FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Jakub Černý

Computational Bounded Rationality

Department of Applied Mathematics

Supervisor of the master thesis:	prof. RNDr. Martin Loebl, CSc.
Consultant of the master thesis:	Mgr. Branislav Bošanský, Ph.D.
Study programme:	Computer Science
Study branch:	Discrete models and algorithms

Prague 2017

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In Prague, July 21 2017 ....

Název práce: Výpočetní omezená racionalita

Autor: Jakub Černý

Katedra: Katedra aplikované matematiky

Vedoucí diplomové práce: prof. RNDr. Martin Loebl, CSc., Katedra aplikované matematiky

Konzultant diplomové práce: Mgr. Branislav Bošanský, Ph.D., Katedra počítačů, Fakulta elektrotechnická, České Vysoké Učení Technické v Praze

Abstrakt: Tato závěrečná práce formalizuje model omezené racionality hráčů v sekvenčních hrách nazvaný herní schémata. Ve zkoumaném modelu jsou strategie reprezentované strukturou skládající se z konečného automatu a dvou výpočetních funkcí. Zatímco konečný automat reprezentuje hráčovu strukturovanou paměť, výpočetní funkce reprezentují jeho schopnost efektivně abstrahovat danou hru. Schémata jsou realizacemi čistých strategií a mohou být hráčem implementovány za účelem hraní sekvenční hry. Práce ukazuje jak zkonstruovat korektně hrající schéma pro jakoukoli strategii v jakékoli sekvenční hře s vícero hráči a jak určit jeho složitost. Dokazuje, že ekvilibrium vždy existuje a jeho výpočet je PPAD-těžký. Navíc práce definuje třídu efektivně reprezentovatelných strategií, pomocí které lze spočítat MAXPAY-EFCE v polynomiálním čase.

Klíčová slova: Konečné automaty, omezená racionalita, extenzivní hry, výpočetní složitost, algoritmy.

Title: Computational Bounded Rationality

Author: Jakub Černý

Department: Department of Applied Mathematics

Supervisor: prof. RNDr. Martin Loebl, CSc., Department of Applied Mathematics

Consultant: Mgr. Branislav Bošanský, Ph.D., Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague

Abstract: This thesis formalizes a model of bounded rationality in extensive-form games called game-playing schemata. In this model, the strategies are represented by a structure consisting of a deterministic finite automaton and two computational functions. The automaton represents a structured memory of the player, while the functions represent the ability of the player to identify efficient abstractions of the game. Together, the schema is a realization of a pure strategy which can be implemented by a player in order to play a given game. The thesis shows how to construct correctly playing schema for every pure strategy in any multi-player extensive-form game with perfect recall and how to evaluate its complexity. It proves that equilibria in schemata strategies always exist and computing them is PPAD-hard. Moreover, for a class of efficiently representable strategies, computing MAXPAY-EFCE can be done in polynomial time.

Keywords: Deterministic finite automata, bounded rationality, extensive-form games, computational complexity, algorithms.

I would like to thank my supervisor, prof. RNDr. Martin Loebl, CSc., for giving me the opportunity to work on this project, for his patient guidance, much appreciated encouragement and frequent helpful critiques, as he guided me through the process of formalizing my ideas and writing this thesis. I would also like to express my gratitude towards my consultant, Mgr. Branislav Bošanský, PhD., for long discussions we had on the consistency of the model and the consequences on computational aspects of different equilibria.

I would especially wish to thank my family for the love, support, and continual encouragement I have gotten throughout my study. In particular, I would like to thank my parents and my sister. I would not be able to stay focused without the constant reminders of Naďa – the reality is always one step ahead of the model.

I also wish to acknowledge the help provided by Hayk Divotyan, Daniel Meister and Raman Samusevich, my friends whom I asked for opinions on individual chapters and they contributed by many valuable insights.

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not with standing any copyright notation here on.

Contents

Introduction

Games and puzzles have been an integral part of human society for hundreds of years. For some reason, the art of conflict fascinates mankind regardless of race, religion or culture. In fact, every age had its own masters of strategic reasoning, might they be the invincible commanders of bloody wars; or cold-blooded decision makers of economic battlefields. But at the beginning, the knowledge about a performance of different strategies could have been gained merely empirically. The only way to beat the game was through excessive study and infinitely repeated game-plays.

Finding ways to better justify how to win led to the mathematical formalism of *noncooperative game theory*. This theory tries to formally define games and identify rational behavior which maximizes the probability of victory. Based on the disparity of information provided to the participating players, game theory establishes different solution concepts known as equilibria. The underlying concept of the game theory is *Nash Equilibrium (NE)* [1], which implies that every game contains a tuple strategies such that no player profits from altering his strategy. The Nash equilibrium was several times refined in order to overcome the perceived flaws, but all these concepts are closely related. Later, more general solution concept called *Correlated Equilibrium (CE)* [2] introduced the external events which can help the players coordinate.

With the computational power provided by computers, the theory gave rise to algorithms able to find strategies even for large games – a process which would be impossible to do manually. This breakthrough encouraged the effort to formally describe many processes in the human society as games. The main applications remain in economics, however, any multi-agent system with limited resources, where the agents are forced to interact, can be modeled as a game. These models proved to be well-suited for many real-world strategic situations, with both descriptive [3] and prescriptive [4] applications.

However, these early game-theoretical models are predominantly based on strong assumptions about the nature of the environment in which they are played. For example, they assume that all players are fully aware of the whole structure of the game and that their interests are easily recognizable. In contrast, experiments performed with human participants have shown several non-negligible deviations from predictions made by traditional game theory. These findings implied that most human beings are not always rational and that their typical choices do not always represent the utility maximizing option.

In an attempt to mathematically explain these findings, a theory of *bounded-rational agents* was developed as a generalization of traditional game theory [5]. Inspired by many conducted experiments, the behavioral economists argued that it is precisely the bounded rationality, which forms many interactions in the modern world. From their point of view [6], the bounded rationality and the complexity of environment are both inextricably linked. First, the emergence of complex social structures would not be possible without interactions between bounded-rational agents. Second, the bounded-rational nature of agents is in itself a consequence of a complex environment. It arises, as the biologically bounded agents interact in situations, which they are not fully able to comprehend. As a result, in one of the most well-known publications about bounded rationality [7], Simon proposed the following taxonomy of emergence of bounded rationality:

1. limited knowledge of the world;
2. limited ability to evoke this knowledge;
3. limited ability to work out consequences of actions;
4. limited ability to conjure up possible courses of action;
5. limited ability to cope with uncertainty; and
6. limited ability to adjudicate among competing wants.

In the recent years, several solution concepts of computational game theory [8, 9] were refined to model some of these limitations proposed by Simon; in order to provide more robust strategies against human adversaries. Incorporating bounded rationality proved to be a way to make the models more accurate and hence more suited for situations where the true intentions of the players might not be well known or when the opponents systematically make mistakes.

This thesis analyzes bounded rationality from the perspective of computational complexity and algorithmic game theory. An effort has been devoted to deriving a mathematical model with suitable properties for computing existing solution concepts, rather than to exactly model human behavior as in cognitive modeling. The work proposes a model of automata playing extensive-form games called *game-playing schemata*, which (in Simon's classification) captures agent's limited knowledge of the world, the ability to evoke it (because the size of the automaton might be bounded) and the limited ability to work out the consequences of actions (as the automata make a compression of the game's strategies). It does so in order to model the trade-off every bounded-rational decision maker faces when choosing a strategy. On one hand, he demands the strategy allows him to achieve his goals; on the other hand, he requires it to be as simple as possible. There are many reasons why a player should focus on simplifying his decision making: the more complex strategies are more likely to break down, they are more difficult to learn, and may require more computational resources to be implemented. As in [10], this thesis does not examine these reasons any further, it just assumes that the complex reasoning is costly, but a complexity of any strategy can be evaluated by the player.

Moreover, it has been shown that people tend to compress the obtained knowledge to reason more efficiently using simple patterns and structures [11]. By using automata, the thesis models such ability in the context of strategic decision making in environments with multiple acting agents. Such representation implements the ability of agents to intentionally forget the information they assume to be unnecessary.

Related work

The model of game-playing schemata generalizes the model of automata in extensive-form games by introducing additional functions evaluating the structure of a game tree. Even though this is a new model which has not yet been analyzed, it is closely related to several other works, especially to the original model of automata playing repeated normal-form games. From the perspective of computational game theory, the literature on bounded rationality can be divided into three main categories: the cognitive models of learning and decision making associated with behavioral game theory, the structural models of limited computational abilities, and the approaches increasing the robustness of traditional solution concepts.

Behavioral game theory generalizes the traditional game theory by introducing cognitive limitations of individual players. The field covers the areas of the behavioral economics, experimental game theory, experimental psychology, cognitive science and also the traditional game theory. The concept of bounded rationality is commonly associated with the works of Simon [12], which are seen as an introductory literature into the subject. Psychological and economic points of views can be found in a book by Selten and Gigerenzer [13] while Camerer's book [14] reviews the most influential experiments conducted so far. The most successful descriptive models used in the empirical investigation of human modeling are Quantal response (QR) model, Instance-based learning (IBL) model and model of Prospect theory (PT).

The quantal response model [15, 16] introduces uncertainty into the decision making of a player by using a softmax function to determine a probability of choosing an action. Instead of maximizing the expected utility, QR assumes that a probability of choosing an action is positively related to the utility of that action. In other words, the player chooses an action that gives a high expected utility with a probability higher than another action which gives a lower expected utility. On the other hand, the instance-based learning model [17, 18, 19] assumes that as players interact with a dynamic task, they recognize a situation according to its similarity to instances observed in the past. They adapt their perception of strategies from heuristic-based to instance-based and update the already obtained knowledge according to feedback given as a result of their actions. IBL model a decay in memorizing of instances observed long in the past and is hence in this sense similar to the model of automata playing repeated games. Finally, the model of prospect theory [20] suggests there are two functions affecting the decision-making process: the probability weighting function and the value function. The probability weighing function models human interpretation of probability and suggests that people weigh probability non-uniformly. Otherwise speaking, PT assumes that people tend to overweight low probabilities and underweight high probabilities.

The structural models of limited computational abilities assume that humans do not implement the strategies as simple mappings from situations to actions, but rather as more complex structures which better describe their cognitive abilities, e.g. deterministic finite automata (DFA) or Turing machines (TM). This line of research was originally suggested by Aumann as he summarized his ideas in [21]. Megiddo [22] questions the structural basics of bounded rationality and proposes more precise definitions, while Rubinstein [10] wrote a great overview from the perspective of game-theorist, including a lot of formal models. The works on strategies implemented by automata focus predominantly on effects of playing repeated normal-form games with automata of bounded size [23, 24, 25] and the complexity of computing a best-response automaton [26]. In the context of extensive-form games, Ramanujam [27] proposed an approach based on automata described in modal and temporal logic. More general models of Turing machines playing extensive-form games can be found in [28, 29], which approximate a finite game using an infinite sequence of converging games.

Recently, bounded rationality was applied to security games in order to *increase the robustness of traditional solution concepts*. In [30], the authors assume the players do not always choose a single optimal strategy, but rather one of the ϵ-best responses. The solution is a strategy optimal against all ϵ-deviations. In the following works [31, 8], the concept of ϵ-deviations was replaced with more precise models of players' behavior based on QR or PT. However, these computed strategies provide no quality guarantees when the estimated models of behavior are inaccurate. The monotonic maximin [32] was proposed as a more conservative solution, but providing theoretical guarantees of optimality against any regular quantal response function. According to performed experiments, all these extensions of traditional solution concepts provably increase the robustness of computed strategies.

Structural motivation

The main motivation for introducing a model of automata playing games into extensive-form games is structural. The model of automata is a well-known model with many known properties, which can be used to analyze strategies represented as automata. The goal of this thesis is to answer the following natural questions.

- **How to define automata playing extensive-form games and how to compute an optimal automaton for a given pure strategy?**

This thesis shows that for every pure strategy it is possible to construct an extended automaton (a schema) which prescribes this strategy. Moreover, in case the pure strategy has an inner structure, the schema can use it to make the representation more compact. For evaluating the compactness, the thesis introduces measures of complexity and an algorithm for minimizing a size of a schema.

- **How to define equilibria in schemata strategies? Do they always exist?**

The thesis defines two types of equilibria in games with schemata strategies. First, an equivalent of traditional Nash equilibrium in which players choose a schema strategy; and second, an equilibrium which balances an expected utility and complexity of a schema strategy. It proves that every game contains these equilibria.

- **Which strategies are efficiently representable? Which classes of games have efficiently representable strategies and how to compute an equilibrium with these strategies?**

The thesis analyzes the strategies which are representable by schemata of polylogarithmic size and low computational complexity. It presents a sufficient condition for games to have efficiently representable (also called small) strategies and shows that computing an originally NP-hard correlated equilibrium can be done in polynomial time with small schemata.

Approach of this thesis

This thesis aims at building a theory of schemata playing finite extensive-form games. For this purpose, the thesis is organized in the following structure:

- The game theory provides the necessary theoretic background for playing games. The basics of its mathematical formalism are formulated in Chapter 1. The chapter also introduces two main solution concepts relevant to this thesis; algorithms for finding the solution and the theoretical analysis of their computational complexity.
- The thesis builds the theory of schemata playing extensive games in Chapter 2. The chapter contains all necessary definitions which ensure the games are being played correctly.
- The analysis of existence and computational complexity of solution concepts with schemata strategies is presented in Chapter 3. Moreover, it defines the concept of a small schema, as an efficiently representable pure strategy. The chapter proves the equilibrium in supersmall schemata can be calculated in polynomial time and gives examples of classes of games with efficiently representable strategies.
- The last chapter concludes this thesis. It discusses the achieved theoretical results and proposes the possible directions for future research.

Chapter 1

Introduction to Game Theory

This chapter introduces the fundamentals of game theory. At the beginning, it formalizes the central notion, utility, and proceeds with the definitions of basic game descriptions. Then it describes the algorithms for computing equilibria in finite imperfect-information games with perfect recall and finite utilities, which is assumed to be a *canonical class* of games. Anytime throughout this thesis a *game* is mentioned in the text, it refers to this class, unless expressly stated otherwise. However, the individual subclasses may differ in the number of players and the existence of chance nodes. The structure of this chapter and the technical background are inspired by [33]. All definitions in this chapter are well-established notions taken from the respective cited literature and used consistently throughout most articles concerning game theory.

Game theory was originally proposed as a mathematical study of winning strategies in games, but was further developed into a theory describing interaction among independent and self-interested agents. In contrast to the cooperative theory, the basic unit of the non-cooperative game theory is an individual, not a group. The interest of each agent is quantified using *utility functions*. The utility functions are a central concept of utility theory, which studies the measures of preferences over a set of possible outcomes of a given interaction. In game theory, this interaction is the very game.

Every utility function is a mapping from states of the game to real numbers and represents the satisfaction of each player of being located in this state. The utility functions can be regarded as ordinal, when only the preference relation is meaningful; or cardinal, when the increments to satisfaction can be compared across different states. The rational models of game theory assume the utility functions to be cardinal.

Moreover, the correlation of utility functions of different players is essential. In so-called *zero-sum* (or more generally, constant-sum) games the sum over utilities of the players in each terminal state of the game is always equal to zero (or any given constant). These games are usually easier to analyze than *general-sum* games, where the property of a constant sum is violated in at least one terminal state.

1.1 Game structures

The games studied in game theory are well-defined mathematical objects. To be fully defined, a game must specify the following four essential elements: the players of the game, the information and actions available to each player at each decision point, and the payoffs for each outcome [34]. Most noncooperative games are defined in the extensive or the normal forms.

1.1.1 Normal form

The normal form is a basic type of game representation in single step games. Each player moves only once and actions are chosen simultaneously. This makes the model simpler than other forms at a cost of neglecting sequential decision making.

Definition 1 (Normal-form game [33]). *A normal-form game (NFG) is a tuple* $G = (N, A, u)$*, where*

- N *is a set of* n *players, indexed by* i*;*
- $A = A_1 \times \cdots \times A_n$*, where* A_i *is a set of actions for player* i*; and*
- $u = (u_1, \ldots, u_n)$ *where* $u_i : A \to \mathbb{R}$ *is a real-valued utility function for player* i*.*

The utility function in a normal-form game is usually visualized as a payoff matrix. The number of dimensions of this matrix is equal to the number of players participating in the game. The elements of the payoff matrix are the tuples of utility values, indexed by the respective actions available to each player.

Strategies can be seen as plans contingency or policy for playing the game. In every situation, player's reaction is defined by his strategy. One option is to choose a *pure strategy* $\pi_i \in \Pi_i$, which assigns exactly one action from A_i to player i. On the other hand, a *mixed strategy* $\delta_i \in \Delta_i$ is a probability distribution over Π_i. From the player's perspective, randomizing the decisions can be seen as a belief that he can profit from playing such action. A strategy profile is a tuple of pure strategies $\pi = (\pi_1, ..., \pi_n) \in \Pi$ or mixed strategies $\delta = (\delta_1, ..., \delta_n) \in \Delta$, which completely defines how the game will progress.

		Player 2	
		Cooperate	Defect
Player 1	Cooperate	-1,-1	-4,0
	Defect	0,-4	-3,-3

Figure 1.1: An example of normal-form game [33]. Each row of the depicted matrix is labeled by a strategy of the first player, while every column is labeled by a strategy of the second player. The tuples in the matrix denotes the utilities of the first player and the second player, respectively.

Example 1. Consider a two-player game in Figure 1.1. The depicted payoff matrix describes the *prisoner's dilemma (PD)*, a standard game modeling the situation when two members of a criminal gang are arrested and kept in isolated confinements. Both are given an opportunity to betray the other prisoner in exchange for a lesser charge. However, none of them is aware of the choice of his colleague. If they both decide to remain silent and *Cooperate*, each of them serves only a year in prison. On the other hand, if they *Defect*, the charge is 3 years. The combined choices lead to a situation when the traitor is freed and the betrayed prisoner serves 4 years.

