Synthesis of Controllable Nash Equilibria in Quantitative Objective Games

Giuseppe Perelli *
University of Leicester

The *synthesis problem* for LTL (linear temporal logic) gets as input a specification in LTL and outputs a reactive system that satisfies it — if such exists (Pnueli and Rosner 1989). The specification is over input signals, controlled by the environment, and output signals, controlled by the system. The system should satisfy the specification in all environments. The environment with which the system interacts is often composed of other systems. For example, the clients interacting with a server are by themselves distinct entities (which we call *agents*). In the traditional approach to synthesis, the agents can be seen as if their only objective is to conspire to fail the system. Hence the term "hostile environment" that is traditionally used in the context of synthesis. In real life, however, many times agents have objectives of their own, other than to fail the system. The approach taken in the field of algorithmic game theory (Nisan et al. 2007) is to assume that agents interacting with a computational system are *rational*; i.e., agents act to achieve their own objectives.

In (Fisman, Kupferman, and Lustig 2010), Fisman et al. introduced *rational synthesis*. The input to the rational-synthesis problem consists of LTL formulas specifying the objectives of the system and the agents that constitute the environment. The signals over which the objectives are defined are partitioned among the system and the agents, so that each of them controls a subset of the signals. There are two approaches to rational synthesis. In *cooperative* rational synthesis, the desired output is a strategy profile such that the objective of the system is satisfied in the computation that is the outcome of the profile, and the agents that constitute the environment have no incentive to deviate from the strategies suggested to them; that is, the profile is a *Nash equilibrium* (NE) (Nash 1950). Thus, in the cooperative setting, we assume that once we suggest to the agents strategies that constitute an equilibrium, they follow them. Then, in *non-cooperative* rational synthesis, studied in (Kupferman, Perelli, and Vardi 2016), the desired output is a strategy for the system such that its objective is satisfied in all NE profiles in which the system follows this strategy. Thus, in the non-cooperative setting, the agents are rational, but need not follow a suggested profile. The rational-synthesis problem for LTL in the cooperative setting is 2EXPTIME-complete (Fisman, Kupferman, and Lustig 2010), as is traditional LTL synthesis. In the non-cooperative setting, the best known complexity is 3EXPTIME (Kupferman, Perelli, and Vardi 2016).[1]

Traditional games in game theory are finite and their outcome depends on the final position of the game (Nisan and Ronen 1999; Nisan et al. 2007). In contrast, the systems we reason about maintain an *on-going interaction* with their environment (Harel and Pnueli 1985), and reasoning about their behavior refers not to their final state (in fact, we considers non-terminating systems, with no final state) but rather to the *language* of computations that they generate. While LTL specifications enable the description of rich on-going behaviors, the semantics of LTL is Boolean: a computation may satisfy a specification or it may not. As argued in (Almagor, Boker, and Kupferman 2016), the Boolean nature of LTL is a real obstacle in synthesis. Indeed, while many systems may satisfy a specification, they may do so at different levels of quality. Consequently, designers would be willing to give up manual design only after being convinced that the automatic procedure that replaces it generates systems of comparable quality. As argued in (Kupferman, Perelli, and Vardi 2016), the extension of the synthesis problem to the rational setting makes the quantitative setting even more appealing. Indeed, objectives in typical game-theory applications are quantitative, and interesting properties of games often refer to their quantitative aspects.

We study the rational-synthesis problem for a very strong quantitative formalism, namely $\mathrm{LTL}[\mathcal{F}]$. The logic $\mathrm{LTL}[\mathcal{F}]$ is a multi-valued logic that augments LTL with quality operators (Almagor, Boker, and Kupferman 2016). The satisfaction value of an $\mathrm{LTL}[\mathcal{F}]$ formula is a real value in $[0, 1]$, where the higher the value is, the higher is the quality in which the computation satisfies the specification. The quality operators in $\mathcal{F}$ can prioritize different scenarios or reduce

*This is a joint work with Shaull Almagor and Orna Kupferman that appeared in the proceeding of the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence.

[1] The complexity specified in (Kupferman, Perelli, and Vardi 2016) is actually 2EXPTIME-complete, yet the complexity analysis there misses one alternation between strategy quantifiers in the strategy-logic formula to which the problem is reduced. Taking this additional alternation into account, the complexity goes up to 3EXPTIME.

the satisfaction value of computations in which delays occur. For example, as in earlier work on multi-valued extensions of LTL (c.f., (Faella, Legay, and Stoelinga 2008)), the set $\mathcal{F}$ may contain the $\min\{x, y\}$, $\max\{x, y\}$, and $1 - x$ functions, which are the standard quantitative analogues of the $\wedge$, $\vee$, and $\neg$ operators. The novelty of $\mathrm{LTL}[\mathcal{F}]$ is the ability to manipulate values by arbitrary functions. For example, $\mathcal{F}$ may contain the weighted-average function $\oplus_\lambda$. The satisfaction value of the formula $\varphi \oplus_\lambda \psi$ is the weighted (according to λ) average between the satisfaction values of φ and ψ. This enables the specification of the quality of the system to interpolate different aspects of it. As an example, consider the $\mathrm{LTL}[\mathcal{F}]$ formula $\mathsf{G}(req \to (grant \oplus_{\frac{2}{3}} \mathsf{X} grant))$. The formula states that we want requests to be granted immediately and the grant to hold for two transactions. When this always holds, the satisfaction value is $\frac{2}{3} + \frac{1}{3} = 1$. We are quite okay with grants that are given immediately and last for only one transaction, in which case the satisfaction value is $\frac{2}{3}$, and less content when grants arrive with a delay, in which case the satisfaction value is $\frac{1}{3}$.

