Chanakya: Computer-Aided Strategic Reasoning

Suguman Bansal
Rice University
suguman@rice.edu

Swarat Chaudhuri
Rice University
swarat@rice.edu

Abstract

This paper introduces *Chanakya*, an attempt to use tools and techniques from formal methods to automatically design and analyze game-theoretic models of strategic multiagent interactions. Here, we discuss two problems of automatic characterization of a game under a given notion of rationality. First, we show how tools from automata theory can be used to completely characterize the set of Nash equilibria of infinitely repeated games. Second, we show how to use the theory of vector addition systems in the characterization of evolutionary stability of evolutionary games.

1. Introduction

Strategic reasoning is the science of deciding how to behave in an interactive environment. Often times, and rightly so, decisions are guided by *selfish* motives, and pertain to *rational behavior*. Rational behavior has various forms. In widely used internet systems such as eBay or Google Adwords, auctioneers bid, and the one with the highest bid wins. In this example rational behavior corresponds to winning at the least possible bid. Next, consider a market in which different firms produce the same product. Each firm needs to determine a price for their product. In this case, mutually agreeing upon the same price is considered rational, as this leads to co-existence and market stability. Each of these notions of rational behavior comes with natural computational questions. For example, we may want to algorithmically analyze a system of selfish agents to understand the nature of their interactions, or to design a protocol of interaction between the agents that guarantees some high-level properties in spite of selfish behavior. Hence, strategic reasoning is a very rich and active field of research that brings social science, mathematics, and computer science under the same hood.

The mathematical study of strategic reasoning is called *Game Theory* [5] . The central question in Game Theory is to characterize rational behavior in various kinds of games. Complete characterization enables better understanding of the outcome(s) of a game with selfish agents. Hence, such characterization is of interest and importance to game theorists. Characterization of games involves rigorous mathematical analysis. Unfortunately, in the present day, this analysis is done manually. Hence, it is error-prone, and time consuming. In addition, manual rigorous analysis of complicated games in virtually intractable.

Formal methods and programming languages can be of immense service here. Games can be treated as *quantitative abstractions* of systems of selfish agents where the essence of agents' motivations is captured through quantitative utility values. We can emulate rigorous mathematical analysis on these program abstractions using techniques from formal methods. This way, we can automate their mathematical analysis. Furthermore, the program abstractions of games can be enriched depending on what level of abstraction is appropriate. Together using formal methods and programming languages, one could formally define various notions of rationality, and design algorithms for their computation.

	Cooperate (C)	Defect (D)
Cooperate (C)	(2,2)	(0,3)
Defect (D)	(3,0)	(1,1)

Table 1: Prisoner's Dilemma: Utility Matrix

It will also be interesting to see the extent to which mathematical analysis can be automated. More complex mathematical analysis will enable our games to become more complex. Hence, it will improve our understanding on the complexity of games for which decidable results can be obtained.

By naming this project *Chanakya* [4], we are paying our respect to one of the greatest economists, strategists, philosophers, and teachers of all times, Chanakya (350 BC - 275 BC).

The rest of the paper is structured in the following way. We discuss the problem of characterization of games under two notions of rationality, Nash equilibra (Section 2.1), and evolutionary stability (Section 2.2).

2. Some interesting problems

This section provides a brief discussion on some concrete directions that we have been pursuing. First, we discuss computation of all Nash equilibria in Infinite Repeated Games. Second, we discuss the study of evolutionary stability in Evolutionary Games.

2.1 All Nash Equilibria in Infinite Repeated Games

Infinite repeated games [3] are used to model real life interactions such as market dynamics, online auction protocols etc.

In our recent work, we model infinite repeated games as weighted regular games. A weighted regular game represents an n-agent game. Syntactically, a weighted regular game is a büchi automaton, in which transitions occur over the alphabet of n-tuples, called the *action profile*. Each transition is labeled by at least one n-tuple, called the *weight tuple*.

The i-th element of each action profile denotes the *action* the i-th agent takes. Similarly, the i-th element of a weight tuple is a numerical value that denotes *utility* attained by the i-th agent. Multiple weight tuples on a transition denote the possibility of multiple utilities over a single transition.

Each accepting word in this automaton denotes a *play* of the game. Every play is an n-tuple, in which the i-th element denotes the strategy of the i-th agent. Since plays are tuples of n strategies, they are also called *strategy profiles*.

We demonstrate our model through the Iterated Prisoner's Dilemma (IPD), a game obtained by infinite repetition of the classic Prisoner's Dilemma game. Recall that Prisoner's Dilemma is a one-shot interaction between two agents who can either *cooperate* (action C) or *defect* (action D). Utility of this game are shown in Table 1. In the infinitely repeated version, each player can play following a *policy*. Here we describe the version of the game where agents play under a policy called *Tit-for-Tat-with-forgiveness*.

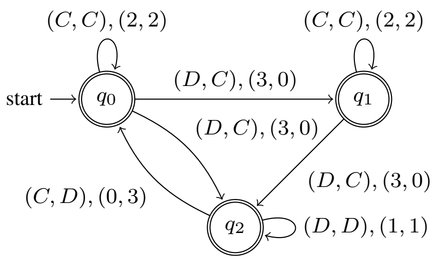

Figure 1: Agent 2 plays with Tit-for-Tat-with-forgiveness. (Start state is marked with an arrow →. Final states are double-circled).