1.1.2 Extensive form

In contrast to NFGs, extensive-form games (EFGs) represent sequential interactions between the players. The structure of an EFG can be visually represented as a tree in a sense of graph theory, with each node representing a different state of the game. Every game-state is uniquely determined by a sequence of moves executed by all players during the gameplay. In every node of a game tree, exactly one player acts. An edge from a node corresponds to an action that can be performed by the player who acts in this node. All actions are deterministic, so that they are always correctly executed. First is discussed a special class of *perfect-information games* and later the thesis moves to the more general case of *imperfect-information games*.

Perfect-information extensive games

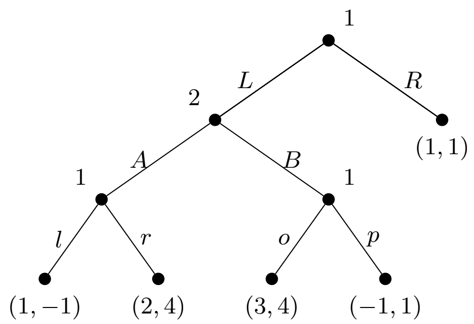

Figure 1.2: An example of EFG with perfect information, inspired by [33]. Each internal node is labeled by a player who acts in this node, while under every terminal node is a tuple of utilities obtained by the first player and the second player, respectively. Every edge is labeled by an action performed on a way from the node above to the node below. The direction of the edges is omitted, but the tree is assumed to be traversed from top to bottom.

Definition 2 (Perfect-information extensive-form game [33]). *A perfect-information EFG is a tuple $G = (N, H, Z, A, \chi, \rho, \sigma\ u)$, where*

- *N is a set of n players;*
- *H is a set of nodes;*
- *$Z \subseteq H$ is a set of terminal nodes;*
- *A is a (single) set of all actions;*
- *χ: $H \setminus Z \to 2^A$ is the action function, which assigns to each node a set of possible actions;*
- *ρ: $H \setminus Z \to N$ is the player function, which assigns to each nonterminal node a player i who acts in that node;*
- *σ: $H \setminus Z \times A \to H$ is the successor function, which maps a node and an action to a new node, and graph induced by σ is a rooted tree; and*
- *$u = (u_1, \ldots, u_n)$, where $u_i : Z \to \mathbb{R}$ is a real-valued utility function for player i on the terminal nodes Z.*

Note that the definition enables the same action to be played in different nodes, which will be useful later in the thesis. Because the nodes form a tree, it holds that whenever $\sigma(h_1, a_1) = \sigma(h_2, a_2)$, then $h_1 = h_2$ and $a_1 = a_2$. Consequently, every node can be identified with its history, which is a unique sequence of actions leading from root to that node.

Similarly to the normal form, player i can play *pure strategies*, which assign an action to every node in which the player acts. The set of all pure strategies is hence a Cartesian product $\Pi_i = \Pi_{h:\rho(h)=i}\chi(h)$ and every pure strategy is an element $\pi_i \in \Pi_i$. This set can be reduced by using only relevant pure strategies in every node. For example in Figure 1.2, the choices of actions in the lower nodes of player 1 are irrelevant, when assuming the player decided to take action R in the root. This subset of pure strategies is called *reduced pure strategies* and is denoted Π_i^*.

In [35], the author justifies the fundamental assumption of game theory: There is no loss of generality in assuming that every player chooses his strategy before the game starts, since a strategy specifies which action to take in every possible situation he might find himself in during the game. Each player therefore chooses his strategy without being informed of the strategies the other players follow. The notion of time or sequence of actions is hence unnecessary and for every perfect-information extensive-form game there exist an equivalent normal-form game. However, representing an EFG using the normal form by enumerating pure strategies is inefficient, because the number of strategies in such equivalent normal-form game is exponential in the size of the game tree.

Player 2 \ Player 1	L, l, o	L, l, p	L, r, o	L, r, p	$R, *, *$
A	1,-1	1,-1	2,4	2,4	1,1
B	3,4	-1,1	3,4	-1,1	1,1

Figure 1.3: An equivalent normal-form representation of the extensive-form game 1.2. The figure follows a standard denotation of a normal-form game.

Example 2. Consider a game depicted in Figure 1.2, in which players take moves according to the game tree. In every node of the tree, the players recognize the state of the game and are hence able to choose a legal action. In order for the first player to define a complete strategy in this game, he has to choose an action in every node which belongs to him. The enumerated reduced pure strategies for him are

$$\Pi_1^* = \{(L, l, o), (L, l, p), (L, r, o), (L, r, p), (R, *, *)\},$$

where the star symbol denotes irrelevant choices. On the other hand, the second player has only one decision node, his strategies are therefore

$$\Pi_2 = \{(A), (B)\}$$

and do not have to be further reduced. The normal-form representation of this game can be found in Figure 1.3.

There is no need to define mixed strategies for perfect-information extensive games. In NFGs, mixed strategies provided a way how to deal with uncertainty, which arises as players move simultaneously. However, in extensive games with perfect information, the players take their actions one by one and are always aware which situation they face. Another reason is that a probabilistic behavior in extensive-form games introduces a new unintuitive subtlety, which will be discussed in the next section.

Imperfect-information extensive games

Up to now, only games in which a player specifies his strategy in every state of the game were discussed. This means that he has to be able to recognize in which state he is during the whole gameplay. For this reason, these games were called *perfect-information* *games.* However, in games like poker, the player does not know in which state the game currently is, because he is not able to say which cards do his opponents have. *Imperfect-information games* model limited observations of the players by grouping certain nodes into information sets, such that a player cannot distinguish between nodes that belong to the same information set. Perfect-information games can be seen as trivial imperfect-information games in which each information set contains exactly one node. The model of imperfect-information EFGs may also represent uncertainty about the environment and stochastic events by introducing random moves of a special *Nature player.*

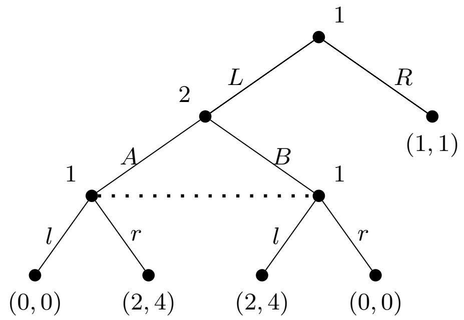

Figure 1.4: An example of EFG with imperfect information, inspired by [33]. The nodes which belong to the same information set are connected by a dashed line. Otherwise, the figure follows a standard denotation of an extensive-form game.

Definition 3 (Imperfect-information extensive-form game [33])**.** *An imperfect-information EFG is a tuple $G = (N, H, Z, A, \chi, \rho, \sigma, u, C, I)$, where*

- *$(N, H, Z, A, \chi, \rho, \sigma, u)$ is a perfect-information game;*
- *$C : A \to [0, 1]$ is a probability function for performing a random action; and*
- *$I = (I_1, \ldots, I_n)$, where $I_i = (I_{i,1}, \ldots, I_{i,k_i})$ is a set of equivalence classes on (i.e., a partition of) $\{h \in H : \rho(h) = i\}$ with the property that $\chi(h) = \chi(h')$ and $\rho(h) = \rho(h')$ whenever there exists a j for which $h \in I_{i,j}$ and $h' \in I_{i,j}$.*

One of the players might be the Nature player. The Nature player chooses each action based on a fixed probability function. This function C is assumed to be known to all players in advance. Consequently, the probability of reaching node h due to Nature (i.e., assuming that all players play all actions required to reach node h) is defined to be the product of the probabilities of all actions taken by the Nature player in the history of h. The function C can be overloaded to denote this product as $C(h)$.

Note that in order for the nodes to be truly indistinguishable, it is required that the set of actions at each choice node in an information set is the same (otherwise, the player would be able to distinguish the nodes). Thus, if $I_{i,j} \in I_i$ is an equivalence class, it is possible to unambiguously use the notation $\chi(I_{i,j})$ to denote the set of actions available to player i at any node in information set $I_{i,j}$.

Example 3. Consider the imperfect-information extensive game depicted in Figure 1.4. In this game, the information sets are denoted by a dotted line. Player 1 has two information sets: the set including the top decision node and the set including both bottom nodes. Note that the two bottom nodes belonging to the second information set have the same set of possible actions. Player 1 can be regarded as not knowing whether player 2 chose A or B when he made his choice between l and r.

Since the set of possible actions is identical for all nodes in a single information set, the set of pure strategies of player i is a Cartesian product $\Pi_i = \Pi_{I_{i,j} \in I_i} \chi(I_{i,j})$.

Similarly to the perfect-information extensive games, an equivalent normal-form game for an imperfect-information extensive game can be obtained by enumerating all pure strategies.

In case the player does not exactly know in which state he finds himself in, the emergence of probabilistic behavior is intuitive and can be seen as a way to cope with the uncertainty. There exist two known approaches into introducing randomness to choosing strategies. First, a player can randomize his strategies the same way as in the normal-form games: by choosing a probability distribution over his pure strategies and picking one pure strategy according to this distribution before the game starts. However, he can also make use of the sequential property of extensive games and rather randomize independently over actions in each information set with a preset probability distribution. This kind of strategies is called *behavioral strategies*.

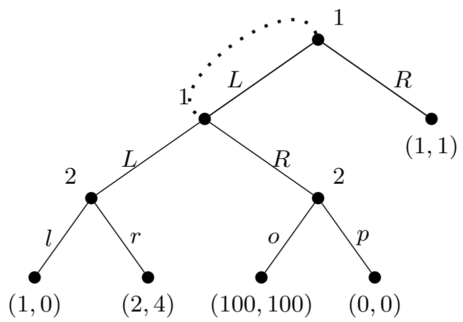

Figure 1.5: An example of EFG with imperfect recall, inspired by [33]. The figure follows a standard denotation of an extensive-form game.

In general, the expressive power of behavioral strategies and the expressive power of mixed strategies are noncomparable; in some games there are outcomes that are achieved via mixed strategies but not any behavioral strategies, and in some games it is the other way around.

Example 4. In the game shown in Figure 1.5, a player using mixed strategies can never obtain a payoff 100, since getting it would have required selecting once L and once R in the same information set, which is impossible when picking pure strategies. However, with behavioral strategies a player acts according to a probability distribution anytime he visits an information set. Every time a player chooses both L and R with non-trivial probability, he might end up in the terminal node with utility 100.

As shown by the example, the essential property of games in which behavioral and mixed strategies differ is the inability of a player to memorize his own actions. Only then he can go through the same information set during one play twice. Such cognitive ability of a player is called *recall*.

Definition 4 (Perfect recall [33])**.** *Player i has perfect recall in an imperfect-information extensive game G if for any two nodes h, h' that are in the same information set for player i, for any path $h_0, a_0, h_1, a_1, h_2, \dots, h_m, a_m, h$ from the root of the game to h (where the h_j are decision nodes and the a_j are actions) and for any path $h_0, a'_0, h'_1, a'_1, h'_2, \dots, h'_{m'}, a'_{m'}, h'$ from the root to h' it must be the case that:*

1. *$m = m'$;*
2. *for all $0 \le j \le m$, if $\rho(h_j) = i$ (i.e., h_j is a decision node of player i), then h_j and h'_j are in the same information set for i; and*
3. *for all $0 \le j \le m$, if $\rho(h_j) = i$ (i.e., h_j is a decision node of player i), then $a_j = a'_j$.*

G is a game of perfect recall if every player has perfect recall in it.

The following theorem justifies the intuition.

Theorem 1 (Equivalence of mixed and behavioral strategies [36])**.** *In a game of perfect recall, any mixed strategy of a given player can be replaced by an equivalent behavioral strategy, and any behavioral strategy can be replaced by an equivalent mixed strategy. Here two strategies are equivalent in the sense that they induce the same probabilities on outcomes, for any fixed strategy profile (mixed or behavioral) of the remaining players.*

Sequence form

The key idea for introducing sequence form is that while the number of pure strategies in an extensive-form game is exponential, there are only linearly many information sets in the game tree. If there is a way to build a strategy employing the paths in the tree, rather than pure strategies, it would avoid the exponential blowup. This can be done by using behavioral strategies and operating directly on the structure of extensive-form game; which is exactly what an equivalent representation called the sequence form does. First, the section defines a sequence and introduces several technical functions on sequences which will help to build the strategies based on them. Then the utility function is extended to work with sequences and used to describe how to create consistent strategies. Finally, the whole sequence form of an extensive game is presented.

Player 2 \ Player 1	$\emptyset$	L	R	Ll	Lr
$\emptyset$	0,0	0,0	1,1	0,0	0,0
A	0,0	0,0	0,0	0,0	2,4
B	0,0	0,0	0,0	2,4	0,0

Figure 1.6: An equivalent sequence-form representation of the extensive-form game 1.4. Each column of the depicted matrix is labeled by a sequence of the first player, while every row is labeled by a sequence of the second player. The tuples in the matrix denote the extended utilities of the first player and the second player, respectively.

Formally, a sequence is an ordered list of all actions of player i that lie on the path from the root state r to any state $h \in H$. The sequence can be also empty and such sequence is denoted $\emptyset$. The set of sequences of player i is denoted Σ_i and the set of all sets of sequences is $\Sigma = (\Sigma_1 \times \cdots \times \Sigma_n)$.

The sequence of player i leading to an information set I or a node h is denoted $seq_i(I)$ and $seq_i(h)$, respectively. The function $inf_i(\sigma_i)$ then denotes the information set in which the last action of the sequence σ_i of player i is taken. Note that for an empty sequence, function $inf_i(\emptyset)$ returns the information set of the root node for every player. Sequences can be extended by finding feasible actions in the information set to which the particular sequence leads. Formally, for every sequence $\sigma_i \in seq_i(I)$, a set of its extensions is a set $\text{Ext}(\sigma_i) = \{\sigma_i a_j | a_j \in \chi(I)\}$. The sequence σ_i' is a prefix of σ_i ($\sigma_i' \sqsubseteq \sigma_i$) if σ_i is obtained by finite number of extensions of σ_i'.

In this sequence representation, the tuple of real-valued *extended utility functions* is a tuple $g = (g_1, \dots, g_n)$, where $g_i : \Sigma_1 \times ... \times \Sigma_n \to \mathbb{R}$ is defined for each player i as

$$g_i(\sigma_1, \dots, \sigma_n) = \sum_{z \in Z | \forall \sigma_i \in \sigma : seq_i(z) = \sigma_i} u_i(z) C(z). \tag{1.1}$$

If no leaf is reachable with a given tuple of sequences, a value of g_i is 0 for every i.

Example 5. Consider a sequence-form representation 1.6 of an extensive-form game depicted in Figure 1.4. Both nodes from the lower level of the game tree belong to the same information set, denote it I. Therefore, $seq_1(I)$ is equal to L. Similarly, both $inf_1(Ll) = inf_1(Lr) = I$. The set of extensions of L is a set $\text{Ext}(L) = \{Ll, Lr\}$. For comparison, the already reduced normal form of the same game has a size 2×3. Written this way, the sequence form is larger than the normal form, however, many of the entries in the game matrix in Figure 1.6 correspond to cases where the utility function is defined to be zero because the given pair of sequences does not correspond to a leaf node in the game tree. Each utility that is defined at a leaf in the game tree occurs exactly once in the sequence-form table. Thus, if g was represented using a sparse encoding, only five values would have to be stored. Compare this to the induced normal form, where all of the six entries (or eight entries in the fully induced form) correspond to leaf nodes from the game tree.

Given a behavioral strategy β_i of player i, some sequences will be preferred over others in a sense of their likelihood to be played. This concept is called a realization plan of β_i and for each player i it is a function $r_i : \Sigma_i \to [0,\, 1]$ defined as $r_i(\sigma_i) = \prod_{a \in \sigma_i} \beta_i(a)$. Each value $r_i(\sigma_i)$ is called a realization probability. Intuitively, realization plans compute the conditional probability of playing a sequence σ_i when considering a behavioral strategy β_i. In fact, given just the realization plan r_i, the behavioral strategy β_i can be fully recovered from it and vice versa. Consequently, a realization plan can be seen as an equivalent representation for any behavioral strategy.

Definition 5 (Realization plan [33])**.** *A realization plan for player $i \in N$ is a function $r_i : \Sigma_i \to [0, 1]$ satisfying the following constraints.*

$$\begin{aligned} r_i(\emptyset) &= 1 \\ \sum_{\sigma'_i \in Ext_i(I)} r_i(\sigma'_i) &= r_i(seq_i(I)) \qquad && \forall I \in I_i \\ r_i(\sigma_i) &\geq 0 && \forall \sigma_i \in \Sigma_i \end{aligned} \tag{1.2}$$

The first constraint says that the conditional probability of playing an empty sequence when considering any behavioral strategy is always 1. The second constraint ensures that the realization plans of sequences leading to the states reachable by one action from the information set I sum up to the realization plan of reaching set I. This also allows the original behavioral strategy to be possibly recovered afterwards, just from these equations. Finally, the third constraint demands the realizations of all sequences to be nonnegative. It comes naturally, since the realization plans are probabilities, which are by definition at least zero. Together, these constraints fully characterize all possible behavioral strategies of player i for a given set of sequences Σ_i.

Definition 6 (Sequence-form representation [33])**.** *Let G be an imperfect-information game of perfect recall. The sequence-form representation of G is a tuple (N, Σ, g, C), where*

- *N is a set of n players ;*
- *$\Sigma = (\Sigma_1, \dots, \Sigma_n)$, where Σ_i is the set of sequences available to player i;*
- *$g = (g_1, \dots, g_n)$, where $g_i : \Sigma \to \mathbb{R}$ is the payoff function for agent i; and*

- $C = (C_1, \ldots, C_n)$*, where* C_i *is a set of linear constraints on the realization probabilities of agent* i*.*

To sum up, the sequence form is much smaller than the normal form, or even reduced normal form. The reason is that every sequence contains only moves of one player along the path from the root. The maximum number of sequences is therefore bounded by the number of nodes in the tree. The realization plan is described by a polynomial number of constraints (one equation for each information set), which uniquely represents any EFG.

Finally, the set of opponents of player i is often denoted as $-i$. This notation is frequently used for tuples of strategies restricted to the opponents, such as $\delta_{-i} = \delta \backslash \delta_i$ or $\pi_{-i} = \pi \backslash \pi_i$. In games with only two players, $-i$ is the only opponent of player i. Therefore, his set of sequences is referred as Σ_{-i} and the same notation is used also in functions seq_{-i} or inf_{-i}.

