Dissertation Title: Automata-Based Quantitative Verification

Suguman Bansal
Advisor: Moshe Y. Vardi
Institute: Rice University, Houston, TX, USA

Abstract. Existing solution approaches for problems in *formal quantitative analysis* suffer from two challenges that adversely impact their theoretical understanding and large-scale applicability. These are the *lack of generalizability*, and *separation-of-techniques*. Lack of generalizability refers to the issue that solution approaches are often specialized to the underlying *cost model* that evaluates the quantitative property. Different cost models deploy such disparate algorithms that there is no transfer of knowledge from one cost model to another. Separation-of-techniques refers to the inherent dichotomy in solving problems in quantitative analysis. Most algorithms comprise of two phases: A *structural phase*, which reasons about the structure of the quantitative system(s) using techniques from automata or graphs; and a *numerical phase*, which reasons about the quantitative dimension/cost model using numerical methods. These techniques are incompatible with one another, forcing the phases to be performed sequentially, thereby impacting scalability.
The dissertation [6] contributes towards a novel framework that addresses the aforementioned challenges. The introduced framework, called *comparator automata* or *comparators* in short, builds on automata-theoretic foundations to generalize across a variety of cost models. The crux of comparators is that they enable automata-based methods in the numerical phase, hence eradicating the dependence on numerical methods. In doing so, comparators are able to integrate the structural and numerical phases. On the theoretical front, we demonstrate that comparator-based solutions have the advantage of generalizable results, and yield complexity-theoretic improvements over a range of problems in quantitative analysis. On the practical front, we demonstrate through empirical analysis that comparator-based solutions render more efficient, scalable, and robust performance, and demonstrate broader applicability than traditional methods for quantitative reasoning.

1 Quantitative analysis

Formal methods are being touted to advance *assured autonomy* by obtaining rigorous guarantees on computational systems from the high-level descriptions of their desired properties. Through diligent research efforts of the past several decades, the *qualitative reasoning* of *logical properties* that describe temporal behaviors, safety, liveness and so on have attained industrial-scale adoption. The notion of correctness in logical properties is, however, Boolean. The demands of current applications are that system properties become richer and more complex, for which new formal methods techniques have to be developed. Recently, *quantitative properties* of systems have emerged in importance across diverse applications. For instance, reinforcement learning-enabled policies are learned from *rewards*, motion-planning tasks are *resource* aware, multi-agent protocols consist of *utility*-maximizing agents, etc. These properties can reason about aspects such as quality measures, cost- and resource- consumption, distance metrics and the like, which logical properties cannot easily express. For example, whether an arbiter grants every request is a logical property; But the promptness with which an arbiter grants requests is a quantitative property. The analysis of quantitative properties of computing systems, or *quantitative analysis* in short, is an emerging area in automated formal analysis. Its applications are wide-ranging, from probabilistic guarantees on hardware or software [4,24,25] to planning for robots with resource-constraints [23], with hard and soft constraints where the soft constraints are expressed quantitatively [26,28] to generating plans of higher quality [2,14]. Yet, quantitative analysis is far behind in scalability to meet the demands of its emerging applications.

1.1 Challenges in quantitative analysis

This dissertation begins with identifying two broad challenges that obstruct the theoretical understanding and algorithmic development in quantitative analysis, thus limiting their scalability.

Challenge 1. Lack of generalizability refers to the issue that solution approaches of problems in quantitative analysis are specialized to the underlying *cost model* which assigns real-valued measures to the quantitative property of interest. Different cost models deploy disparate techniques to solve. This disparity prohibits the transfer of knowledge of solving problems over one cost model to another cost model. Furthermore, owning to the large variety of cost models, soon it will become too cumbersome to digest progress made across cost models.