The automaton for the game is shown in Fig 1. Under this policy, the second agent P_2 co-operates with the first agent P_1 until P_1 defects (state q_0). After P_1 defects for the first time, P_2 may chose to defect immediately in the next round, or *forgive* and defect only if P_1 defects again. In the former case, the game shifts to state q_1, else to state q_2.

We compute the utility of an agent along an accepting run by computing the *discounted sum* of utilities attained by the agent from transitions along that run. The discounted sum of sequence A with discount factor d is given by $DS(A, d) = \Sigma_{i=0}^{\infty} a_i/d^i$.

A *Nash equilibrium* of a game is a strategy profile such that if any agent unilaterally changes its strategy, then it will receive lower aggregate utility. Computation of *a* Nash equilibrium has been extensively studied in various kinds of games. Complete characterization under this notion of rationality involves computation of *all* Nash equilibria in a game. To the best of our knowledge, the problem of computing all Nash equilibria in purely quantitative games such as the ones described above has not been studied before. Note that in a weighted regular game a play may have more that one accepting run, hence a play may have multiple possible utilities for its agents. In this case, we define a strategy profile to be in Nash equilibrium if it has at least one accepting run that is in Nash equilibrium.

We are working on designing an algorithm that takes a weighted regular game with rational rewards, and a discount factor d for $d \in \mathbb{N}$ as input. The output is a büchi automaton that accepts words only if they are in Nash equilibria in the input game. What is interesting in this algorithm is that at no point do we compute the utility of any run in the game. Instead, we construct a comparator automaton that accepts a pair of bounded rational number sequences (A, B) iff $DS(A, d) > DS(B, d)$ when $d \in \mathbb{N}$. For the construction of this automaton, we use insights gained from arithmetic over numbers in base d, and the fact that for bounded number sequences, the discounted sum will also be bounded.

Preliminary Results An early prototype of this algorithm has shown promise. We have been able to re-discover known results in classical repeated games such as IPD and Repeated Auctions. More significantly, we have also been able to compute all Nash equilibria in much more complex games. One example of a complex game is that of a model of the Bitcoin Protocol. This result is extremely encouraging as manual computation of all Nash equilibria in the Bitcoin protocol is extremely tedious.

2.2 Parametric Games

Parametric games form a class of games which are parameterized on the number of agents i.e. games that are defined over an arbitrary number of agents. Evolutionary games [2] are examples of parametric games. In evolutionary games, agents do not explicitly make decisions. We illustrate this point through an example. Consider an initial population of small beetles and large beetles. Here each beetle chooses to be small or large. However this is a genetic behavior, hence beetles can not make this decision explicitly. In this population, beetles interact with each other by competing over food. The outcome of such a game measured after a period of time. It has been noticed that over a period of time, the population of small beetles gets wiped out, while the big beetles continue to thrive. In this case, being large is said to be *evolutionary stable*.

Questions regarding characterization of evolutionary games pertain to evolutionary stable behaviors. One such example is, given an initial configuration, what is the set of all possible outcomes? In other words, which behaviors are evolutionary stable from a given initial configuration. Next, one could ask if the outcome is dependent on the initial configuration. In other words, does a behavior exits that is evolutionary stable from every initial configuration. As earlier, tools and techniques from formal methods can be employed to answer these questions automatically.

However, we cannot model parametric games using weighted regular games, as we will have to construct one weighted regular game for every value of n. In addition, there is a marked difference between the notion of rationality in evolutionary games and in repeated games. Evolutionary games begin at some initial configuration of observable behaviors, such as small beetles and large beetles. Over time these configurations change. Eventually, only the evolutionary stable behaviors are observed, such as large beetle size. The game-theoretic way of reasoning about which behaviors in such scenarios, is by associating a quantitative measure of *fitness* to each behavior in a configuration. Fitness changes with each interaction. Behaviors with larger fitness value after a period of time, are said to be evolutionary stable.

Currently we are working on defining a model for parametric games that captures the notion of fitness. We propose the use of a weighted version of *Population Protocols* [1]. A population protocol can be thought of as a vector addition system (or alternatively, a Petri net) defined over non-negative integers with transitions between vectors. The sum of elements in any vector is preserved under all transitions. This work is currently in its nascent stage.

Parametric games are not only restricted to evolutionary games. We can also talk about a parameterized version of repeated games. For example, is it true that for all values of n, in an n-agent auction, auctioneers will bid truthfully?

3. Concluding Remarks

We conclude with reiterating that the marriage of Formal Methods with Game Theory opens a plethora of new and interesting problems. These problems are of theoretical and practical interest. This is a novel combination, and hence there are plenty opportunities for making significant progress in the area. This project is still in its infancy. It has the flexibility to grow in any direction. It will be interesting to see how this shapes up in the future.

Acknowledgement We thank our collaborators Krishnendu Chatterjee, Pierre Ganty, Rupak Majumdar, and Kuldeep S. Meel.

References

[1] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed computing*, 18(4):235–253, 2006.

[2] D. Easley and J. Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world.* Cambridge University Press, 2010.

[3] G. J. Mailath and L. Samuelson. *Repeated games and reputations*, volume 2. Oxford university press Oxford, 2006.

[4] CHANAKYA. Chanakya. `https://en.wikipedia.org/wiki/Chanakya`.

[5] J. Von Neumann and O. Morgenstern. *Theory of games and economic behavior.* Princeton university press, 2007.