1.1.3 Repeated games

At the beginning of this chapter, the normal-form games were defined as games in which the players move simultaneously and only once. On the other hand, extensive-form games seem to represent an opposite concept – games in which the players move point to point and in every situation they face new decisions. *Finitely repeated games* (FRG) consist of a finite number of repetitions of a normal-form game and are hence a type of games lying in between of those two extremes.

Definition 7 (Finitely repeated game [33])**.** *A finitely repeated game is a tuple* $G = (N, A, u, k)$*, where*

- (N, A, u) *is a normal-form game; and*
- k *is a finite number of repetitions.*

FRGs can be seen as a special case of extensive-form games with imperfect information, a representation used for large game trees with a specific structure. For example, consider an extensive-form representation of twice repeated prisoner's dilemma shown in Figure 1.7.

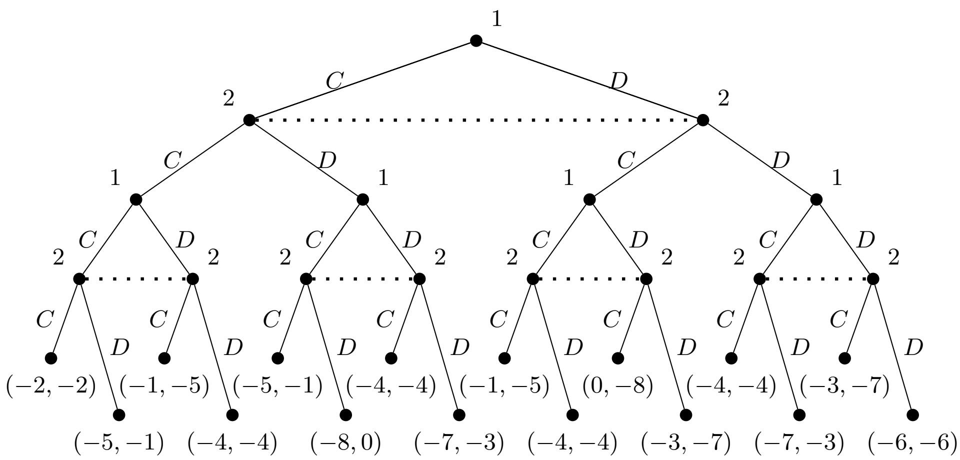

Figure 1.7: An equivalent extensive-form representation of twice repeated prisoner's dilemma. The figure follows a standard denotation of an extensive-form game. Note that subtrees rooted at the same level are isomorphic (up to utility).

Because repeated games are a special case of extensive-form games, it is possible to reason about them the same way as about regular extensive-form games. However, recall that in imperfect-information extensive-form games, a pure strategy for player i specifies an action for every information set which belongs to that player. A naive representation of pure strategies in FRGs as a mapping from each possible history to an action is hence very inefficient, since the size of every pure strategy is exponential in the number of repetitions. More precisely, a strategy for k repetitions of a normal-form game in which a player selects one of n actions requires defining $\frac{n^k-1}{n-1}$ different actions. On the other hand, in the repeated prisoner's dilemma depicted in Figure 1.7, the strategy that chooses C in every round of the game can be represented by just the original normal-form pure strategy C, or the strategy that cooperates until the opponent defects can be implemented by simply playing C and checking whether the opponent defected in the previous round. Therefore, as a representation that captures this structure of strategies in FRGs can be used a finite-state game automaton.

Definition 8 (Repeated game automaton [33])**.** *A game automaton for player i in a finitely repeated game $G = (N, A, u, k)$ is a tuple $M = (Q_i, q_i^0, B_i, \delta_i)$, where*

- *Q_i is a finite set of states;*
- *$q_i^0 \in Q_i$ as the initial state;*
- *$B_i : Q_i \to A_i$ is an output function that assigns an action to every state; and*
- *$\delta : Q_i \times A_{-i} \to Q_i$ is the transition function that assigns a state to every pair of a state and tuples of actions of the other players.*

The set of all automata of player i is denoted M_i and it replaces the set of pure strategies in the original extensive-form game. Even in finite games, for every pure strategy, there exists an infinite number of automata, which implements this strategy. These automata differ either in the structure (e.g. by having states unreachable in a game being played, similarly to unreduced pure strategies) or just in the names of the states. Furthermore, in the same way as in extensive-form games, *mixed automata strategies* are finite probability distributions over the set of all automata. Also behavioral strategies have their automata counterparts, which are the automata with probabilistic transition functions: *behavioral automata strategies*. At the beginning of a game, each player can select one automaton strategy, which will play the game instead of the player.

Example 6. Consider a game automaton depicted in Figure 3.3. This automaton represents the Tit for Tat strategy for playing the game of repeated prisoner's dilemma, an example of which is shown in Figure 1.7. Tit for Tat is a well-known strategy in the field of repeated games which mimics the action the opponent made in the previous round. The equivalent pure strategy in k-times repeated prisoner's dilemma has size $2^k - 1$, while the automaton has only 2 states and 4 edges.

Because there is an infinite number of automata which implements every pure strategy, the complexity of any such strategy was proposed to be equal to a size of the smallest automaton implementing the pure strategy [37].

Structural complexity

The structural complexity measure of an automaton $M = (Q, q^0, B, \delta)$ captures the notion of a size of M's structure and it generates a partial ordering on the set of possible automata of each player. The traditional measure referred by most literature is the state measure [37], defined as $\mu_Q = |Q|$. The lower the number of states is, the less costly is the strategy the automaton implements.

However, this measure does not capture the whole complexity of the automaton, which could also emerge from the size of the transition function. In [38], the authors

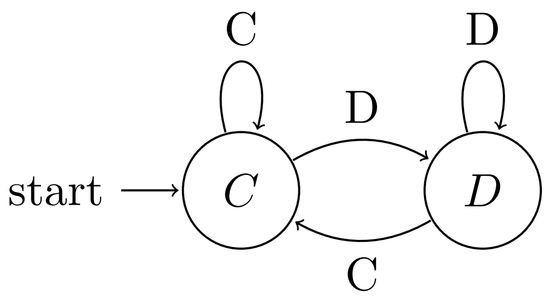

Figure 1.8: An example of a game automaton playing the game of repeated prisoner's dilemma. Each state is labeled by the action the automaton assigns to that state, while the label above every transition denotes the action of the opponent. The strategy the automaton implements is the so-called Tit for Tat, the strategy which mimics the action the opponent made in the previous round.

proposed the structural complexity measures distinguishing such cases. The measure $\mu_\delta = |\delta|$, where $|\delta| = |\{(q,a) | \exists q' \; \delta(q,a) = q'\}|$ represents the preference of automata with lower number of transitions. The size of any transition function δ is hence equal to a distinct number of transitions of an automaton. Other two two-dimensional measures $\mu_{Q,\delta} = (|Q|, |\delta|)$ and $\mu_{\delta-Q,\delta} = (|\delta| - |Q|, |Q|)$ balance the number of transitions and the number of states. The difference between $\mu_{Q,\delta}$ and $\mu_{\delta-Q,\delta}$ is that the first one does not capture the case when the states are strictly cheaper than the transitions.

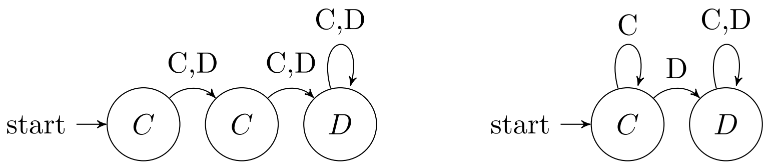

Figure 1.9: An example of two automata which under two different orderings generated by complexity measures $\mu_{Q,\delta}$ and $\mu_{\delta-Q,\delta}$ have opposite infima. The figure follows a standard denotation of game automata.

Example 7. Consider the two automata depicted in Figure 1.9. Both use 3 transitions, automaton on the left uses 3 states while automaton on the right uses only 2. Measure $\mu_{Q,\delta}$ would therefore suggest that the right automaton is less complex than the left automaton. On the other hand, measure $\mu_{\delta-Q,\delta}$ picks the left automaton as less complex, because it concludes that the transitions of this automaton require no monitoring at all, while the right automaton has to distinguish between the actions C and D of the opponent in the initial state.

The linear combination of the number of states and the number of transitions (e.g. the one used in measure $\mu_{\delta-Q,\delta}$) can be seen as employing different costs of maintaining the strategy in the memory (the number of states) and monitoring the progress of a game (the number of transitions). All of the presented measures are pairwise different, which means that there exist two automata M_1 and M_2, such that $\mu_1(M_1) \leq \mu_1(M_2)$ while $\mu_2(M_1) > \mu_2(M_2)$ for any $\mu_1 \neq \mu_2$ from the presented measures. Two such automata were already shown in Figure 1.9 for measures $\mu_{Q,\delta}$ and $\mu_{\delta-Q,\delta}$; and similar examples can be found for any pair of the complexity measures described so far.

Besides the utility, the complexity measure is yet another property of strategies the players might consider. The concept of strategies in games played by automata is hence different from the perspective of traditional games, as players might strive to balance the potentially achievable utilities of any strategy with the required complexity of implementing it.

1.2 Solution concepts

In games, players have a plenty of strategies to choose from. However, only a fraction of the possible strategies is rational with respect to the utility. For example, having a royal flush in poker, it is obviously unwise to fold the game. In contrast to the decision theory, when only one agent acts, assessing the rationality of strategies in multi-player games depends on the strategies of the opponents. Raising might be a good strategy with full house against flush, but whether to raise against quads might be questionable. Rationality in games is therefore always related to strategy profiles, rather than independent strategies.

Game equilibria are the central solution concepts of game theory, describing optimal strategy profiles. To evaluate the optimality of a profile, the players can use an *expected utility.* Taking other players' strategies as given, every player that plays a mixed strategy can gain a various range of outcomes. An expected utility for player i in a strategy profile δ is defined as

$$u_i(\delta) = \sum_{a \in A_1 \times \ldots \times A_n} u_i(a) \prod_{j=1}^{n} \delta_j(a_j). \tag{1.3}$$

By $\delta_j(\mathrm{a}_j)$ is denoted a probability of player j taking j^{th} action from a. Now it is possible to ask which strategy is the best against fixed opponents' strategies.

Definition 9 (Best response [33])**.** *Given a normal-form game G = (N, A, u), a strategy δ_i^* of player i is a best response to strategies δ_{-i} of player i's opponents if and only if*

$$u_i(\delta_i, \delta_{-i}) \leq u_i(\delta_i^*, \delta_{-i}) \qquad \forall \delta_i \in \Delta_i. \tag{1.4}$$

1.2.1 Nash equilibrium

As every player intends to do his best to maximize his utility and considers the decision-making of his opponents, the reasoning about optimal strategy profile from the perspective of different players might go through an infinite chain of changes. However, at a certain point of finding their best responses, the players might realize that changing their strategy would not lead to earn more than with their current strategy. This concept of balance is called an equilibrium.

Definition 10 (Nash equilibrium (NE) [33])**.** *Given a game G = (N, A, u) and a strategy profile $\delta_{Nash} = (\delta_1, ..., \delta_n) \in \Delta$, δ_{Nash} is a Nash equilibrium if and only if for each player i it holds that δ_i is a best response to δ_{-i}.*

If the strategy profile allows no one to benefit from changing his strategy, the situation remains stable. It has been proven that in every game with finitely many players and with the finite set of pure strategies, there has to be such strategy profile, although it might consist of mixed strategies.

Theorem 2 (Existence of Nash equilibrium [1])**.** *Every game with a finite number of players and action profiles has at least one Nash equilibrium.*

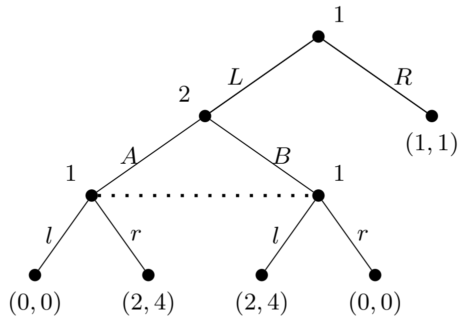

Figure 1.10: An example of an extensive-form game with imperfect information, the same as in Figure 1.4. The figure follows a standard denotation of an extensive-form game.

Example 8. Consider a game depicted in Figure 1.10, which is the same as in Figure 1.4. Given that the second player will move left, the expected utility of the first player playing L with probability 0.1, R with probability 0.9, l with probability 0.3 and r with probability 0.7 is

$$u_1(((L = .1, R = .9; l = .3, r = .7), A)) = 0.1 \cdot 0.3 \cdot 0 + 0.1 \cdot 0.7 \cdot 2 + 0.9 \cdot 1 = 1.04. \quad (1.5)$$

This strategy is not a best response against first player's A, since playing r with higher probability guarantees a higher expected utility. There are four Nash equilibria in this game and in one of them the first player's strategy is a pure A. The equilibria are (Ll, B), (Lr, A), $(R*, (A = .5, B = 0.5))$ and $((Ll = .5, Lr = .5), (A = .5, B = .5))$.

Due to Theorem 2, it is known that Nash equilibrium always exists in every finite game. Such property distinguishes the problem of computing Nash equilibrium from known NP-complete problems, which are defined as decision problems that do not always have to have solutions. The problem of finding a sample Nash equilibrium hence belongs to another, less familiar complexity class that is a subclass of the class of problems for which the solution is guaranteed to exist. This class is called *PPAD*, which means "polynomial parity argument, directed version".

Definition 11 (Complexity class PPAD [39])**.** *A problem A is in PPAD if there is a polynomial time reduction from A to the End-of-Line problem, where End-Of-The-Line is the following search problem: The input consists of a directed graph in which each node has in-degree and out-degree at most 1. The graph is given by a polynomial-time computable function $f(x)$ that returns the predecessor and successor of x. In addition, a node v with a successor but no predecessor is given. Find a node $t \neq v$ that has no successor or no predecessor.*

It is known that computing a Nash equilibrium is a complete problem for this class.

Theorem 3 (Complexity of finding Nash equilibrium [33])**.** *The problem of finding a sample Nash equilibrium of a general-sum finite game with two or more players is PPAD-complete.*

Even though PPAD-completeness is a weaker evidence of intractability than NP-completeness, there is still no known polynomial algorithm for complete problems in this class. However, in some special cases, the Nash equilibrium can be found efficiently.

Theorem 4 (Computing best response in zero-sum games with two players [33])**.** *The realization plan r_1 of the first player is a best response to a given realization plan*

r_1 *of the first player of a zero-sum finite extensive game with two players if and only if it is a solution of the following linear program.*

$$
\begin{aligned}
\max \quad & \sum_{\sigma_1 \in \Sigma_1} r_1(\sigma_1) \sum_{\sigma_2 \in \Sigma_2} g_1(\sigma_1, \sigma_2) r_2(\sigma_2) & \\
s.t. \quad & r_1(\emptyset) = 1 & \\
& \sum_{\sigma_1' \in Ext_1(I)} r_1(\sigma_1') = r_1(seq_1(I)) & \forall I \in I_1 \\
& r_1(\sigma_1) \geq 0 & \forall \sigma_1 \in \Sigma_1
\end{aligned}
\tag{1.6}
$$

The criterion ensures that the realization plan of the first player is optimal against the realization plan of his opponent, while the constraints guarantee the realization is well-formed. Using the dual to the program computing the best response, it is possible to formulate a linear program computing the equilibrium.

Theorem 5 (Computing Nash equilibrium in zero-sum games with two players [33])**.** *The realization plan* r_2 *of the second player belongs to a strategy profile of a Nash equilibrium of a zero-sum finite extensive game with two players if and only if it is a solution of the following linear program.*

$$
\begin{aligned}
\min \quad & v_0 & \\
s.t. \quad & v_{inf_1(\sigma_1)} - \sum_{I' \in inf_1(Ext_1(\sigma_1))} v_{I'} \geq \sum_{\sigma_2 \in \Sigma_2} g_1(\sigma_1, \sigma_2) r_2(\sigma_2) & \forall \sigma_1 \in \Sigma_1 \\
& r_2(\emptyset) = 1 & \\
& \sum_{\sigma_2' \in Ext_2(I)} r_2(\sigma_2') = r_2(seq_2(I)) & \forall I \in I_2 \\
& r_2(\sigma_2) \geq 0 & \forall \sigma_2 \in \Sigma_2
\end{aligned}
\tag{1.7}
$$

The variable v_0 represents the first player's expected utility in equilibrium. Each other variable $v_{inf(\dots)}$ can be understood as the portion of this expected utility that player 1 will achieve under his best-response realization plan in the subgame starting from information set $inf(\dots)$, given second player's realization plan r_2. Note that the first constraint and the criterion form the dual of the linear program computing best response. The first constraint in this linear program hence ensures the first player plays the best response in each information set. The other constraints guarantee that the second player's realization plan is well-formed.

The complexity of solving the game is dependent on the size of the linear program describing the desired equilibrium. Every linear program can be solved in time polynomial in its size [40, 41], even though the most well-known simplex algorithm is faster, but with worst-case exponential time [42]. The size of the linear program computing Nash equilibrium in zero-sum games in extensive form with two players is polynomial in the size of the tree, the equilibrium can be hence found in polynomial time.

1.2.2 Correlated equilibrium

Correlated equilibrium describes a situation when players are given a chance to coordinate according to an external event. This mechanism for coordination is called *a correlation device* and can be imagined as a signaling device (e.g. the traffic lights) helping the players to synchronize. In the canonical representation of correlated equilibrium, the recommendations to the players are their own moves, not arbitrarily signals. It was shown that this can be assumed without loss of generality [43]. The essential aspect of correlated equilibrium is that even if the probability distribution that correlation device uses to generate signals is known, each player is not aware of the recommendations given to the other players. All he knows is that they are proposed the best move with respect to the opponents.

Computing a correlated equilibrium is easier than computing a Nash equilibrium, because the set of correlated equilibrium distributions is convex. This property enables the description to be more compact and thus less computationally demanding.