As a concrete demonstration of the lack of generalizability, consider the problem of *quantitative inclusion over weighted ω-automata. Weighted ω-automaton* [22] are a well-established, finite-state (quantitative) abstraction used to model quantitative systems. An execution is an infinite sequence of labels that arise from an infinite sequence of subsequent transitions. Weighted ω-automata assign a real-valued cost to all executions over the machine using a cost model $f : \mathbb{Z}^\omega \to \mathbb{R}$. The cost of an execution is assigned by applying the cost model f to the weight-sequence arising from transitions along the execution. In case the transition relation is non-deterministic, each execution may have multiple runs. In these cases, the cost of the execution is resolved by taking the infimum/supremum of cost of all its runs. *Quantitative inclusion* is a fundamental problem on weighted ω-automata that formalizes the goal to compare between two given weighted ω-automata.

Definition 1 (Quantitative inclusion). *Given two weighted ω-automata P and Q over the* same *cost model $f : \mathbb{Z}^\omega \to \mathbb{R}$,* quantitative inclusion over *f, or f-*inclusion *in short, denoted by $P \subseteq_f Q$, asks whether for every execution, its cost in P is less than (or equal to) its cost in Q.*

Thus, quantitative inclusion can be seen as a quantitative generalization of *language inclusion over Büchi automata* which forms the basis of model-checking over temporal properties.

The lack of generalizability is apparent from the complexity-theoretic results. While quantitative inclusion is PSPACE-hard, there is ample variance in upper bounds. Quantitative inclusion is PSPACE-complete under limsup/liminf [17] and undecidable for limit-average [21]. For discounted-sum (DS), currently even decidability is unknown. In the special case of integer discount factors, DS inclusion is EXPTIME [15,17]. For the decidable problems, even the algorithms are very varied. Limsup/Liminf-inlcusion is solved by examination of maximum weight of cycles [16], while under DS inclusion for integer discount factors combines subset-construction and linear-programming [3].

Question. *Can one develop a unifying theory for quantitative analysis that generalizes across a variety of cost models?*

Challenge 2. Separation-of-techniques refers to the inherent dichotomy in solving problems in quantitative analysis. Most algorithms comprise of two phases: A *structural phase*, which reasons about the structure of the quantitative system(s) using techniques from automata or graphs; and a *numerical phase*, which reasons about the quantitative dimension/cost model using numerical methods. The techniques used in both phases are incompatible and difficult to combine. This adversely affects the scalability and applicability of these algorithms.

A concrete instance of this appears in planning under resource constraints against an adversary. This situation can be formulated as a *two-player quantitative graph game*, or *quantitative game* in short. In these games, players take turns to pass a token along the *transition relation* between the states. As the token is pushed around, the play accumulates costs along the transitions using the underlying cost model $f : Z^\omega - \mathbb{R}$. One player attempts to maximize the cost, while the other player minimizes it.

Definition 2 (Solving quantitative games). *A quantitative game can be solved in two ways:*

Optimization problem *Generate a strategy which computes the optimal cost from all plays of the game.*

Satisficing problem *Given a threshold value $v \in \mathbb{Q}$, the problem is to determine whether the minimizing (or maximizing) player has a strategy that ensures the cost of all resulting plays is lower (or greater) than the threshold v.*

Definition 3 (Solving quantitative games with qualitative objectives). *Given a quantitative game and a qualitative objective, to solve it means to generate a policy that solves the quantitative game (via optimization or satisficing) and ensures every resulting play satisfies the qualitative objective.*

Separation-of-techniques appears in solving quantitative games with qualitative objectives since quantitative games are solved with numerical methods such as mix-max optimization, while qualitative objective solves with automata-based methods. For most cost-models, it is known that solutions satisfying a qualitative objective will not be optimal.

