEC-decomposition once in the beginning and then check for each MEC and each target set whether a certain set of vertices contains all vertices of the MEC.

Theorem 4.3. *For an MDP and target sets T_i for $1 \leq i \leq k$ the almost-sure winning set for disjunctive coBüchi objectives and queries can be computed in $O(km + \mathrm{MEC})$ time.*

Graphs with Disjunctive Singleton coBüchi Objectives. To compute the winning set for graphs with disjunctive coBüchi objectives that are all singletons, it is sufficient to detect whether a strongly connected graph contains a cycle that does *not contain all* the vertices in the union of the coBüchi target sets. We first take one of the target vertices and check whether there is a cycle not containing this vertex by computing SCCs once. If not, then this vertex s is contained in all cycles of the graph. We then assign 0-1 edge weights to the edges of G such that all edges that lead to target vertices have weight 1 and all other edges have weight 0. Then there exists a cycle not containing all target vertices iff we can go from s to s (using at least one edge) on a path of total weight less than k. We can compute the total weight of the shortest such path in linear time using a breadth-first search like algorithm.

[3] It can also be computed in $O((\mathrm{MEC} + b) \cdot k)$ time, which is faster for some combinations of parameters with $k = O(\log n)$.

Theorem 4.4. *Given a graph and coBüchi objectives with target sets T_i with $|T_i| = 1$ for $1 \leq i \leq k$, the winning set for the disjunction over the coBüchi objectives can be computed in $O(m)$ time.*

5. Conclusion

In this work we present improved algorithms and the first conditional super-linear lower bounds for several fundamental model-checking problems in graphs and MDPs w.r.t. to ω-regular objectives. Our results establish the first model separation results for graphs and MDPs w.r.t. to classical ω-regular objectives, and the first objective separation results both in graphs and MDPs for dual objectives, and the conjunction and disjunction of objectives of the same type. An interesting direction of future work is to consider similar results for other models, such as, games on graphs.

Acknowledgments

K. C., M. H., and W. D. are partially supported by the Vienna Science and Technology Fund (WWTF) through project ICT15-003. K. C. is partially supported by the Austrian Science Fund (FWF) NFN Grant No S11407-N23 (RiSE/SHiNE) and an ERC Start grant (279307: Graph Games). For W. D., M. H., and V. L. the research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 340506.

References

[1] A. Abboud and V. Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *FOCS*, pages 434–443, 2014.

[2] A. Abboud, V. Vassilevska Williams, and O. Weimann. Consequences of faster alignment of sequences. In *ICALP, Proceedings, Part I*, pages 39–51, 2014.

[3] A. Abboud, A. Backurs, and V. Vassilevska Williams. If the current clique algorithms are optimal, so is Valiant's parser. In *FOCS*, pages 98–117, 2015.

[4] A. Abboud, A. Backurs, and V. Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In *FOCS*, pages 59–78, 2015.

[5] A. Abboud, V. Vassilevska Williams, and H. Yu. Matching triangles and basing hardness on an extremely popular conjecture. In *STOC*, pages 41–50, 2015.

[6] A. Abboud, V. Vassilevska Williams, and J. R. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *SODA*, pages 377–391, 2016.

[7] A. Backurs and P. Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In *STOC*, pages 51–58, 2015.

[8] C. Baier and J.-P. Katoen. *Principles of model checking*. MIT Press, 2008.

[9] K. Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails. In *FOCS*, pages 661–670, 2014.

[10] K. Bringmann and M. Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *FOCS*, pages 79–97, 2015.

[11] C. Calabro, R. Impagliazzo, and R. Paturi. The complexity of satisfiability of small depth circuits. In *IWPEC*, pages 75–85, 2009.

[12] K. Chatterjee and L. Doyen. Partial-observation stochastic games: How to win when belief fails. *ACM Trans. Comput. Log.*, 15(2):16, 2014.

[13] K. Chatterjee and M. Henzinger. An $O(n^2)$ time algorithm for alternating Büchi games. In *SODA*, pages 1386–1399, 2012.