Normal-form CE

In normal-form games, the players' strategies form a correlated equilibrium when given a move according to the correlation device λ, none of the players have an intention to unilaterally deviate from the recommended strategy, given his posterior on the recommendations sent to the opponents.

Definition 12 (Normal-form correlated equilibrium)**.** The distribution λ on Π is a correlation device of a correlated equilibrium if and only if for all i, every $\pi_i \in \Pi_i$ with $\lambda(\pi_i) > 0$ and every $\pi'_i \in \Pi_i$,

$$\sum_{\pi_{-i} \in \Pi_{-i}} u_i(\pi_i, \pi_{-i})\lambda(\pi_{-i}|\pi_i) \geq \sum_{\pi_{-i} \in \Pi_{-i}} u_i(\pi'_i, \pi_{-i})\lambda(\pi_{-i}|\pi_i) \tag{1.8}$$

A correlation device λ is implemented by randomly picking a strategy profile π^* according to the distribution λ. Then it privately recommends the component π_i of π^* to each player i before the game starts. It was shown that the equilibrium can be found in polynomial time even in multi-player games [44, 45].

Wife \ **Husband**	LW	WL
LW	2,1	0,0
WL	0,0	1,2

Figure 1.11: An example of correlated equilibrium in normal-form game [33]. The figure follows a standard denotation of a normal-form game.

Example 9. Consider the Battle of Sexes game depicted in Figure 1.11. This game has a unique mixed-strategy Nash equilibrium δ_{Nash} =((1/3, 2/3), (2/3, 1/3)) which guarantees each player an expected payoff of 2/3. However, imagine the scenario of correlated equilibrium, in which the players can observe an external event and coordinate their actions according to this event. Their strategies are hence conditioned on this event. For example, in case they decide their strategies based on flipping a fair coin, their strategies are dependent on the possibility of "head" or "tail". A pair of strategies "WL if heads, LW if tails" forms an equilibrium in this richer strategy space, because whenever one player adopts this strategy, the other one can only lose if he decides to play differently. Moreover, the expected payoff of each player is $0.5 \cdot 2 + 0.5 \cdot 1 = 1.5$, which is strictly more than they receive in the mixed-strategy equilibrium in the original game.

The example shows that coordination might help the players to obtain higher payoffs. However, note that also the Nash equilibrium satisfies to conditions of a correlated equilibrium.

Theorem 6 (Every Nash equilibrium is a correlated equilibrium [33]). *For every Nash equilibrium δ_{Nash} there exists a corresponding correlated equilibrium λ.*

Extensive-form CE

The correlation device in extensive-form games differs from the correlation device in normal-form games. In EFGs, the device recommends an action just in the moment when an information set in which the action can be taken is reached. For this reason, the recommendations become rather local and players are less aware of the intended progress of the game. Consequently, also the set of extensive-form correlated equilibria (EFCE) in an extensive-form game is larger than the set of CE of the equivalent normal-form game. Similarly to the normal form, the distribution λ describes a correlation device of EFCE if any rational player follows his recommendations while assuming that

1. all other players also follow their recommendations – this is a standard assumption for any equilibrium; and
2. when any player decides to deviate from the recommended move, he gets no further information. Consequently, the posterior of the player at all following information sets is equal to that at the last information set before he deviates.

This assumption can be made without any loss of generality, because the EFCE can be defined using only reduced strategies [46]. Moreover, because in CE the players do not have an intention to deviate when they are recommended the whole strategy at once, they can not have this intention when the same strategy is revealed to them as the game progresses.

Example 10. Consider an extensive-form game with two players depicted in Figure 1.12. This game is described in [46] as a costless variant of a signaling game introduced in [47]. In this game, a professor (player 2) decides whether to accept a student (player 1) applying for a summer research job. With an equal probability, the student is consciously either well-educated (type G) or inexperienced (type B). He sends a costless signal X or Y to the professor, who is able to distinguish between the signals, but the education level of the student is hidden from him. The student is hence capable to impersonate an experienced person and get the job even if his education is poor. In case the professor lets the student work with him, the payoffs are either $(4, 10)$ for a well-educated student or $(6, 0)$ for an inexperienced student. Refusal leads to the utility $(0, 6)$ for both types.

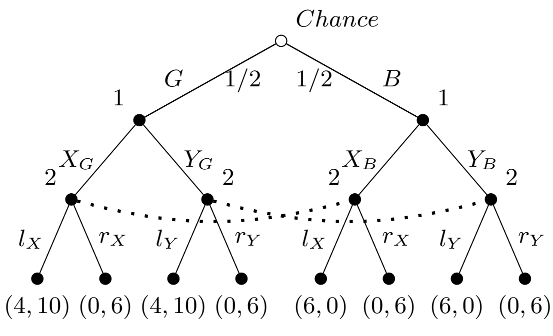

Figure 1.12: Signaling game with costless signals (X or Y) for the first player [46]. The figure follows a standard denotation of an extensive-form game.

In [46], the authors analyzed the correlated equilibria of this game and showed that in every Nash equilibrium the professor plays $r_X r_Y$ with probability 1. Therefore, the student never gets hired and both players always receive the payoff $(0, 6)$.

On the other hand, the situation changes when the players are able to coordinate. The signal for the well-educated student is now hidden from the student with insufficient skills, which prevents any inexperienced student to overvalue his education. The professor is hence able to evaluate student's knowledge more precisely. Therefore, this situation is advantageous for both of them. For example, a distribution λ_{EFCE} which with the equal probability picks one of the following strategies – $\{(X_G X_B, l_X r_Y), (X_G Y_B, l_X r_Y), (Y_G X_B, r_X l_Y), (Y_G Y_B, r_X l_Y)\}$ – is an EFCE. Note that to prohibit an inexperienced student from impersonating a knowledgeable student, λ_{EFCE} recommends the signal corresponding to the correct type exactly in half of the cases. Otherwise, the student would do exactly the opposite of what he is given and λ_{EFCE} would not be an EFCE. The expected payoff of this equilibrium is $(3.5, 6.5)$, which is strictly more than each of the players obtains in the normal-form correlated equilibrium.

It was shown that computing an EFCE in more general classes of games than two-player games without chance moves can not be done using sequence form [46]. In [48], the author points out that not only a sequence form, but generally no compact representation like sequence form can be expected when computing EFCE in these classes of games. In perfect-recall games of two players, the structure of information sets enables the recommendations to be generated uniquely for each information set, while maintaining the compact description of sequence form. In more general classes of games, the constraints satisfied by a sequence form are only a necessary condition. Consequently, the size of the game representation used for describing EFCE is always asymptotically equal to the size of an equivalent normal-form game.

To compute an EFCE, the players compare the expected utility of a recommended action with an expected utility of a possible deviation. The players hence evaluate the optimality of the recommendation in every information set reachable from its preceding information set in which the recommendation is given. The strategy relevant for a given sequence of recommendations is called *agreeing.*

Definition 13 (Agreeing strategy [46])**.** *A strategy* $\pi_i \in \Pi_i$ *agrees with a sequence* σ *if and only if* $\forall a \in \sigma\ \exists I \in I_i :\ \pi_i(I) = a$*. A partial strategy profile* π_J *where* $J \subseteq \{1, .., n\}$ *agrees with a node* $h \in H$ *if and only if every* $\pi_i \in \pi_J$ *agrees with* $seq_i(h)$*.*

Furthermore, the set of all agreeing strategies for sequence σ or (possibly partial) strategy profiles for node h is denoted $agr(\sigma)$ and $agr(h)$, respectively.

The conditions for EFCE in multi-player games can be expressed using inequalities. First, in the equilibrium, no player has an intention to deviate from his received recommendations. To consider a deviation, a player i calculates the expected payoff contribution of $a \in A_i$ as a sum of expected utilities from all leaves reachable by playing a.

$$u(a) = \sum_{\substack{t \in Z: \\ a \in seq_i(t)}} u_i(t) C(t) \sum_{\pi \in agr(t)} \lambda(\pi) \tag{1.9}$$

The second constraint then compares the expected payoff contribution of action $a \in A(I')$ with the potential payoff the player is able to obtain in case he deviates from his recommendation at this information set I'. The deviation will affect the expected utilities in all subsequent information sets. The optimal expected payoff at information set I (under the assumption the player is recommended move $a \in A(I')$ where I'

precedes I) is the maximum of the utilities the player expects for actions $b \in A(I)$.

$$\begin{aligned} v(I,a) \geq & \sum_{\pi_i \in agr(seq_i(I')a)} \sum_{\substack{t \in Z: \\ seq_i(I)d = seq_i(t)}} \sum_{\pi_{-i} \in agr(t)} u_i(t) C(t) \lambda(\pi_i, \pi_{-i}) \\ & + \sum_{\hat{l}: seq(\hat{l}) = seq_i(I)b} v(\hat{l}, a) \end{aligned} \tag{1.10}$$

Note that by definition, once the player decides not to follow the recommendation, he starts to ignore every further signal. Equivalently, he may as well not receive the recommendations any more. Finally, the last equation makes sure that the expected payoff contribution and the optimal expected payoff of every move $a \in A(I)$ is equal.

$$u(a) = v(I,a) \tag{1.11}$$

Because the introduced constraints guarantee that no player has an intention to deviate, every probability distribution λ on pure strategy profiles that satisfies these constraints represents an EFCE. Similarly to the case of normal-form games, the correlation device picks one of the strategy profiles according to the equilibrium distribution before the game starts. Then it privately recommends the appropriate moves to every player once they reach a state where they can take that move.

Theorem 7 (Computation of EFCE [48])**.** *Every multi-player, perfect-recall extensive-form game, which may have chance moves, has an EFCE, which can be computed in polynomial time.*

To find an EFCE in polynomial time, the authors applied the ellipsoid algorithm to the dual of the system of equations 1.9, 1.10 and 1.11 that characterizes the set of EFCE. The linear program has polynomially many constraints, but exponentially many variables, since every pure strategy profile is described by exactly one variable. Thus for the dual the opposite holds, which makes it suitable for the ellipsoid algorithm. However, this approach does not apply to situations when a specific EFCE is required (e.g. the one maximizing the sum of expected utilities of all players called MAXPAY).

Theorem 8 (Computational complexity of equilibrium maximizing payoff [46])**.** *For every multi-player perfect-recall extensive-form game with chance moves, the problems MAXPAY-NE, MAXPAY-CE and MAXPAY-EFCE are NP-hard.*

Chapter 2

Finite Automata Play an Extensive-Form Game

This chapter considers extensive-form games with perfect recall in which the players are restricted to carry out their strategies using finite automata. In other words, the original model of pure strategies in traditional game theory is replaced by a model of strategies as automata. Every player then has to choose an automaton strategy, according to which the game will be played. The chapter first introduces the model of automata representing a strategy in the context of extensive-form games. It describes how to construct automata consistent with the tree of the game and presents several methods for evaluating and minimizing the complexity of the automata.

The approach of replacing a pure strategy with an automaton is known to be fruitful in a subset of extensive-form games called repeated games, already described earlier. Recall that in these games, a designated normal-form game is played again and again, either finitely or infinitely many times. The theory focuses on two main phenomena: it describes the difference between the structure of equilibria in games with traditional strategies and in games with automaton strategies; or it analyzes the effect of restricting the set of possible strategies to strategies with bounded automaton complexity. For example, by using automata, the theory was able to describe several deviations from the predictions of traditional game theory observed in the experiments with human participants, such as the effort to establish cooperation in repeated prisoner's dilemma. In contrast to the traditional solution concepts, in games with automata, the cooperation can be included in the equilibrium strategies of finitely repeated PD [49]. The reason is that finite automata provide a cognition model of limited ability to memorize the past and to carry out demanding computational tasks. Therefore, the model enables to forgive the opponents and rather to focus on following easily implementable simple rational strategies. Such property corresponds well to real-world settings, in which in implementing a strategy the optimality is only one of the criteria for choosing it. More importantly, the player also considers how much memory is needed to realize such strategy. Automata strategies intuitively capture such perspective by providing a suitable way to measure the complexity of different strategies.

Similar thoughts encouraged the effort to generalize this model also to general extensive-form games. A lot of real-world games (and also the tabletop ones) enable similar actions to be repeated in different situations in the game. Moreover, many intuitive rational strategies describable in natural language and often used by humans are easily implementable using automata. In order to obtain structural results similar to automata in finitely repeated games, the definitions from the theory of automata playing repeated games have to be suitably generalized. Such generalization is not trivial, as the set of feasible actions might not be the same in different information sets.

A complexity measure on automata in extensive-form games is introduced to speak about optimality in a sense of compensating the costs of maintaining an automaton by an average utility it can provide. Such balancing can be seen as a realization of Simon's satisficing principle [50], which states that bounded-rational players do not always look for optimal strategies, but rather play the not-so-complex strategies with satisfiable expected utilities.

2.1 The model

In repeated games, the structure of the game tree is very specific. In every information set, the player can choose from the same set of feasible actions. Therefore, the automata do not have to consider whether the proposed action is applicable – it always is. The player who behaves according to an automaton can hence easily relate the state of the automaton to the situation in the game tree and recognize an appropriate action he should take. Then, he immediately observes the actions of his opponents and identifies a corresponding transition.

In contrast, an automaton playing an extensive-form game has to be aware whether the action it prescribes is even feasible. Moreover, general game trees can be very asymmetric with players taking their moves aperiodically. The structure of the automaton has to comply with it. Such generalization is non-trivial and it is the main contribution of this thesis. First, the definition of a repeated game automaton is generalized in order to ensure an automaton is consistent with an extensive-form game.

Definition 14 (Game automaton)**.** *Every game automaton is a tuple $M = (Q, q_0, A, B, O, \delta)$, where*

- *Q is a set of states of the automaton; with*
- *q_0 as an initial state;*
- *A is a set of actions proposed by the automaton;*
- *$B : Q \to \{A \cup \emptyset\}$ is a behavior function prescribing an automaton action in a given state;*
- *O is a set of observations available for the automaton; and*
- *$\delta : Q \times O \to Q$ is a transition function of the observation of player.*

Note that the game automaton is a special case of a Moore machine, adjusted to comply with the language of game theory. Similarly to a repeated-game automaton, every state of the automaton prescribes an action the player should take. Moreover, the behavior function can also prescribe an explicit "noop" action, denoted as $\emptyset$, if the player is not obligated to take any action in a current state. Once he receives the information of what happened in the game, the transition function transits the automaton to a new state from an old one. Given a series of observations $o_1 o_2 \dots o_m; o_i \in O$, the state $\delta(\delta(\dots\delta(q_0, o_1), o_{m-1}), o_m)$ reached from the initial state is denoted $\delta(q_0, o_1 o_2 \dots o_m)$.

A player who represents his strategy using an automaton always correlates the situation he faces in the game with the state of the automaton and vice versa. The schema of such correlation is shown in Figure 2.1.

Example 11. Consider a schema of correspondence of a pure strategy (L, r) of the first player represented by an automaton and a game tree shown in Figure 2.1, in which the strategy is being played. The set of actions the automaton prescribes is a subset of the set of actions of the first player in the game tree – the actions of the strategy(L, r). The set of possible observations is a set of sets of sequences, such that each sequence leads from an information set of the first player to one of his immediately succeeding information sets. For this automaton, it is a set of sequences $\{LA, LB\}$. The automaton's initial state corresponds to the root state of the tree, in which the automaton prescribes an action L. Independently on the opponent's action, which is unknown to the first player because they both lead to the same information set, the transition function changes the state of the automaton to the lower one, in which it prescribes to play an action r. After taking the action the game ends, so no observation can be made in this second state.

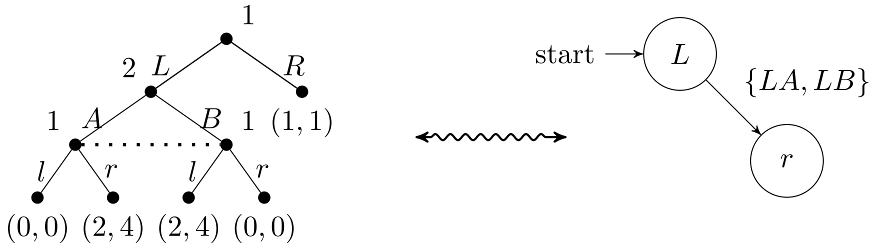

Figure 2.1: A correspondence of a pure strategy (L, r) of the first player in a game tree with a game automaton, which represents it. The figure follows the standard denotations of an extensive-form game and a game automaton.

Automata well describe the inner structure of the pure strategy and are hence suitable for finding strategic patterns in the game tree, such as repetitions or symmetries. Even though a number of information sets in a game can be very large, the meaning of actions taken in different information sets can be similar. For example, in games like Tic-Tac-Toe, the player might want to play the same action with respect to a given axis of reflection in all states which are the same in this reflection. The player hence recognizes a "prototype" action, so that he is always able to distinguish which action he has to play in a given situation. To formalize the intuition of maintaining a compact representation of actions, he can use *abstracted actions.*

Definition 15 (Abstracted actions). *Given an extensive-form game G = (N, H, Z, A, χ, ρ, σ, u, C, I), let P_A be a partition of A_i into mutually disjoint subsets, such that no two actions belonging to the same part of P_A can be performed in the same information set. Then the parts of partition P_A form the set of abstracted actions for player i, associated with P_A.*

Instead of prescribing an action from a game tree, an automaton can rather prescribe an abstracted action. Player's abstracted strategy is a strategy consisting of abstracted actions. Using abstracted actions, it is possible to easily describe intuitive strategies. As an example of a simple strategy, consider a strategy in Tic-Tac-Toe, which prescribes an action which completes the line in all symmetric situations. Note, that the whole game history does not have to be maintained by this strategy. A perfect memory is not required, the strategy depends only on the state of the board. This is the reason, why the automata implementing such strategies can be seen as the models of players with bounded memory.

The same principle can be also applied to observations. Note that in Figure 2.1 the sequences from the game tree served as observations for the automaton. Such interpretation is suitable for the definition of extensive-form games, where no other information is provided. Note that the definition of game automaton is more general and does not assume any specific representation of games, so that it can be used also as a model of strategies in extended representations of sequential games, e.g in first-order game logic of general game playing (GGP) [51], in which the observations are satisfied first-order formulas. However, in extensive-form games, the set of observations available to an automaton is limited to sequences of actions in the game tree.