Question. *Can one design an* integrated approach *for quantitative analysis that combines the two phases as opposed to separation-of-techniques? Will the approach be efficient and scalable?*

2 Contributions

This dissertation contributes towards a novel theoretical framework that addresses both of the challenges, and demonstrates utility on problems of quantitative inclusion and solving quantitative games. The introduced framework, called *comparator automata* or *comparators* in short, builds on automata-theoretic foundations to generalize across a variety of cost models. The crux of comparators is that they substitute the numerical analysis phase with automata-based methods, and hence naturally offer an integrated method for quantitative analysis. In all, we show that comparator-based algorithms have the advantages of generalizable results, yield complexity-theoretic and algorithmic advances, render practically scalable solutions, and broaden the applicability of quantitative reasoning.

2.1 Comparator automata

The dissertation takes the view that many classic questions in formal methods can be seen as involving comparisons between different system runs or inputs. For instance, the classical *model checking problem* of verifying if a system S satisfies a linear-time temporal specification P [20]. Traditionally, this problem is phrased language-theoretically: S and P are interpreted as sets of (infinite) words, and S is determined to satisfy P if $S \subseteq P$. The problem, however, can also be framed in terms of a *comparison* between words in S and P. Suppose a word w is assigned a weight of 1 if it belongs to the language of the system or property, and 0 otherwise. Then determining if $S \subseteq P$ amounts to checking whether the weight of every word in S is less than or equal to its weight in P [5].

The ubiquity of comparisons is more pronounced in quantitative analysis: Firstly, because every system execution is assigned a real-valued cost. W.l.o.g, we can assume that the cost model is an *aggregate function* $f : \mathbb{Z}^\omega \to \mathbb{R}$. The cost of an execution is the value of f applied to the weight-sequence of the execution; Secondly, because problems in quantitative analysis reduce to comparing the cost of executions to a constant value (such as in quantitative games), or more generally to the cost of another execution (as in quantitative inclusion). Thus, the dissertation takes the view that the comparison of costs of system executions with the costs on other executions is the fundamental operation in quantitative reasoning, and hence that should be brought to the forefront. To this end, we introduce *comparator automata* (*comparators*, in short), a class of automata that read pairs of infinite weight sequences synchronously and compare their aggregate values in an online manner [10].

Definition 4 (Comparator automata [10]). *A* comparator automata for aggregate function f, relation R, and upper bound $\mu > 0$ *is an automaton that accepts a pair* $(A, B) \in (\Sigma \times \Sigma)^\omega$ *of sequences of bounded integers, where* $\Sigma = \{-\mu, \mu - 1, \ldots, \mu\}$*, iff* $f(A)\ \mathsf{R}\ f(B)$*, where* $\mathsf{R} \in \{>, <, \geq, \leq, \neq =\}$ *is an inequality or equality relation.*

The dissertation lays the foundations of comparator automata. It undertakes an investigation of ω-*regular comparators* which refer to comparators that are finite-state and accept by the Büchi condition:

Theorem. [10,11] 1. Comparators for limsup, liminf, discounted-sum (DS) with integer discount factors, and ω-regular functions [19] are ω-regular. When the upper bound μ is represented in unary, the size of these comparators is polynomial in μ.
2. Comparators for limit-average and DS with non-integer discount factors are not ω-regular.
3. ω-regular comparators are closed under set-theoretic operations.

The advantage of ω-regular comparators is that they yield *generalizable solutions* to a variety of problems in quantitative analysis for all aggregate functions that permit an ω-regular comparators. Furthermore, they resolve the issue of separation-of-techniques by substituting the numerical phase with automata-based operations obtained from using the comparator. Thus, with ω-regular comparators, problems in quantitative analysis are reduced to problems in qualitative analysis, as shown below:

Theorem. [10] Let the underlying aggregate function permit ω-regular comparators. Then,

1. Quantitative inclusion reduces to language inclusion in polynomial time.
   (a) Quantitative inclusion over aggregate functions with ω-regular comparators is PSPACE-complete in size of the input weighted automata and the comparator automaton.
2. Solving the satisficing problem on quantitative games with perfect and imperfect information reduces to solving (non-quantitative) graph games with perfect or imperfect information, respectively, in polynomial time.