[14] K. Chatterjee and M. Henzinger. Efficient and dynamic algorithms for alternating Büchi games and maximal end-component decomposition. *Journal of the ACM*, 61(3):15, 2014.

[15] K. Chatterjee, M. Jurdziński, and T. A. Henzinger. Simple stochastic parity games. In *CSL*, pages 100–113, 2003.

[16] K. Chatterjee, T. A. Henzinger, and N. Piterman. Generalized parity games. In *FOSSACS*, volume 4423, pages 153–167, 2007.

[17] K. Chatterjee, L. Doyen, and T. A. Henzinger. Qualitative analysis of partially-observable Markov decision processes. In *MFCS*, pages 258–269, 2010.

[18] K. Chatterjee, L. de Alfaro, and R. Majumdar. The complexity of coverage. *Int. J. Found. Comput. Sci.*, 24(2):165–186, 2013.

[19] K. Chatterjee, M. Henzinger, and V. Loitzenbauer. Improved algorithms for one-pair and k-pair Streett objectives. In *LICS*, pages 269–280, 2015.

[20] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri. NUSMV: a new symbolic model checker. *Int. J. STTT*, 2:410–425, 2000.

[21] E. A. Emerson and C. S. Jutla. The complexity of tree automata and logics of programs. *SIAM J. Comput.*, 29(1):132–158, 1999.

[22] N. Fijalkow and F. Horn. The surprizing complexity of reachability games. *CoRR*, abs/1010.2420, 2010.

[23] A. Gajentaan and M. H. Overmars. On a class of O(n^2) problems in computational geometry. *Comput. Geom.*, 45(4):140–152, 2012.

[24] M. Henzinger, V. King, and T. Warnow. Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology. *Algorithmica*, 24(1):1–13, 1999.

[25] M. Henzinger and J. A. Telle. Faster algorithms for the nonemptiness of Streett automata and for communication protocol pruning. In *SWAT*, pages 16–27, 1996.

[26] M. Henzinger, S. Krinninger, D. Nanongkai, and T. Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In *STOC*, pages 21–30, 2015.

[27] G. J. Holzmann. The model checker spin. *IEEE Trans. Softw. Eng.*, 23 (5):279–295, May 1997.

[28] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[29] M. Z. Kwiatkowska, G. Norman, and D. Parker. Prism 4.0: Verification of probabilistic real-time systems. In *CAV*, LNCS 6806, pages 585–591, 2011.

[30] F. Le Gall. Powers of tensors and fast matrix multiplication. In *ISSAC*, pages 296–303, 2014.

[31] L. Lee. Fast context-free grammar parsing requires fast boolean matrix multiplication. *J. ACM*, 49(1):1–15, Jan. 2002.

[32] M. Patrascu and R. Williams. On the possibility of faster SAT algorithms. In *SODA*, pages 1065–1075, 2010.

[33] L. Roditty and U. Zwick. On dynamic shortest paths problems. *Algorithmica*, 61(2):389–401, 2011.

[34] W. Thomas. On the synthesis of strategies in infinite games. In *STACS*, LNCS 900, pages 1–13. Springer, 1995.

[35] W. Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, vol. 3, Beyond Words, chapter 7, pages 389–455. Springer, 1997.

[36] V. Vassilevska Williams and R. Williams. Subcubic equivalences between path, matrix and triangle problems. In *FOCS*, pages 645–654, 2010.

[37] R. Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *TCS*, 348(2-3):357–365, 2005.

[38] R. Williams. Faster decision of first-order graph properties. In *CSL-LICS*, pages 80:1–80:6, 2014.

[39] R. Williams. Faster all-pairs shortest paths via circuit complexity. In *STOC*, pages 664–673, 2014.

[40] P. Wolper. Constructing automata from temporal logic formulas: A tutorial. In *Lectures on Formal Methods and Performance Analysis*, pages 261–277, 2000.