Definition 16 (Extensive-form-game observations)**.** *Given an extensive-form game G = (N, H, Z, A, χ, ρ, σ, u, C, I), an extensive-form-game observation is a set of sequences of all players' actions interconnecting either two immediately succeeding information sets of the same player i; or a root and the first information set of player i under the root. The set of all possible extensive-form-game observations is denoted $\overline{\Sigma_i}$, while the extensive-form-game observation leading to an information set I is denoted $seq_{\overline{\Sigma_i}}(I)$.*

Example 12. In an extensive-form game depicted in Figure 2.1, the set of extensive-form-game observations for the first player is a set $\{\{LA, LB\}\}$ with only one possible observation $\{LA, LB\}$, which leads from the upper information set of this player to his lower information set. Note that neither sequence R, nor l or r belongs to this set, since they interconnect an information set of the first player with one of the leaves and therefore, they do not comply with the definition of extensive-form-game observations. The reason is that the player does not have to react to such sequences, which makes them obsolete. The set of extensive-form-game observations for the second player is also a set with only a singleton observation – $\{\{L\}\}$, which is a sequence interconnecting a root with the only information set of the second player. This information set is trivial, since it also contains only one game state.

Using extensive-form-game observations, the player is able to recognize abstracted observations.

Definition 17 (Abstracted observations)**.** *Given an extensive-form game G = (N, H, Z, A, χ, ρ, σ, u, C, I), let P_O be a partition of $\overline{\Sigma_i}$ into mutually disjoint subsets.Then the parts of partition P_O form the set of abstracted observations for player i, associated with P_O.*

Similarly to abstracted actions, instead of receiving extensive-form-game observations, an automaton can have a transition function based on abstracted observations. Note that in contrast to abstracted actions, abstracted observations do not further restrict the way the partitions are created. In the case of abstracted actions, it was required that no two actions belonging to the same part of P_A can be taken in the same information set. The reason is that if an automaton prescribed such part of P_A, which is an abstracted action, it would not be clear which one of these two possible actions it should actually play in the game. On the other hand, making abstracted observations does not require such restrictions, because the case of two possible observations obtainable in the same information set (or equivalently, in the same state of the automaton) which belong to the same part of P_O causes only the inability of the automaton to distinguish future information sets. Unlike abstracted actions, it does not necessarily prevent the automaton to play the game.

By using abstracted actions and abstracted observations, the correspondence between pure strategies in a game tree and the automata implementing them becomes more complex than the one shown in Figure 2.1. In this case, the actions the automaton prescribes and the observations the automaton uses to identify which transitions it should take do not have to directly correspond to the actions in the game tree and the extensive-form-game observations. Instead, they are abstracted.

In particular, consider how a player whose strategy is determined by an automaton plays a given extensive-form game, which is described in Figure 2.2. At the beginning, the player finds himself in a root of the game, whereas his automaton is in the initial state. In case he does not act in a root, the initial state prescribes a "noop" action and the player waits for the actions of his opponents. Now, assume the game reaches an information set I, in which the player i is supposed to take an action. According to the actions performed by all players in the game so far, player i receives an extensive-form-game observation σ. At this time, player i knows the current state q of his automaton, so based on the pair (σ, q) the player identifies an abstracted observation o for his automaton. According to the pair (q, o), the automaton transits to the next state q'. The player observes the new state q' and therefore the abstracted action $B(q')$ the automaton prescribes. Because he knows in which information set the game is, the player has to decide which action in I corresponds to an abstracted action $B(q')$ recommended by the automaton. By definition of abstracted actions, there is at most one such action. Based on this action, the state of the game changes. In this manner, every player using an automaton strategy has to continuously switch between a situation in a game tree and an automaton.

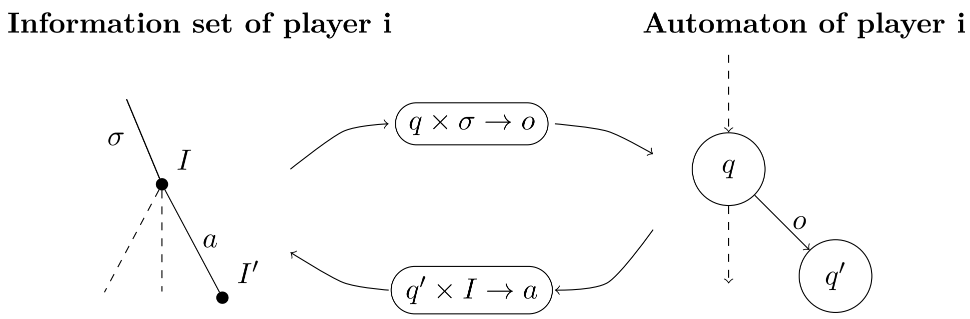

Figure 2.2: Player i plays one step in an extensive-form game using his automaton.

Example 13. Consider an extensive-form game depicted in Figure 2.3 together with an automaton playing an abstracted strategy of the first player. First, the player receives an empty observation, since he takes an action in the root. His automaton prescribes an abstracted action L, which is a singleton action directly corresponding to the same action in the game tree and the player hence takes it. Now he can receive two possible EFG observations, either LA or LB, based on the actions of his opponent. Both these observations belong to the same abstracted observation, so no matter what, the player takes the only transition available. The next abstracted action the automaton recommends is $\{m, o\}$. He has to check which of these actions is applicable in the current information set and after figuring it out, he performs it.

Therefore, in case the automaton uses abstracted actions and observations to make its structure more compact, its computational ability alone is not enough to ensure a correct game-play. Apart from the automaton, a player has to implement *distinguishing functions* which identify both which abstracted observation corresponds to a given extensive-form observation; and which action to take in the game tree for a given abstracted action. Formally, this structure consisting of an automaton and corresponding distinguishing functions is called a game-playing schema.

Definition 18 (Game-playing schema)**.** *Let* $G = (N, H, Z, A_G, \chi, \rho, \sigma, u, C, I)$ *be an extensive-form game. A tuple* (M_i, f_i, g_i) *is a game-playing schema of player i, if* $M_i = (Q, q_0, A_M, B, O, \delta)$ *is an automaton of player i,* f_i *is a* μ*-recursive function* $f_i : Q \times \overline{\Sigma_i} \to O$ *which specifies a transition in* M_i*; and* g_i *is a* μ*-recursive function* $g_i : Q \times I \to A_G$ *which distinguishes which action to play in G.*

From the perspective of behavioral game theory, game-playing schemata are models of bounded rationality which represent both structured memory of the players through

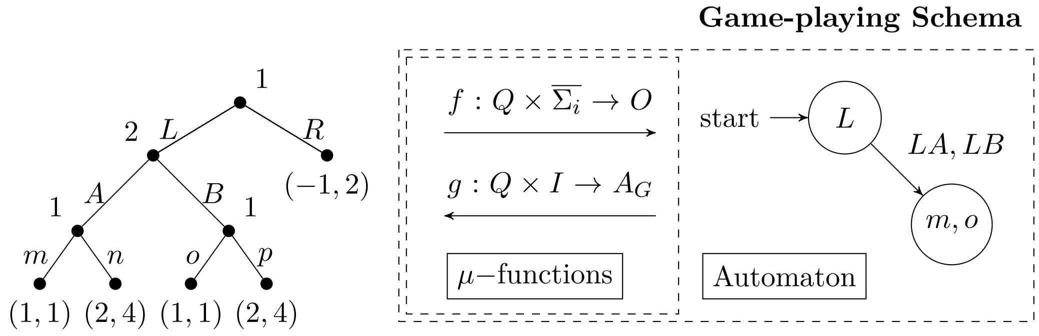

Figure 2.3: A more complex correspondence of an abstracted pure strategy in a game tree with its game automaton. The distinguishing functions identify the abstracted observation $\{A, B\}$ which enables to reduce the number of transitions to one; performing an abstracted action $\{m, o\}$ in the lower information sets in the tree. The figure follows the standard denotations of an extensive-form game and a game automaton.

automata; and computational abilities of the players through distinguishing functions. Every schema can be seen as an abstraction of its corresponding pure strategy, in which an automaton captures the inner structure of the strategy (i.e. how individual decisions of which the strategy consists relate), while the distinguishing functions ensure the abstraction is still applicable in the original game. More precisely, function f abstracts the extensive-form-game observation, while function g implements the abstracted action in the game tree. Game-playing schemata can model limited rationality of the players by introducing bounds on the size of the automata or the complexity of the distinguishing functions.

Note that since the game tree is always finite, the computational complexity of the distinguishing functions can not be analyzed in the traditional sense. Because of theorems about linear speed-ups [52], both space and time complexity can be arbitrarily altered. This can be fixed either by choosing a specific computational model which does not allow similar speed-ups, or by playing arbitrarily large games from a given class of games using the same distinguishing functions. This issue will be discussed more closely in the next chapter.

2.2 Correspondence with pure strategies

For every game G, there exists an infinite number of game-playing schemata, with various automata and distinguishing functions. However, not all schemata are able to correctly play the game. The reason is that the set of actions feasible in a given information set can be a strict subset of A_G. Therefore, there is no guarantee that the abstracted action prescribed by the automaton in the information set will correspond (by a distinguishing function g) to some action in the information set. Hence, in order to speak about automata playing extensive-form games, only schemata consistent with the game should be considered.

Definition 19 (Game-tree-consistent schema)**.** *Let G = (N, H, Z, A_G, χ, ρ, σ, u, C, I) be an extensive-form game. A game-playing schema (M_i, f_i, g_i) of player i, where M_i = (Q, q_0, A_M, B, O, δ), is said to be consistent with G (G-consistent) if and only if for every possible extensive-form-game observation $\sigma \in \overline{\Sigma_i}$ available to player i in the information set I, for an abstracted observation $o = f_i(q, \sigma)$ that M_i perceives in state q it holds that $g_i(\delta(q, o), I')$ belongs to $A(I')$, where $seq_{\overline{\Sigma_i}}(I') = \sigma$. The state q is a state of M_i reached after the sequence of EFG observations $\sigma_0\sigma_1 \ldots, \sigma_m$ leading to I.*

Being G-consistent hence means that the schema is able to play the game correctly and it never makes an illegal move. However, the schema can be larger than necessary. Not all extensive-form-game observations are obtainable in a given state of the automaton and the domain of f_i can hence be smaller, which makes f_i a partial function. Furthermore, not all abstract observations are obtainable in a given state of the automaton and the automaton can have a smaller number of transitions. Finally, not all abstract actions are obtainable in a given information set, so that the domain of g_i can be smaller.

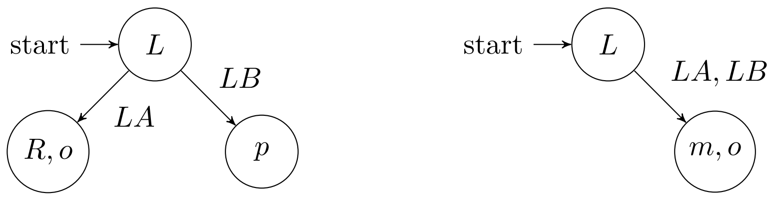

Figure 2.4: An example of an inconsistent game automaton (on the left) and a consistent game automaton (on the right) in a game depicted in Figure 2.3. The figure follows a standard denotation of a game automaton.

Based on the structure of the game tree, it is possible to identify unreachable abstracted observations and thus eliminate unreachable states and transitions from the structure of the game automaton. Similarly to the transition function of the automaton, the sequence of abstracted observations $o_0 \dots o_n$ generated from the initial state of the automaton after seeing a sequence of extensive-form-game observations $\sigma_0 \dots \sigma_n$ is denoted $f(q_0, \sigma_0 \dots \sigma_n)$.

Definition 20. *(Game-tree-unreachable states and transitions) Let G = (N, H, Z, A_G, χ, ρ, σ, u, C, I) be an extensive-form game and (M_i, f_i, g_i) be a G-consistent schema of player i consisting of a game automaton M_i = (Q, q_0, A_M, B, O, δ) and distinguishing functions f_i and g_i. A transition (q, o) is said to be unreachable if and only if for every possible sequence $\sigma_0 \dots \sigma_n \in \overline{\Sigma_i}^n$ of extensive-form-game observations generated from the game tree, such that $\sigma_0 \dots \sigma_n$ is consistent with actions prescribed by (M_i, f_i, g_i) it holds that $\delta(q_0, o_n) \neq q$ or $o_n \neq o$, for $o_n = f_i(q_0, \sigma_0 \dots \sigma_n)$. A state q' is unreachable if and only if all transitions (q, o) such that $\delta(q, o) = q'$ are unreachable. A schema (M_i, f_i, g_i) is called minimal if M_i has no unreachable states and transitions and f_i, g_i are undefined for unreachable inputs.*

Equivalently, a G-consistent schema has unreachable parts if there exists a strictly smaller sub-schema, which is also G-consistent. Assuming that any extensive-form game G can be encoded in time $O(|H|)$, all unreachable states and transitions can be eliminated in time polynomial in the size of the game G.

Proposition 1. *Given an extensive-form game G = (N, H, Z, A_G, χ, ρ, σ, u, C, I) and a G-consistent schema (M_i, f_i, g_i) of player i, there exists a subschema (M'_i, f'_i, g'_i) with no unreachable states and transitions, which can be effectively found in time $O(|G|)$.*

Proof. The algorithm will find the set Q of all reachable transitions of the automaton M_i. It proceeds by induction on the depth of the game tree of G. Initially, let Q be an empty set and R_0 be a set of triplets $(I, q_0, seq_{\overline{\Sigma_i}}(I))$, such that I is the shallowest information set of player i in G and $seq_{\overline{\Sigma_i}}(I)$ is an extensive-form-game observation leading to I. In the inductive step $j+1$, for each triplet $(I, q, \sigma) \in R_j$ add $(q, f_i(q, \sigma))$ to Q, and for each extensive-form-game observation σ' obtainable in I, such that σ' is consistent with action $g_i(\delta(q, f_i(q, \sigma)), I)$ prescribed by the schema, add $(inf(\sigma'), \delta(q, f_i(q, \sigma)), \sigma')$ to R_{j+1}. Because the tree is finite, the algorithm will stop and produce the set of all reachable transitions. Let $R = \bigcup_j R_j$, then $M \restriction Q, f_i \restriction R, g_i \restriction R$ is a G-consistent schema with partial distinguishing functions and no unreachable transitions and states of the automaton. $\square$

Recall the definition of automata playing repeated games. The game tree of any repeated game G is constructed by unfolding the underlining normal-form game (possibly infinitely) many times. From the perspective of automata playing extensive-form games, the distinguishing functions are therefore the same for all schemata of all players, defined as $f(q, a_i a_{-i}) = a_i$ and $g(q, I) = B(q)$. Whenever the behavioral function of the automaton of player i prescribes only actions of this player and the transition function is defined for every action of the opponent, the schema is G-consistent. Such simplification enables to speak directly about the automata, without the concept of game-playing schema. The automata in repeated games are hence a special case of automata in extensive-form games.

Game-tree-consistent schemata characterize feasible automata and distinguishing functions for every extensive-form game. Every schema corresponds to exactly one reduced pure strategy π.

Definition 21 (Pure strategy prescribed by (M_i, f_i, g_i))**.** *A G-consistent schema (M_i, f_i, g_i) of player i with no unreachable states and transitions prescribes a pure strategy $\pi \in \Pi_i$ in game G, if and only if for all actions $a \in A_G$ the schema prescribes it holds that $a \in \pi$.*

Moreover, for every pure strategy π it is possible to construct a canonical G- consistent schema (M_π, f_π, g_π), prescribing the strategy π. First, the canonical automaton is constructed. Recall that every set of nodes H of an extensive-form game G contains exactly one unique root node h_0.

Definition 22 (Canonical automaton)**.** *Let G = (N, H, Z, A_G, χ, ρ, σ, u, C, I) be an extensive-form game with a root node h_0, and a pure strategy $\pi \in \Pi_i$ of player i. The canonical automaton M_π = (Q, q_0, A_M, B, O, δ) of player i associated with π is constructed as follows. Let*

1. $Q := I_i \cup \{h_0\}$;
2. $q_0 := h_0$;
3. $A_M := A_G$;
4. $B(I) := \pi(I)$, *where* $I \in Q$ *and* $\pi(I)$ *is an action which* π *prescribes in I. If* $\pi(I)$ *is not defined,* $B(I) := \emptyset$;
5. $O := 2^{\overline{\Sigma_i}}$; *and*
6. $\delta(I_1, seq_{\overline{\Sigma_i}}(I_2)) = I_2$, *where* $I_1 \in Q$, $I_2 \in I_i$ *and* I_2 *immediately succeeds* I_1.

If the root h_0 of a game tree belongs to an information set of the same player as the automaton M_π, Q is equal to just I_i (since $q_0 \in I_i$). Otherwise, the initial state of the automaton does not correspond to any information set of the player and the player plays a "noop" action in this state. Then, he just waits for the first observation. Canonical automata are well-defined, since every pure strategy π in the game G corresponds to one G-consistent schema consisting of a canonical automaton M_π with trivial f_π, g_π being defined as

$$f_\pi(\_, \sigma) = \sigma, \qquad g_\pi(q, \_) = B(q). \tag{2.1}$$

However, the structure of the canonical schema can be often greater than necessary, not fully employing the capabilities of the automaton model with the distinguishing functions. For example, the automaton can eliminate the chains of repeating states with the same prescribed actions and the same transitions; or the distinguishing functions can be used to abstract the actions, creating directed cycles in the structure of the automaton. The reason for looking for smaller automata is that the fundamental property of the automaton is its behavior in a game, which should be preserved, rather than the structure.

Definition 23 (Behaviorally equivalent schemata). *Let $G = (N, H, Z, A_G, \chi, \rho, \sigma, u, C, I)$ be an extensive-form game, and (M_i^1, f_i^1, g_i^1) and (M_i^2, f_i^2, g_i^2) be two schemata of player i, where $M_i^1 = (Q_1, q_0^1, A_1, B_1, O_1, \delta_1)$ and $M_i^2 = (Q_2, q_0^2, A_2, B_2, O_2, \delta_2)$. (M_i^1, f_i^1, g_i^1) and (M_i^2, f_i^2, g_i^2) are said to be behaviorally equivalent if and only if they prescribe the same pure strategy.*

A special case of behaviorally equivalent automata are the isomorphic ones.