The above results renders a recipe for generalizable algorithms for the two problems. A crucial feature is that these algorithms are also efficient, since the reductions are polynomial in size of the inputs and the comparator automata. In fact, in specific problems, we demonstrate that comparator-based algorithms result in complexity-theoretic and algorithmic advances, as detailed below.

2.2 Quantitative inclusion over discounted-sum

Quantitative inclusion over the discounted-sum aggregation function, or *DS inclusion* in short, has been shown to have applications in the analysis of rational-behaviors in multi-agent systems, where discounted-sum is used to compute agent rewards [1,7]. Yet it is not used in practice since the decidability of DS inclusion is unknown. This has been an open problem for more than 15 years now. In the special case where the discount factor is an integer, the problem is known to have an EXPTIME upper bound and a PSPACE lower bound. Hence, its exact complexity is unknown.

Comparator-based arguments make resolutions towards both of these questions:

Theorem. 1. For integer discount factors, DS inclusion is PSPACE-complete [10].
2. For non-integer discount factors,, a co-recursively-enumerable, anytime algorithm for DS inclusion (over finite-length words) can be designed with comparator-based methods [12].

The first result follows from comparator-based reduction of quantitative inclusion since comparators for discounted-sum, called DS comparators henceforth, are ω-regular for integer discount factors. This resolves the open question w.r.t integer discount factors while also offering an integrated algorithm for the same. The merit of this integrated approach is highlighted through an empirical comparison of the comparator-based solution with prior known separation-of-techniques approaches. The analysis demonstrates that despite having poorer worst-case complexity, the comparator-based algorithm outperforms the later [9]. Furthermore, since all intermediate operations are derived from automata-based reasoning, comparators present a unique opportunity to implore language-theoretic properties to boost algorithmic performance of problems in quantitative analysis. In particular, we show that DS comparators are safety or co-safety languages [13]. This is used to replace the notoriously non-performant step of Büchi complementation [27] in the comparator-based algorithm for DS inclusion with subset-construction, resulting in improvements in algorithmic performance across the board [13].

The second result makes in-roads to solve DS inclusion for non-integer discount factors for *practical purposes* as opposed to resolving its unknown decidability. To this end, we design an anytime algorithm for DS inclusion, which guarantees to either terminate with a crisp boolean answer to whether DS inclusion holds or generates a tighter approximation of DS inclusion as time elapses [12]. Over finite-sequence semantics, the algorithm is also guaranteed to terminate on all false input instances, making it co-recursively enumerable. A part of the challenge is that DS comparators are known to not be ω-regular for non-integer discount factors. To this end, we prove that for non-integer discount factors, approximations of DS permit ω-regular comparators [12]. These comparators are then used to design the said algorithm. To the best of our knowledge, this is the first practical algorithm for the problem.

The study of the DS inclusion problem with comparator-based algorithms exhibits how comparators may lead to complexity-theoretic as well as algorithmic advances to DS inclusion. The properties of comparators utilized in the exploration of DS inclusion are as follows:

Theorem. 1. For integer discount factors, DS comparators are safety or co-safety automata [13].
2. For non-integer discount factors, comparators for approximations of DS are ω-regular [12].

This encourages the evaluation of comparator-based solutions on other classes of problems in quantitative reasoning in order to obtain a well-rounded perspective on the impact of comparators.