The extension to $\mathrm{LTL}[\mathcal{F}]$ significantly strengthens the framework of rational synthesis. In addition, we study the stability of rational synthesis and additional game- and social-choice theoretic aspects of it. We generalize the setting to an arbitrary partition of the set of agents to controllable and uncontrollable ones. In particular, the case there are no controllable agents corresponds to interactions with no authority. We refine the stability-inefficiency measures of *price of stability* (PoS) (Anshelevich et al. 2008) and *price of anarchy* (PoA) (Koutsoupias and Papadimitriou 2009; Papadimitriou 2001) to a setting where some of the agents are controllable. Essentially, these notions measure how much we lose from the absence of a central authority by comparing the utility of a social-optimum profile (that is, a profile that maximizes the profits of all agents together) with that of NE profiles. Our refinement enables a distinction between cases where the behavior of the controllable agents is fixed and cases it is not.

Studying the stability of rational synthesis, we prove that a rational-synthesis game need not have an NE, and that for some utility functions, the PoS and PoA may not be bounded. We relate the cooperative and non-cooperative settings with the two stability-inefficiency measures. In the cooperative setting, we may suggest to the agents a best NE, thus the cooperative setting corresponds to the PoS measure. On the other hand, in the non-cooperative setting, the agents may follow the worst NE, which corresponds to the PoA measure. This settles a discussion in the community about the necessity of both settings, and also implies that the profit to the controllable components in the non-cooperative setting may be unboundedly smaller than the profit in the cooperative setting.

We solve decision problems for rational synthesis with $\mathrm{LTL}[\mathcal{F}]$ objectives. Our algorithms make use of *strategy logic* and decision procedures for it (Mogavero et al. 2012; 2014). Thus, we are able to handle the richer quantitative setting using existing tools. In particular, we show that the cooperative and non-cooperative versions of $\mathrm{LTL}[\mathcal{F}]$ rational synthesis are 2EXPTIME-complete and in 3EXPTIME, respectively, and that so are the problems of calculating the various stability-inefficiency measures, and other measures that quantify the game and its outcomes. Thus, the complexity of rational synthesis in the quantitative setting is not harder than the best known complexity in the Boolean setting. Due to the lack of space, some of the proofs are omitted and can be found in the full version, in the authors' URLs.

References

Almagor, S.; Boker, U.; and Kupferman, O. 2016. Formally reasoning about quality. *J. ACM* 63(3):24:1–24:56.

Anshelevich, E.; Dasgupta, A.; Kleinberg, J.; Tardos, E.; Wexler, T.; and Roughgarden, T. 2008. The price of stability for network design with fair cost allocation. *SIAM J. Comput.* 38(4):1602–1623.

Faella, M.; Legay, A.; and Stoelinga, M. 2008. Model checking quantitative linear time logic. *Electr. Notes Theor. Comput. Sci.* 220(3):61–77.

Fisman, D.; Kupferman, O.; and Lustig, Y. 2010. Rational synthesis. In *Proc. 16th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, volume 6015 of *Lecture Notes in Computer Science*, 190–204. Springer.

Harel, D., and Pnueli, A. 1985. On the development of reactive systems. In Apt, K., ed., *Logics and Models of Concurrent Systems*, volume F-13 of *NATO Advanced Summer Institutes*. Springer. 477–498.

Koutsoupias, E., and Papadimitriou, C. 2009. Worst-case equilibria. *Computer Science Review* 3(2):65–69.

Kupferman, O.; Perelli, G.; and Vardi, M. Y. 2016. Synthesis with rational environments. *Ann. Math. Artif. Intell.* 78(1):3–20.

Mogavero, F.; Murano, A.; Perelli, G.; and Vardi, M. 2012. What Makes Atl* Decidable? A Decidable Fragment of Strategy Logic. In *CONCUR'12*, LNCS 7454, 193–208. Springer.

Mogavero, F.; Murano, A.; Perelli, G.; and Vardi, M. 2014. Reasoning About Strategies: On the Model-Checking Problem. *TOCL* 15(4):34:1–47.

Nash, J. 1950. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences of the United States of America*.

Nisan, N., and Ronen, A. 1999. Algorithmic mechanism design. In *Proc. 31st ACM Symp. on Theory of Computing*, 129–140.

Nisan, N.; Roughgarden, T.; Tardos, E.; and Vazirani, V. V. 2007. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge.

Papadimitriou, C. H. 2001. Algorithms, games, and the internet. In *Proc. 33rd ACM Symp. on Theory of Computing*, 749–753.

Pnueli, A., and Rosner, R. 1989. On the Synthesis of a Reactive Module. In *POPL'89*, 179–190. Association for Computing Machinery.