Definition 24 (Isomorphic schemata). *Let $G = (N, H, Z, A_G, \chi, \rho, \sigma, u, C, I)$ be an extensive-form game, and (M_i^1, f_i^1, g_i^1) and (M_i^2, f_i^2, g_i^2) be two schemata of player i, where $M_i^1 = (Q_1, q_0^1, A_1, B_1, O_1, \delta_1)$ and $M_i^2 = (Q_2, q_0^2, A_2, B_2, O_2, \delta_2)$. (M_i^1, f_i^1, g_i^1) and (M_i^2, f_i^2, g_i^2) are said to be isomorphic if and only if there exists a one-to-one onto mapping $h : Q_1 \rightarrow Q_2$, which preserves behavior, transitions and distinguishing functions.*

The schemata are isomorphic if they differ only in the labels of their components, but not in the structures of the schemata themselves.

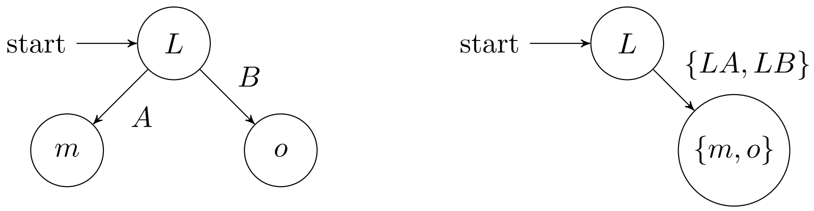

Figure 2.5: An example of two automata of behaviorally equivalent schemata implementing a pure strategy (L, m, o). Note that the schemata are not isomorphic. The figure follows a standard denotation of a game automaton.

Example 14. Consider the two automata depicted in Figure 2.5. These automata both prescribe a pure strategy (L, m, o) in a game G, which game tree is depicted in Figure 2.3. The automaton of the left is a canonical automaton for strategy (L, m, o). For the automaton on the right, let the corresponding distinguishing functions be defined as $f(q_0^R, LA) = f(q_0^R, LB) = \{LA, LB\}$ and $g(q_0^R, I_1) = L;\ g(q_1^R, I_2) = m;\ g(q_1^R, I_2) = o$, using an abstracted observation $\{LA, LB\}$ in the upper state q_0^R (corresponding to an upper information set I_1) and abstracted action $\{m, o\}$ in the lower state q_1^R (corresponding to the lower information sets I_2, I_3). Therefore, both automata are G-consistent and mutually behaviorally equivalent, but not isomorphic, since they do not have the same number of states.

Behavioral equivalence is an equivalence relation on the set of G-consistent schemata. All schemata belonging to the same class implements the same pure strategy, even though the structure of the automata and the distinguishing functions can differ. Similarly to the case of repeated games, players can evaluate the structural complexity of schemata using different measures of strategic complexity.

2.3 Strategic complexity measures

Given an extensive-form game, behavioral equivalence creates the equivalence classes on the set of all consistent schemata, such that every class prescribes the same pure strategy. The player can evaluate the schemata belonging to the same class in order to find the schema which realizes a given pure strategy the best; or compare the schemata across different behavioral classes to determine the differences between complexities of pure strategies.

Definition 25 (Measure of schemata complexity)**.** *Function $\mu : (M, f, g) \to \mathbb{R}^m$ is a measure of schemata complexity in an extensive-form game G if the i^{th} component of the function is a linear combination of the following real-valued functions:*

1. *a structural complexity measure associated with the automaton $M = (Q, q_0, A_M, B, O, \delta)$'s size, expressed as either $\mu_Q = |Q|$ or $\mu_\delta = |\delta|$; and*
2. *a computational complexity measure associated with the complexity of distinguishing functions f, g : either space (S) or time (T) complexity, which can be best-case (μ_S^b or μ_T^b), worst-case (μ_S^w or μ_T^w) or average-case (μ_S^a or μ_T^a) in a fixed finite game G;*

and for each component there is an associated finite upper bound $\overline{\mu^i}$.

In other words, a player's measure of complexity of a given schema is an m- dimensional real-valued vector, such that each coordinate represents a numeric evaluation of a general schemata property the player finds important. Schemata complexity focuses on both the structure of the automaton, which is either a number of states or a number of transitions – an approach recommended in [53]; and the complexity of distinguishing functions. For now, assume that the complexity is analyzed in one of the alternative models of computation, in which the complexity is uniquely determined and linear speed-ups are not possible. Such models have already been specified [54, 55, 56]. The measure of schemata complexity in extensive-form games generalizes the measure of automata complexity as introduced in repeated games. Note that every measure of automata complexity mentioned in section 1.1.3 is a special case of the schemata complexity. In case the complexity of a schema exceeds a predefined upper bound it is not considered a feasible schema: the thesis only assumes that these is always such an upper bound. Similarly to repeated games, every measure of schemata complexity μ generates a natural partial order $\preceq_\mu$ on the set of all feasible schemata.

Definition 26 (Partial order $\preceq_\mu$ generated by μ)**.** *Let μ be a measure of schemata complexity of player i in an extensive-form game G. A partial order $\preceq_\mu$ is a partial order generated by μ if and only if for every pair of schemata (M_i^1, f_i^1, g_i^1) and (M_i^2, f_i^2, g_i^2) of player i it holds that $(M_i^1, f_i^1, g_i^1) \preceq_\mu (M_i^2, f_i^2, g_i^2)$ if and only if for every coordinate j of μ*

$$\mu^j(M_i^1, f_i^1, g_i^1) \leq \mu^j(M_i^2, f_i^2, g_i^2). \tag{2.2}$$

As in every partial order, also in $\preceq_\mu$ there exist minimal schemata which are the minimal elements of $\preceq_\mu$. Such minimal elements are important especially for schemata belonging to the same class B_π of behaviorally equivalent schemata. These schemata correspond to the minimal realizations of a given pure strategy π, which are incomparable with respect to μ. In case the schema is not a minimal element of $\preceq_\mu$ on the set B_π, there is a schema with a reduced size which preserves the behavior.

Definition 27 (Reducible schema)**.** *Let μ be a measure of schemata complexity in an extensive-form game G. A game-playing schema (M_i^1, f_i^1, g_i^1) of player i is said to be reducible if and only if there exists a schema (M_i^2, f_i^2, g_i^2), such that (M_i^1, f_i^1, g_i^1) is behaviorally equivalent to (M_i^2, f_i^2, g_i^2) and $(M_i^2, f_i^2, g_i^2) \preceq_\mu (M_i^1, f_i^1, g_i^1)$.*

However, the schema which is a minimal element of $\preceq_\mu$ on B_π does not have to be a minimum of this partial order. For the minimum to always exist, the partial order has to be completed.

For the individual measures, the definition of reducibility can be made more precise. From now on consider the reducibility for the state measure of schemata complexity μ_Q with preserved distinguishing functions. The definition of reducibility therefore reduces to the structure of automaton itself.

Observation 1 (Reducible schema with respect to μ_Q)**.** *Let (M_i, f_i, g_i) be a game-playing schema of player i in an extensive-form game G, where $M_i = (Q,\ B,\ O, \delta,\ q_0)$. (M_i, f_i, g_i) is reducible if and only if there exists a partition $Q = (Q_1, Q_2, \ldots, Q_k)$ on the states of the automaton M_i such that:*

1. *the decomposition is non-trivial, hence $k < |Q|$; and*
2. *the behavior of the automaton is preserved, which means that for all $q, q' \in Q$*
   (a) *B(q) = B(q'); and*
   (b) *for all $\sigma \in O$ exists t, such that $\delta(q, \sigma), \delta(q', \sigma) \in Q_t$.*

Note that a game automaton is reducible with respect to μ_Q only if it has some unnecessary states. These unnecessary states can be found using a method similar to minimization of a size of the finite automata [57]. The idea is based on the Myhill-Nerode characterization of regular languages. Given a language L, and a pair of strings a and b, a distinguishing extension is a string x such that exactly one of the two strings ax and bx belongs to L. The characterization defines a relation R_L on strings so that aR_Lb if and only if there is no distinguishing extension for a and b. R_L is hence an equivalence relation on strings, and thus it divides the set of all strings into equivalence classes. The Myhill-Nerode theorem [57] states that L is regular if and only if R_L has a finite number of equivalence classes. Moreover, the number of states of the smallest deterministic finite automaton recognizing L is equal to the number of equivalence classes of R_L. Each state of such smallest automaton corresponds to exactly one equivalence class. Reducing the size of an automaton recognizing L is hence done by merging its states (viewed as sets of R_L-equivalent strings), whenever two states correspond to the same set of equivalent strings.

To introduce such technique for game-playing schemata, it is necessary to redefine the equivalence relation in a sense of transducers [58]. While the algorithm for merging states of a finite state automata used original relation R_L for checking whether any future accepting input would also guarantee to end in an accepting state, the relation for game automata has to ensure that future observations will lead to prescribing the same actions.

Definition 28 (Distinguishing extension)**.** *Given a game automaton M, and two observations o_1 and o_2, a distinguishing extension is an observation o' such that $B(\delta(q_0, o_1o')) = u$ and $B(\delta(q_0, o_2o')) = v$ with $u \neq v \in A_M$.*

Therefore, the distinguishing extension is an observation such that the behavior of the game automaton differs for o' depending on whether it receives o_1 or o_2. Again, the corresponding relation R_M is an equivalence relation, dividing all finite sequences of elements from O into equivalence classes. In case of a game automaton, these classes will again correspond to the states of the non-reducible automaton.

Proposition 2. *For every game-playing schema (M, f, g), where $M = (Q,\ B,\ O,\ \delta,\ q_0)$, there exists a behaviorally equivalent schema (M', f', g'), such that M' is a μ_Q-non-reducible automaton. The schema (M', f', g') can be found in time $O(|O||Q|^2)$.*

Proof. Let $\overline{F}$ be a set of all possible functions $\overline{f} : \overline{\Sigma_i} \to O$. First, the initial partition $\mathcal{P}$ of Q into classes of states $S_{o,\overline{f}}$ is defined as follows.

$$\forall o \in O\ \forall \overline{f} \in \overline{F} : S_{o,\overline{f}} = \{q \in Q \mid B(q) = o,\ f \restriction q \downarrow = \overline{f}\} \tag{2.3}$$

Second, the classes in $\mathcal{P}$ are split according to the following algorithm.

repeat
  for $S \in \mathcal{P}$ **do**
    for $o \in O$ **do**
      if $\forall S' \in P \quad \exists q \in S \quad \delta(q, o) \notin S'$ **then**
        split S into subsets S_i such that
        for each subset S_i, there is a different class $S' \in \mathcal{P}$ such that
        $\forall q \in S_i,\ \delta(q, o) \in S'$
        (the subsets S_i replace S in $\mathcal{P}$)
      end if
    end for
  end for
until no changes

When there is no class left that needs to be split, the remaining classes of states will form the states of the μ_Q-non-reducible game automaton. By construction, all states in a class prescribe the same action which is the behavior of the class. Similarly, for any observation $o \in O$, all states in a class go to some state in the same other class, which defines the transition function for the μ_Q-non-reducible game automaton.

The initial partition can be generated in time $|Q|(|O| + 1)$, because for each state all possible outputs of f are considered. The main loop is executed at most $|Q|$ times, because in each iteration at least one class of states must be split, and each class contains at least one state. Each iteration of the loop examines each state a number of times equal to the number of input observations. Hence the complexity of the algorithm is $O(|O||Q|^2)$. □

The distinguishing functions of the μ_Q-non-reducible schema are set to $f \upharpoonright Q$ and $g \upharpoonright Q$, where Q is a set of states of the automaton M'. Finding μ_Q-non-reducible schema can be done for every schema belonging to a class B_π for a given π. Such schemata form a subset of minimal elements with respect to the partial order $\preceq_{\mu_Q}$ on B_π. The measure is one-dimensional and therefore a linear order. Consequently, it always has a minimum. However, this minimum can be obtained in several distinct schemata with the same value of the schemata complexity measure. For the state measure, it holds that there can not be two non-isomorphic minimal behaviorally equivalent schemata with the same distinguishing functions. The reasoning is similar to the case of DFA [59].

Proposition 3. *Every minimal μ_Q-non-reducible schema is unique (up to isomorphism).*

Proof. Assume there are two minimal non-reducible game schemata with automata M_1 and M_2 and with the same distinguishing functions which are behaviorally equivalent, but $|Q_1| < |Q_2|$. Run the algorithm from the proof of Proposition 2 on the states of M_1 and M_2 together, as if they were one game automaton. The initial states of M_1 and M_2 have to be indistinguishable, because the automata are behaviorally equivalent. If states $q_1 \in Q_1$ and $q_2 \in Q_2$ are indistinguishable, so are all their successors, otherwise it is possible to distinguish q_1 and q_2. Both automata are minimal and therefore every state of M_1 is indistinguishable from at least one state of M_2 and vice versa. But M_1 is smaller than M_2, so there have to be two states of M_2 in the same equivalence class, which contradicts the assumption of non-reducibility. Consequently, because of the minimality, the automata has to be isomorphic. □

Therefore, up to isomorphism, the minimum of $\preceq_{\mu_Q}$ on B_π is always one unique minimal schema for any fixed sets of abstracted actions and abstracted observations.

Chapter 3

Properties of Schemata Equilibria

This chapter introduces solution concepts in games with schemata strategies. It formally defines the extensions of Nash equilibrium and analyzes their computational complexity. The rest of the chapter considers properties of efficiently representable strategies and their effect on computing specific EFCE.

The previous chapter explained that the suitable analogy of strategies in games with schemata strategies are minimal G-consistent schemata. The reason is they represent the minimal strategies which are able to correctly play a given game. Player's preference over this set of schemata is represented by his schemata complexity measure μ, which evaluates a non-reducibility of a given schema. Therefore, minimal G-consistent non-reducible schemata are essential for analyzing equilibria.

Definition 29 (Pure schemata strategy)**.** *Let G = (N, H, Z, A_G, χ, ρ, σ u, C, I) be an extensive-form game and $\mu = \mu_i$ be a schemata complexity measure of player i. A set of pure schemata strategies is a set of minimal G-consistent non-reducible schemata with respect to μ, denoted as S_i^μ.*

Because all schemata in S_i^μ are minimal, the set S_i^μ is always finite. The reason is that the partitioning of actions into abstracted actions is finite, hence the number of possible states in every minimal automaton is bounded and so is the number of possible transitions. Because both the automaton and the game tree are finite, the number of possible distinguishing functions is also finite.

With a non-trivial schemata complexity measure, which has multiple non-isomorphic minimal elements in at least one class of behaviorally equivalent schemata, the set of pure schemata strategies is richer than the set of pure strategies in an extensive-form game. However, this makes sense from the perspective of the player, as behaviorally equivalent minimal schemata can be seen as different realizations of the same strategy, e.g. with the different ratio of time and space complexity.

Similarly to the traditional game theory, in scenarios where a player faces uncertainty, it is advisable to randomize his strategy.

Definition 30 (Mixed schemata strategy)**.** *Let G = (N, H, Z, A_G, χ, ρ, σ u, C, I) be an extensive-form game, μ be a schemata complexity measure of player i and S_i^μ his set of pure schemata strategies. The set of player's mixed schemata strategies is a set Φ_i^μ of all possible probability distributions over S_i^μ.*

In a mixed strategy $\phi_i \in \Phi_i^\mu$ of player i, by $\phi_i(M, f, g)$ is denoted the probability of playing the game G according to schema (M, f, g). The progression of any game is fully determined by specifying a strategy for each player.

Definition 31 (Schemata profile)**.** *Let G = (N, H, Z, A_G, χ, ρ, σ u, C, I) be an extensive-form game and $(\mu_1, \mu_2, \ldots, \mu_n)$ be a tuple of schemata complexity measures* *for each player. A schemata profile $\phi \in \Phi_1^{\mu_1} \times \cdots \times \Phi_n^{\mu_n}$ is a tuple of one mixed schemata strategy for each player.*

A partial profile ϕ_{-i} denotes a tuple of schemata strategies played by the opponents of player i. In case the player i knows these strategies, he is able to reason about which schemata strategy he should take. Recall, that an equilibrium in traditional game theory is a specific profile such that no player has an intention to change his strategy.

In order to analyze the structure of equilibria in schemata, it is necessary to define three essential concepts of strategic interaction. First, given a partial schemata profile, what is a best-response schemata strategy. Second, under which assumptions does a Nash equilibrium always exist. And finally third, how to find a Nash equilibrium.

To find a best-response schemata strategy, a player has to be able to compare the utilities he expects to receive from any two different strategies against a fixed partial schemata strategy profile of his opponents. To evaluate an expected utility, for every leaf z of the game tree the player determines whether it can be reached by a given strategy profile and he calculates the probability of reaching it. The probability of reaching z is a product of probabilities of two independent events: that in the chance nodes the actions are taken so that they are on the path from the root state to z and that the schemata profile is chosen so that z can be reached.

From the definition of extensive-form games, the probabilities of individual actions of chance are independent. Therefore, the probability of reaching z due to chance is simply a product of probabilities of all random actions on the path from the root state to z. Because all players choose their strategies at the beginning of a game and each strategy defines which actions a player will take in every situation in the game, the probability of reaching z due to a strategy of player i is independent on the strategies of other players. The probability of reaching z due to individual strategies of the players is hence also a product of probabilities of reaching z according to a strategy of each player.

In the traditional game theory, for every z there exists a set of reduced strategies Π_i^z per each player i, such that each strategy $\pi_i^z \in \Pi_i^z$ enables to reach z in the game. On the other hand, in games with schemata, there might be multiple minimal G-consistent non-reducible schemata prescribing π_i^z. Assuming all other players take actions such that z can be reached, the probability of reaching z is the sum of probabilities of playing game G according to minimal G-consistent non-reducible schemata prescribing π_i^z.