2.3 Quantitative games over discounted-sum

Quantitative games over the discounted-sum aggregation function, *DS games* in short, finds applications in decision-making domains such as planning and reactive synthesis. Here the quantitative properties describe *soft constraints* such as quality measures [14], cost and resources [23,26], rewards [29], and the like. Due to the application domain, often times DS games are also accompanied with a *hard* qualitative constraint expressed by a temporal goal. Since the quantitative constraints are soft, it suffices to generate solutions that are *good enough* w.r.t. the quantitative property, i.e., generate a satisficing solution. Yet, the most common form of analysis of DS games is via the optimization problem. The issue is that it is proven that solutions which are both optimal and satisfy a temporal goal may not exist [18], thus limiting the applicability of DS games. In addition, we prove that solution to the optimization problem via the celebrated approach of Value-iteration (VI) is expensive: On a DS game with $|V|$ states and $|E|$ transitions, VI takes $\Theta(|V|^2)$ iterations, and solves in $\mathcal{O}(|V|^2 \cdot |E|)$ and $\mathcal{O}(|V|^4 \cdot |E|)$ under the unit-cost and bit-cost models of arithmetic, respectively. A naive algorithm for the satisficing problem goes through solving the optimization problem. However, both the limitations of optimization transcend to satisficing by this method.

A comparator-based argument solves satisficing and alleviates both of these issues:

Theorem. [8] 1. For integer discount factors, satisficing problem on DS games reduces to solving a safety or reachability game. The resulting algorithm is $\mathcal{O}(|V| + |E|)$ where V and E are the sets of states and transitions, respectively, of the DS game.
2. For integer discount factors, solving the satisficing problem over DS games with temporal objectives reduces to solving a parity game.

The reduction to safety or reachability games is a consequence of the safety/co-safety property of DS comparators. As was the case for DS inclusion, the resulting algorithm is purely automata-based. Since temporal objectives are also solved using automata methods, the solutions of comparator-based satisficing and satisfying the temporal objective can be seamlessly integrated. In this case, it reduces to solving a parity game, where the objective is, intuitively, the combined objective of quantitative and qualitative constraints. We also demonstrate the efficacy through empirical evaluations which shows that comparator-based solutions are scalable, efficient, and robust in performance.

In ongoing work, we are extending these result to non-integer discount factors. For that, we solve a notion of *approximate satisficing* which makes use of the ω-regular comparators for approximation of DS for non-integer discount factors. We are applying these techniques to planning with soft quantitative constraints and hard qualitative constraints under adversarial environments in robotics.

3 Concluding remarks and future work

This dissertation introduces and begins the investigation of comparator automata. Through the investigation of ω-regular comparators and their impact on DS inclusion and DS games, we establish that comparator automata are a promising technique to mitigate some of the challenges posed by traditional quantitative reasoning. In terms of techniques, they bridge quantitative reasoning with qualitative reasoning, and creates a channel by which the former can benefit from the advances in the later. We believe this dissertation has opened a new direction of research - the application of automata-based methods in quantitative reasoning. There are several directions for future work, few of which are described below:

Theory of comparator automata The dissertation has begun the theory of comparator automata, but several questions remain unanswered. For instance, what are the necessary and sufficient conditions for an aggregate function to permit ω-regular comparators? The study of comparators which are not ω-regular is completed untapped. Since we are aware of aggregate function which do not permit ω-regular comparators, these questions are of interest to the theory of comparators.

Applications to probabilistic domains and reinforcement learning (RL) The issue of separation-of-techniques has been identified in the formal reasoning of Markov Decision Processes (MDPs) especially when they are combined with rewards or qualitative objectives [24]. MDPs with discounted-rewards are increasing in importance due to their relevance to RL. The question here is whether one could use DS comparators for formal reasoning of RL or RL-enabled policies by facilitating the incorporation of temporal goals.