Observation 2 (Expected utility of a schemata profile)**.** *The expected utility of player i with respect to a schemata profile ϕ and a tuple of schemata complexity measures $(\mu_1, \mu_2, \ldots, \mu_n)$ in an extensive-form game $G = (N, H, Z, A_G, \chi, \rho, \sigma\ u, C, I)$ is equal to*

$$u_i(\phi) = \sum_{z \in Z} u_i(z) C(z) \prod_{j \in [N]} \sum_{\substack{(M,f,g) \in S_j^{\mu_j} \\ z \in Z_{(M,f,g)}}} \phi_j(M, f, g), \tag{3.1}$$

where $Z_{(M,f,g)} \subseteq Z$ is a set of leaves reachable by schema (M, f, g).

Given a partial profile ϕ_{-i}, the player i can use an expected utility to identify optimal schemata strategy.

Definition 32 (Best-response schemata strategy)**.** *Let $G = (N, H, Z, A_G, \chi, \rho, \sigma\ u, C, I)$ be an extensive-form game and ϕ_{-i} be a partial profile of schemata strategies. $\phi_i \in \Phi_i$ is player i's best response to ϕ_{-i} if and only if for all $\phi_i' \in \Phi_i$ it holds that*

$$u_i(\phi_i, \phi_{-i}) \geq u_i(\phi_i', \phi_{-i}). \tag{3.2}$$

A Nash equilibrium in schemata strategies is a situation in which no player profits from changing his schemata strategy.

Definition 33 (Nash equilibrium in schemata strategies)**.** *Let G = (N, H, Z, A_G, χ, ρ, σ u, C, I) be an extensive-form game and $\phi^{Nash} = (\phi_1, ..., \phi_n)$ be a schemata profile. ϕ^{Nash} is a Nash equilibrium if and only if for each player $i \in N$ it holds that ϕ_i is a best response to ϕ_{-i}.*

Similarly to the games with ordinary strategies, Nash equilibrium always exists even if the players are obligated to use schemata. This holds even if the set of schemata strategies is much richer than the set of pure strategies.

Proposition 4 (Existence of Nash equilibrium in schemata strategies)**.** *Every multi-player, perfect-recall extensive-form game has a schemata profile ϕ^{Nash} such that ϕ^{Nash} is a Nash equilibrium in schemata strategies.*

Proof. Let G = (N, H, Z, A_G, χ, ρ, σ u, C, I) be an extensive-form game and $(\mu_1, \mu_2, \ldots, \mu_n)$ be a tuple of schemata complexity measures for each player. A function NF is a reduction from G to a normal-form game $NF(G) = (N,\ A_{NF},\ u_{NF})$, where

- N is the same set of players as in G;
- $A_{NF} = A_1 \times \cdots \times A_n$, where $A_i = S_i^{\mu_i}$ is a set of actions for player i; and
- $u_{NF} = (u_1, \ldots, u_n)$ where $u_i(a) = \sum_{z \in \bigcap_{(M,f,g)\in a} Z_{(M,f,g)}} u_i(z)C(z)$ is a real-valued utility function for player i.

Note that by definition A_{NF} is a set of all pure schemata profiles and u_i corresponds to an expected utility of player i of a given pure schemata profile. The definition of u_i is hence consistent with equation 3.1 precisely on the set of pure schemata profiles. Recall that in traditional game theory, the expected utility of player i in $NF(G)$ with respect to a mixed strategy profile $\delta \in \Delta$, where Δ is a set of all mixed strategy profiles, is equal to

$$u_i(\delta) = \sum_{a \in A_{NF}} u_i(a) \prod_{j \in [N]} \delta_j(a_j) \tag{3.3}$$

which by the definition of reduction NF is equal to

$$u_i(\delta) = \sum_{a \in A_{NF}} \Big(\sum_{z \in \bigcap_{(M,f,g)\in a} Z_{(M,f,g)}} u_i(z)C(z) \Big) \prod_{j \in [N]} \delta_j(a_j). \tag{3.4}$$

Because calculating a sum over reachable leaves for each pure schemata profile is the same as calculating a sum over schemata profiles leading to a given leaf for every leaf, the first two sums can be rearranged into

$$u_i(\delta) = \sum_{z \in Z} u_i(z)C(z) \prod_{j \in [N]} \sum_{\substack{a \in A_{NF} \\ z \in \bigcap_{(M,f,g)\in a} Z_{(M,f,g)}}} \delta_j(a_j), \tag{3.5}$$

which is the same as the expected utility of a schemata profile. The equilibrium in $NF(G)$ is therefore the equilibrium in G. By Nash's theorem, an equilibrium in $NF(G)$ always exists and hence there is always an equilibrium in schemata strategies in G. □

In this definition of Nash equilibrium, the player uses his measure of schemata complexity to select minimal realizations of the pure strategies. However, the best response is then based solely on the expected utility of a given profile and does not take into account the exact complexity of individual strategies in the profile. There are two reasons why this approach of extending Nash equilibrium into schemata strategies is rational.

First, the complexity of a schema is a property independent of the choices of strategies of other players, while the expected utility is always related to the complete schemata profile. Therefore, it can be analyzed separately. The player can introduce further restrictions on the set of minimal G-consistent non-reducible schemata and the existence of a Nash equilibrium will be preserved, because the fact that all schemata in S_i^{μ} have this property has never been used in the proof of the existence of a Nash equilibrium. For example, a player can choose more strict upper bounds on the complexity of schemata and reject all schemata with complexity exceeding this upper bound. In case there exists at least one minimal G-consistent non-reducible schema satisfying this restriction, a Nash equilibrium in this game exists, even though it does not have to be the same as in the original unrestricted game. The bounds on the complexity of schemata introduce bounded rationality into the model as a representation of player's limited memory and computational abilities.

Second, even though different realizations of the same pure strategy guarantee the same expected utility against fixed partial profile of opponents' strategies, the complexities of the realizations are incomparable. Every player i can evaluate his expected complexity of a schemata strategy ϕ_i as

$$\mu_i(\phi_i) = \sum_{z \in Z} \sum_{\substack{(M,f,g) \in S_i^{\mu_i} \\ z \in Z_{(M,f,g)}}} \phi_i(M, f, g)\mu_i(M, f, g). \tag{3.6}$$

By playing a non-trivial mixed strategy over the different realizations of the same pure strategy, the player is able to guarantee an expected complexity which might be better than the complexities of individual schemata.

On the other hand, it is also possible to include the intention to balance expected utility and schemata complexity directly into the definition of the best response.

Definition 34 (Utility/complexity balancing function)**.** *Let $G =$ (N, H, Z, A_G, χ, ρ, σ u, C, I) be an extensive-form game and $\mu = \mu_i : (M, f, g) \to \mathbb{R}^n$ be a schemata complexity measure for player i. A real-valued function $\psi_i^{\mu} : \mathbb{R}^{n+1} \to \mathbb{R}$ is a utility/complexity balancing function of player i if and only if ψ_i^{μ} is a player-wise continuous linear function, i.e. for a fixed ϕ_{-j} and $i \neq j$*

$$\sum_{s \in S_j^{\mu_j}} \phi_j(s)\psi_i^{\mu}(\mu_i(\phi_i), u_i(s, \phi_{-j})) = \psi_i^{\mu}(\mu_i(\phi_i), u_i(\phi_j, \phi_{-j})) \tag{3.7}$$

and especially if $i = j$

$$\sum_{s \in S_i^{\mu_i}} \phi_i(s)\psi_i^{\mu}(\mu_i(s), u_i(s, \phi_{-i})) = \psi_i^{\mu}(\mu_i(\phi_i), u_i(\phi_i, \phi_{-i})). \tag{3.8}$$

Utility/complexity balancing functions might also preserve the preference with respect to schemata complexity measures, but not necessarily with respect to expected utility. This can be captured by an additional condition: for every partial profile of schemata strategies ϕ_{-i} and schemata strategies ϕ_i and ϕ_i' it should hold that

$$\mu_i(\phi_i) \prec \mu_i(\phi_i') \to \psi_i^{\mu}(\mu_i(\phi_i), u_i(\phi_i, \phi_{-i})) \geq \psi_i^{\mu}(\mu_i(\phi_i'), u_i(\phi_i', \phi_{-i})) \tag{3.9}$$

For better readability, the expression $\psi_i^{\mu}(\mu_i(\phi_i), u_i(\phi_i, \phi_{-i}))$ is abbreviated as $\psi_i^{\mu}(\phi)$. Note that balancing functions generalize the functions which alter the utilities only in the leaves of the game tree, such that they take into account the complexity of schemata prescribing a pure strategy leading to this leaf. Using utility/complexity balancing functions, the best-response strategy can be defined.

Definition 35 (Extended best-response schemata strategy)**.** *Let G = (N, H, Z, A_G, χ, ρ, σ u, C, I) be an extensive-form game, ψ_i^μ be a utility/complexity balancing function of player i and ϕ_{-i} be a partial profile of schemata strategies. ϕ_i is player i's best response to ϕ_{-i} if and only if for all $\phi'_i \in \Phi_i$ it holds that*

$$\psi_i^\mu(\phi_i, \phi_{-i}) \geq \psi_i^\mu(\phi'_i, \phi_{-i}). \tag{3.10}$$

A Nash equilibrium which balances the complexity of strategies and the expected utility hence uses this definition of best response.

Definition 36 (Balanced Nash equilibrium in schemata strategies)**.** *Let G = (N, H, Z, A_G, χ, ρ, σ u, C, I) be an extensive-form game, $(\psi_1^{\mu_1}, \psi_2^{\mu_2}, \ldots, \psi_n^{\mu_n})$ be a tuple of one utility/complexity balancing functions per each player and $\phi^{Nash} = (\phi_1, ..., \phi_n)$ be a schemata profile. ϕ^{Nash} is a Nash equilibrium if and only if for each player $i \in N$ it holds that ϕ_i is an extended best response to ϕ_{-i}.*

Also in games with extended best responses, a Nash equilibrium always exists.

Proposition 5 (Existence of balanced Nash equilibrium in schemata strategies)**.** *Every multi-player, perfect-recall extensive-form game has a schemata profile ϕ^{Nash} such that ϕ^{Nash} is a balanced Nash equilibrium in schemata strategies, according to an extended best response.*

Proof. Similarly to the previous proof of the existence of Nash equilibrium in schemata strategies, an equivalent normal form is constructed. It is shown that the best response in the normal form is the same as the extended best response with utility/complexity balancing functions. Therefore, a Nash equilibrium in the normal-form game is a Nash equilibrium in the original extensive-form game. Because in every normal-form game there is always an equilibrium, there exists an equilibrium also in the equivalent extensive-form form.

Let G = (N, H, Z, A_G, χ, ρ, σ u, C, I) be an extensive-form game and $(\psi_1^{\mu_1}, \psi_2^{\mu_2}, \ldots, \psi_n^{\mu_n})$ be a tuple of utility/complexity balancing functions for each player. A function NF_e is a reduction of G to a normal-form game $NF_e(G) = (N, A_{NF}, u_{NF})$, where

- N is the same set of players as in G;
- $A_{NF_e} = A_1 \times \cdots \times A_n$, where $A_i = S_i^{\mu_i}$ is a set of actions for player i; and
- $u_{NF_e} = (u_1, \ldots, u_n)$ where $u_i(a) = \psi_i^{\mu_i}(\mu_i(a_i), \overline{u_i}(a))$ and $\overline{u_i}(a) = \sum_{z \in \bigcap_{(M,f,g) \in a} Z_{(M,f,g)}} u_i(z)C(z)$ is a real-valued utility function for player i.

The definition of the utility function is well-defined, because a_i is a schema a $\mu_i(a_i)$ is therefore properly defined. According to the traditional game theory, the expected utility of player i in $NF_e(G)$ with respect to a mixed strategy profile $\delta \in \Delta$, where Δ is a set of all mixed strategy profiles, is equal to

$$u_i(\delta) = \sum_{a \in A_{NF_e}} u_i(a) \prod_{j \in [N]} \delta_j(a_j) \tag{3.11}$$

which by the definition of reduction NF_e is equal to

$$u_i(\delta) = \sum_{a \in A_{NF}} \psi_i^{\mu_i}(\mu_i(a_i), \overline{u_i}(a)) \prod_{j \in [N]} \delta_j(a_j). \tag{3.12}$$

This is equivalent to

$$u_i(\delta) = \sum_{a_1 \in A_1} \delta_1(a_1) \sum_{a_2 \in A_2} \delta_2(a_2) \cdots \sum_{a_n \in A_n} \delta_n(a_n) \psi_i^{\mu_i}(\mu_i(a_i), \overline{u_i}(a_1, a_2, \ldots, a_n)). \tag{3.13}$$

Because $A_i = S_i^{\mu_i}$, δ is a mixed schemata strategy and $\overline{u_i}$ is an expected utility of a schemata profile in G. Since $\psi_i^{\mu_i}$ is a player-wise continuous linear function, the individual sums in 3.13 can be backwardly eliminated according to the properties of $\psi_i^{\mu_i}$. In the j^{th} step of the elimination one of the following equalities holds:

1. $i < j$ and according to property 3.7

$$\begin{aligned}\sum_{a_j \in A_j} \delta_j(a_j)\psi_i^{\mu_i}(\mu_i(a_i), \overline{u_i}(a_1, a_2, \ldots, a_j, \delta_{j+1} \ldots, \delta_n)) = \\ \psi_i^{\mu_i}(\mu_i(a_i), \overline{u_i}(a_1, a_2, \ldots, a_{j-1}, \delta_j, \ldots, \delta_n))\end{aligned} \tag{3.14}$$

2. $i = j$ and according to property 3.8

$$\begin{aligned}\sum_{a_i \in A_i} \delta_i(a_i)\psi_i^{\mu_i}(\mu_i(a_i), \overline{u_i}(a_1, a_2, \ldots, a_i, \delta_{i+1} \ldots, \delta_n)) = \\ \psi_i^{\mu_i}(\mu_i(\delta_i), \overline{u_i}(a_1, a_2, \ldots, a_{i-1}, \delta_i, \ldots, \delta_n))\end{aligned} \tag{3.15}$$

3. $i > j$ and according to property 3.7

$$\begin{aligned}\sum_{a_j \in A_j} \delta_j(a_j)\psi_i^{\mu_i}(\mu_i(\delta_i), \overline{u_i}(a_1, a_2, \ldots, a_j, \delta_{j+1} \ldots, \delta_n)) = \\ \psi_i^{\mu_i}(\mu_i(\delta_i), \overline{u_i}(a_1, a_2, \ldots, a_{j-1}, \delta_j, \ldots, \delta_n))\end{aligned} \tag{3.16}$$

After all sums ale eliminated, the expected utility is expressed as

$$u_i(\delta) = \psi_i^{\mu_i}(\mu_i(\delta_i), \overline{u_i}(\delta_1, \delta_2, \ldots, \delta_n)). \tag{3.17}$$

Recall that a mixed strategy $\delta_i \in \Delta_i$ is a best response to a partial profile δ_{-i} if and only if for all strategies $\delta_i' \in \Delta_i$

$$u_i(\delta_i, \delta_{-i}) \geq u_i(\delta_i', \delta_{-i}). \tag{3.18}$$

By the equivalence 3.17 the best-response condition is equivalent to

$$\psi_i^{\mu_i}(\mu_i(\delta_i), \overline{u_i}(\delta_i, \delta_{-i})) \geq \psi_i^{\mu_i}(\mu_i(\delta_i'), \overline{u_i}(\delta_i', \delta_{-i})) \tag{3.19}$$

and therefore the best response in the reduced game is also the best response in the original game. □

Since models of schemata as strategies extend the traditional game theory, computing Nash equilibrium in schemata strategies in general class of extensive-form games is at least as hard as computing Nash equilibrium in the original game.

Observation 3. *For an imperfect-information perfect-recall extensive-form game with at least two players, the problems of finding Nash equilibrium in schemata strategies and balanced Nash equilibrium in schemata strategies are PPAD-hard.*

Proof. The proof is based on the reductions from an arbitrary imperfect-information perfect-recall extensive-form game G to a game in which the (balanced) Nash equilibrium in schemata strategies is computed.

For the Nash equilibrium in schemata strategies, let every player select a schemata complexity measure inducing a linear order (e.g. the one-dimensional complexity measure μ_Q which considers only the number of states of the automaton) and the upper bound on the complexity such that for every pure strategy there is at least one schema which prescribes it. Since the complexity measures are linear, every pure strategy has exactly one minimal complexity (which is a minimum element of the induced linear order) and a schema of this complexity which represents it. The mixed schemata strategies of the Nash equilibrium in this game therefore directly corresponds to the mixed strategies of the Nash equilibrium in the original game.

In case of the balanced Nash equilibrium in schemata strategies, let all players select the schemata complexity measures inducing linear orders again and let $\psi_i(\mu_i(\phi_i), u_i(\phi)) = u_i(\phi)$. The expected utility is a continuous linear function and therefore it satisfies conditions 3.7 and 3.8. Similarly to the previous case, also in this game the mixed schemata strategies of the balanced Nash equilibrium directly corresponds to the mixed strategies of the Nash equilibrium in the original game. Moreover, if the players select a state complexity measure μ_Q that minimizes the size of the automaton in the schema, the expected utility satisfies also condition 3.9. Because the complexity of the distinguishing functions is not restricted, for every player i and every pure strategy $\pi_i \in \Pi_i$ it is possible to define one abstracted action $A = \pi_i$. Because the game tree is finite, the distinguishing functions are able to correctly verify to which action in the current information set in the game tree the abstracted action corresponds. All minimal schemata has a trivial complexity $\mu_Q(M, f, g) = 1$. Also in this case the balanced Nash equilibrium corresponds to the Nash equilibrium in the original game. □

Note that because every Nash equilibrium is a correlated equilibrium, the existence of Nash equilibrium also implies the existence of correlated equilibrium. Similarly, because every correlated equilibrium is an extensive-form correlated equilibrium, the existence of extensive-form correlated equilibrium is also guaranteed.

3.1 Efficiently representable strategies

In the proof of the PPAD-hardness, the reduction to the balanced Nash equilibrium with satisfied condition 3.9 used the fact that the extensive-form games are given as finite trees. Without any bounds on the complexity of the structure of the automaton or computational complexity of the distinguishing functions, the whole game tree of the game can be coded up into either the distinguishing functions (as it was used in the reduction) or to the automaton.