References

1. D. Abreu. On the theory of infinitely repeated games with discounting. *Econometrica*, pages 383–396, 1988.
2. S. Almagor, U. Boker, and O. Kupferman. Formalizing and reasoning about quality. In *In Proc. of ICALP*, pages 15–27. Springer, 2013.
3. D. Andersson. An improved algorithm for discounted payoff games. In *ESSLLI Student Session*, pages 91–98, 2006.
4. C. Baier. Probabilistic model checking. In *Dependable Software Systems Engineering*, pages 1–23. 2016.
5. C. Baier, J.-P. Katoen, et al. *Principles of model checking*. MIT press Cambridge, 2008.
6. S. Bansal. Automata-based quantitative analysis. Doctoral Dissertation (Submitted), Rice University, June 2020.
7. S. Bansal. Algorithmic analysis of regular repeated games. Master's thesis, Rice University, 2016.
8. S. Bansal, K. Chatterjee, and M. Y. Vardi. On satisficing in quantitative games. In *submission*.
9. S. Bansal, S. Chaudhuri, and M. Y. Vardi. Automata vs linear-programming discounted-sum inclusion. In *Proc. of CAV*, 2018.
10. S. Bansal, S. Chaudhuri, and M. Y. Vardi. Comparator automata in quantitative verification. In *Proc. of FOSSACS*, 2018.
11. S. Bansal, S. Chaudhuri, and M. Y. Vardi. Comparator automata in quantitative verification (full version). *CoRR*, abs/1812.06569, 2018.
12. S. Bansal and M. Y. Vardi. Anytime discounted-sum inclusion. In *submission*.
13. S. Bansal and M. Y. Vardi. Safety and co-safety comparator automata for discounted-sum inclusion. In *Proc. of CAV*, 2019.
14. R. Bloem, K. Chatterjee, T. A. Henzinger, and B. Jobstmann. Better quality in synthesis through quantitative objectives. In *Proc. of CAV*, pages 140–156. Springer, 2009.
15. U. Boker and T. A. Henzinger. Exact and approximate determinization of discounted-sum automata. *LMCS*, 10(1), 2014.
16. K. Chatterjee, L. Doyen, and T. A. Henzinger. Expressiveness and closure properties for quantitative languages. In *Proc. of LICS*, pages 199–208. IEEE, 2009.
17. K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. *Transactions on Computational Logic*, 11(4):23, 2010.
18. K. Chatterjee, T. A. Henzinger, J. Otop, and Y. Velner. Quantitative fair simulation games. *Information and Computation*, 254:143–166, 2017.
19. S. Chaudhuri, S. Sankaranarayanan, and M. Y. Vardi. Regular real analysis. In *Proc. of LICS*, pages 509–518, 2013.
20. E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 8(2):244–263, 1986.
21. A. Degorre, L. Doyen, R. Gentilini, J.-F. Raskin, and S. Toruńczyk. Energy and mean-payoff games with imperfect information. In *International Workshop on Computer Science Logic*, pages 260–274. Springer, 2010.
22. M. Droste, W. Kuich, and H. Vogler. *Handbook of weighted automata*. Springer, 2009.
23. K. He, M. Lahijanian, L. Kavraki, and M. Vardi. Reactive synthesis for finite tasks under resource constraints. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 5326–5332. IEEE, 2017.
24. M. Kwiatkowska. Quantitative verification: Models, techniques and tools. In *Proc. 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE)*, pages 449–458. ACM Press, September 2007.
25. M. Kwiatkowska, G. Norman, and D. Parker. Advances and challenges of probabilistic model checking. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1691–1698. IEEE, 2010.
26. M. Lahijanian, S. Almagor, D. Fried, L. Kavraki, and M. Vardi. This time the robot settles for a cost: A quantitative approach to temporal logic planning with partial satisfaction. In *AAAI*, pages 3664–3671, 2015.
27. M. Y. Vardi. The büchi complementation saga. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 12–22. Springer, 2007.
28. A. Wakankar, P. K. Pandya, and R. M. Matteplackel. Dcsynth: Guided reactive synthesis with soft requirements. In *Working Conference on Verified Software: Theories, Tools, and Experiments*, pages 124–142. Springer, 2019.
29. M. Wen, R. Ehlers, and U. Topcu. Correct-by-synthesis reinforcement learning with temporal logic constraints. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4983–4990. IEEE, 2015.