Observation 4. *Let G be an extensive-form game. Then the automaton of every μ_Q-non-reducible schema has only one state and one transition and the complexity of the distinguishing functions of every μ_S-minimal schema is constant.*

Proof. The μ_Q measure does not take into account the complexity of the distinguishing functions. Therefore, it is possible to encode the whole pure strategy the schema should prescribe solely into the memory of the g function of the schema. Let $\pi_i \in \Pi_i$ be a pure strategy the player i would like to play in the game G. The schema (M_i, f_i, g_i) is defined so that $M_i = (Q, q_0, A, B, O, \delta)$, where

- $Q = \{q_0\}$ is a one-state set of states;
- $A = \{\pi_i\}$ is the only abstracted action the automaton prescribes;
- $B(Q) = B(q_0) \to \pi_i$ is defined in the automaton's only state;
- $O = \{\overline{\Sigma_i}\}$ is a set of all extensive-form-game observations, which is the only abstracted observation the automaton is able to perceive; and
- $\delta(Q, O) = \delta(q_0, \overline{\Sigma_i}) \to q_0$ is defined for the only state and only observation which is available to the automaton.

The f_i distinguishing function is then defined as

$$f_i(\_, \_) = \overline{\Sigma_i}, \tag{3.20}$$

which means that for any state of the automaton and any extensive-form-game observation provided by the game the f_i function returns only the single observation available to the automaton. The time and space complexity of the f_i function are hence constant. Let the g_i function be defined as

$$g_i(\_, I) = lookup(I), \tag{3.21}$$

where the lookup function searches the lookup table in the memory of the g_i function which for each information set stores the action $\pi_i(I)$ the pure strategy prescribes in this information set. The space complexity of the g_i function is linear in the size of the game tree and searching the lookup table can be also done in time at most linear in the size of the game tree.

Because the g_i function always returns the action prescribed by π_i in the given information set, the schema (M_i, f_i, g_i) is consistent with the game G. It prescribes the strategy π_i and since the automaton has only one state, the schema is also minimal and μ_Q-non-reducible. Such schema can be generated for every pure strategy and therefore every μ_Q-non-reducible schema in G has only one state and one transition. An example of a μ_Q-non-reducible schema is depicted in Figure 3.1.

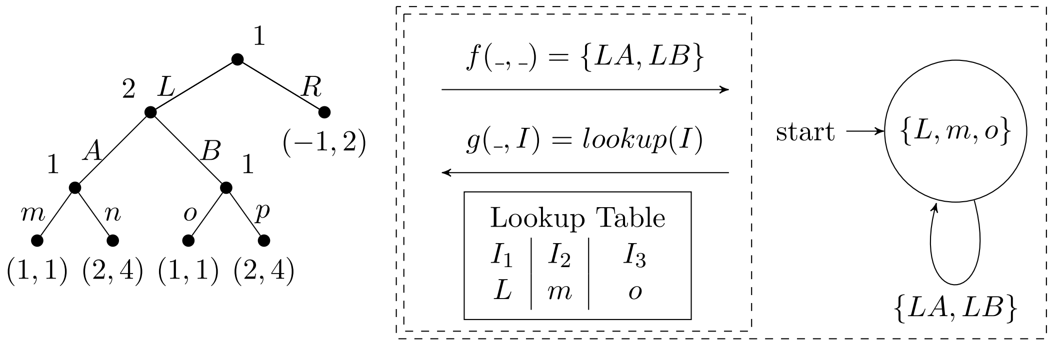

Figure 3.1: A schema with memory-demanding distinguishing functions, but a one-state automaton. The figure follows the standard denotations of an extensive-form game and a game automaton.

Similarly to the μ_Q measure, where the whole pure strategy was coded up into the structure of the distinguishing function, with the μ_S measure the strategy can be stored in the structure of the automaton. Again, let $\pi_i \in \Pi_i$ be a pure strategy the player i would like to play in the game $G = (\mathrm{N}, \mathrm{H}, \mathrm{Z}, \mathrm{A}_G, \chi, \rho, \sigma\ \mathrm{u}, \mathrm{C}, \mathrm{I})$. The schema (M_i, f_i, g_i) is defined so that M_i is the canonical automaton prescribing π_i. The number of states in this automaton is linear in the size of the game tree, because for every information set of player i there is one state. The f_i distinguishing function is then defined as

$$f_i(\_, \sigma) = \sigma, \tag{3.22}$$

which means the f_i function just passes the extensive-form-game observation it receives from the game to the automaton. The space complexity of f_i is hence minimal – the same for all pure strategies. The g_i function is defined as

$$g_i(q, \_) = B(q), \tag{3.23}$$

such that g_i only return the action the automaton prescribes in a given state. In case the value of function B can be retrieved in constant space, the space complexity of the

g_i function is also minimal and the same for all strategies – dependent only on function B.

Because the automaton M_i is the canonical automaton of pure strategy π_i and the distinguishing functions just pass the observations from game tree to the automaton and the actions from the automaton to the game tree, the schema (M_i, f_i, g_i) is consistent with the game G. It prescribes the strategy π_i and since the distinguishing functions do not perform any computations, the schema is also μ_S-non-reducible. The algorithm for reducing the size of the automaton from Proposition 2 can be used to make the schema also μ_Q-non-reducible. The same approach for defining a schema can be used for any pure strategy and therefore every μ_S-non-reducible schema in G has only trivial distinguishing functions. In Figure 3.2 is given an example of a constructed μ_S-non-reducible schema playing a pure strategy $\{L, m, o\}$. □

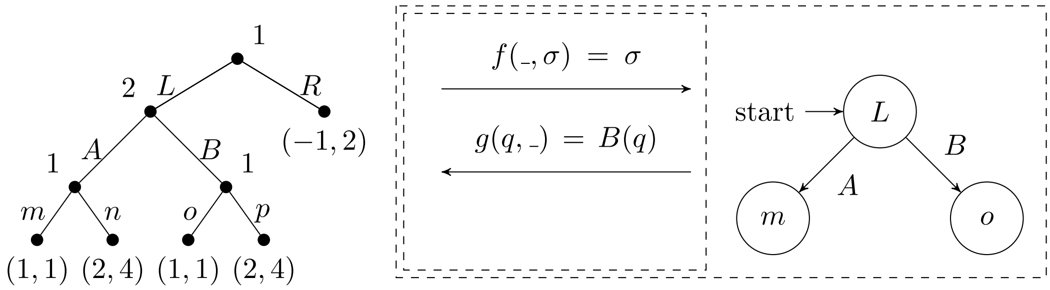

Figure 3.2: A schema with a large automaton, but simple distinguishing functions. The figure follows the standard denotations of an extensive-form game and a game automaton.

However, for sufficiently large games, every bounded-rational player would see only an abstraction of the tree. For example in poker, not even professional players are able to reason about all possible future situations in the game. In this case, the ability of the players to strategize is affected by the abstraction of the game the players use to reason about the future [27]. As Rubinstein noticed [60], reasoning in games with large game trees is best modeled as reasoning in infinite games. In this spirit, the distinguishing functions are defined for games with a parameter affecting a size of a game tree.

Definition 37 (Size-parametric class of games)**.** *A sequence of extensive-form games $G_1, G_2, \ldots$ is a size-parametric class of games $\mathcal{L}$ if and only if for all $i \in \mathbb{N}$ and for two consecutive games $G_i = (N_i, H_i, Z_i, A_i, \chi_i, \rho_i, \sigma_i, u_i, C_i, I_i)$ and $G_{i+1} = (N_{i+1}, H_{i+1}, Z_{i+1}, A_{i+1}, \chi_{i+1}, \rho_{i+1}, \sigma_{i+1}, u_{i+1}, C_{i+1}, I_{i+1})$ it holds that*

1. $N_i = N_{i+1}$;
2. $H_i \subseteq H_{i+1}$;
3. $A_i = A_{i+1}$;
4. $\chi_{i+1} \restriction (H_i \backslash Z_i) = \chi_i$;
5. $\rho_{i+1} \restriction (H_i \backslash Z_i) = \rho_i$;
6. $\sigma_{i+1} \restriction (H_i \backslash Z_i) = \sigma_i$;
7. $C_i = C_{i+1}$; *and*
8. $I_i \subseteq I_{i+1}$.

The n^{th} game in the size-parametric class $\mathcal{L}$ is denoted $\mathcal{L}(n)$. The idea of distinguishing functions for schemata playing games in class $\mathcal{L}$ is that a player's reasoning about his strategy is similar in all games belonging to this class. Given a finite game $\mathcal{L}(n)$, a player decides which pure strategy he would like to play.

Definition 38 ($\mathcal{L}$-consistent distinguishing functions)**.** *The distinguishing function f, g are consistent with a size-parametric class $\mathcal{L}$ ($\mathcal{L}$-consistent) if and only if for every extensive-form game $\mathcal{L}(n)$ there exists an automaton M such that a schema (M, f, g) is $\mathcal{L}(n)$-consistent.*

In order to formalize the ability of the players to maintain only an abstraction of the large game, the bounds on the maximal complexity of the schemata are introduced. A schema which complies with these bounds is called *small*.

Definition 39 (Small schema)**.** *Let G = (N, H, Z, A_G, χ, ρ, σ u, C, I) be an extensive-form game belonging to a size-parametric class $\mathcal{L}$ and (M_i, f_i, g_i) be a G-consistent schema of player i consisting of a game automaton M_i = (Q, q_0, A_M, B, O, δ) and $\mathcal{L}$-consistent distinguishing functions f_i and g_i. (M_i, f_i, g_i) is called small (supersmall) if and only if the number of states of M_i is polylogarithmic (square-root-of-logarithmic) in the size of the game tree and the distinguishing functions belong to the class of LOGSPACE functions in the size of the game tree.*

Small schemata are similar to the concept of small automata, related to efficient representation of strategies in repeated games. For small schemata, the constructions which were done in the proof of Observation 4 are generally not possible. The reason is that to have a small automaton, the distinguishing functions have to implement a mapping of information sets to actions. This is not possible with $\mathcal{L}$-consistent distinguishing functions used for all games in this class. Being small also affects the computational complexity of the equilibria. In case the strategies are restricted to those which have supersmall representation, the computation of a specific EFCE (defined in Section 1.2.2) can be done efficiently.

Proposition 6. *Let G = (N, H, Z, A, χ, ρ, σ u, C, I) be a multi-player perfect-recall extensive-form game from a size-parametric class of games $\mathcal{L}$ with a number of actions independent on the size of the game tree and a guaranteed maximum number of actions before every player acts again (or the game ends for him), and let the verification that the schema is supersmall be polynomial for $\mathcal{L}$. With pure schemata strategies restricted to supersmall schemata, the problem MAXPAY-EFCE in schemata is polynomial.*

Proof. Let every player select a state schemata complexity measure μ_Q and the upper bound $\sqrt{\log(|I|)}$ on the complexity such that every supersmall schema complies with this upper bound. Consider the number of possible abstracted actions, abstracted observations, and automata of supersmall size. Behaviorally equivalent distinguishing functions do not have to be considered because of the complexity measure and the definition of EFCE. The number of possible abstracted actions B_A in a given game tree is at most the number of nonempty subsets of A, in case the condition forbidding the actions which can be played in the same information set to be in the same subset in the partition is omitted. This number can be bounded as follows. The constants C_1 and C_2 are suitable constants independent on the number of information sets.

$$B_A \leq 2^{|A|} - 1 \leq C_1 \tag{3.24}$$

The number of possible abstracted observations B_O is equal to the number of nonempty subsets of the set of possible extensive-form-game observations, which means it is

$$B_O \leq 2^{|A|^c} - 1 \leq C_2, \tag{3.25}$$

because the number of extensive-form-game observations is at most $|A|^c$. It is the number of sequences in the $|A|$-ary tree with constant depth $c \geq 1$, where c is a guaranteed maximum number of actions before every player acts again. In the automaton with k states, there are k possibilities of which state can be the initial one and each state prescribes one abstracted action or a "noop" action. The number of transitions is at most k^2 and each can correspond to one of the B_O abstracted observations. The number of supersmall schemata with size at most $\sqrt{\log(|I|)}$ is hence at most

$$\sum_{k=1}^{\sqrt{\log(|I|)}} k{B_O}^{k^2}(B_A+1)^k \leq \log(|I|){C_2}^{\log(|I|)}(C_1+1)^{\sqrt{\log(|I|)}}, \tag{3.26}$$

in case that every mapping using a distinguishing function is realizable by a LOGSPACE function. Because the number of actions does not depend on the number of information sets, the number of supersmall schemata is polynomial in the size of the game tree (i.e. $|I|$). Because both the minimization and the reduction of a schema can be done in polynomial time (by Propositions 1 and 2) and the verification that the schema is supersmall is polynomial for $\mathcal{L}$, the set $\tilde{S}_i$ of all supersmall schemata can be generated in polynomial time.

Now the constraints for EFCE are adapted for schemata. Recall that a strategy profile *agrees* with a given leaf z if z is reached in case the players take their actions according to the strategy profile. With schemata profiles, a (partial) pure schemata profile ϕ *agrees* with z if and only if $z \in \bigcap_{(M,f,g)\in\phi} Z_{(M,f,g)}$. Moreover, a schema $s \in \tilde{S}_i$ *agrees* with a sequence $\sigma \in \Sigma_i$ if and only if it prescribes all actions which belong to σ. The first constraint defined for every action a in the game tree hence looks as follows.

$$u(a) = \sum_{\substack{t\in Z:\\ a\in seq_i(t)}} u_i(t)C(t) \sum_{\substack{\phi\in\tilde{S}_1\times\cdots\times\tilde{S}_n\\ z\in\bigcap_{(M,f,g)\in\phi} Z_{(M,f,g)}}} \lambda(s) \tag{3.27}$$

The second constraint is then defined for every pair of actions (a, d), such that $a \in \chi(I'), d \in \chi(I)$ and I precedes I'.

$$\begin{aligned} v(I,a) \geq & \sum_{s_i\in agr(seq_i(I')a)} \sum_{\substack{t\in Z:\\ seq_i(I)d=seq_i(t)}} \sum_{\phi_{-i}\in agr(t)} u_i(t)C(t)\lambda(s_i,\phi_{-i}) \\ & + \sum_{\hat{l}:seq(\hat{l})=seq_i(I)b} v(\hat{l},a) \end{aligned} \tag{3.28}$$

The last constraint does not change and is defined for every actions $a \in \chi(I)$ in the game tree.

$$u(a) = v(I,a) \tag{3.29}$$

Finally, the criterion function maximizes the sum of individual expected utilities, which means it is

$$\max \sum_{i\in N}\sum_{t\in Z} u_i(t)C(t) \sum_{\substack{\phi\in\tilde{S}_1\times\cdots\times\tilde{S}_n\\ z\in\bigcap_{(M,f,g)\in\phi}}} \lambda(s) \tag{3.30}$$

In general extensive-form games with traditional strategies, the description of MAXPAY-EFCE using a liner program has a polynomial number of constraints, but an exponential number of variables. The reason is that the linear program has one variable for every pure strategy profile and the number of pure strategy profiles is exponential in the size of the game tree. However, because the number of supersmall schemata is polynomial, the linear program for computing EFCE has polynomial number of both the constraints and variables and hence it can be solved in polynomial time.

□

The class of games which satisfies the conditions required for MAXPAY-EFCE to be computed efficiently (Proposition 6) is large. The first condition, requesting that the number of actions is independent on the number of informations sets, is besides other classes of games satisfied by most card games in which the game is played with fixed set of cards over a variable number of rounds (e.g. poker); or games on graphs with variable number of steps. The second condition requests that for every player there is a constant upper bound on the maximum number of actions of other players during which the player has to wait for his turn. Also this condition is satisfied by most classes of man-made games.

(Super)small schemata are models of efficiently representable strategies. Computing a specific EFCE is easy with supersmall schemata, but generally an NP-hard problem [46] in an unrestricted game. The same observation cannot be made about Nash equilibrium, because there is no known description of Nash equilibrium in general games which cannot be solved efficiently only because of the exponential number of pure strategies.

3.2 Games with a small number of observations

This section presents a sufficient condition for a class of games to have efficiently representable pure strategies using small schemata. Moreover, it gives an example of such a class – a situation which regularly occurs in security scenarios.

Because every pure strategy defines an action in every information set of a game tree, its representation using a game-playing schema has to be able to react to every possible progression of the game accordingly. These reactions have to be consistent with the game tree, which is ensured by the observations perceived by the player. One way to make the representations small is to force the number of observations to be always bounded, as a player takes his actions during the game. In the following observation, I_i^j denotes the number of information sets of player i in which he can find himself after taking any sequence of j actions according to a fixed pure strategy π_i.

Observation 5. *Let G be an extensive-form game and π_i be a pure strategy of player i and let b_1, b_2 be two constants, such that for every j the number of informations sets in which he can find himself after taking a sequence of j actions according to π_i satisfies*

$$I_i^j \leq I_i^{j-1} + b_1 j^{b_2}, \quad b_1, b_2 \in \mathbb{N}. \tag{3.31}$$

Then there exists a small schema which implements π_i.

Proof. The initial number of information sets I_i^0 before performing any action is always constant. Because the number of information sets grows at most according to inequality 3.31, by the theory of recurrent relations it means that the number of information sets after taking a fixed sequence of j actions is at most polynomial in depth of the game tree.

$$I_i^j \leq p(j), \tag{3.32}$$

where p is a suitable polynomial. Therefore, the number of decisions a player might be able to make during the game-play is at most polynomial, because

$$\sum_{d=0}^{d_{max}} I_i^d \leq \sum_{d=0}^{d_{max}} p(d) \leq d_{max} p(d_{max}), \tag{3.33}$$

where d_{max} is a maximum length of sequence of actions player i can take in the game. Even with trivial distinguishing functions, the number of states of an automaton in schema prescribing π_i is also polynomial.

On the other hand, the number of information sets of player i in the whole game tree is still exponential in the depth of the game tree, because in games of perfect recall, an approximate number of information sets is at least

$$\sum_{d=0}^{d_{max}} a_{min}^{d} \geq a_{min}^{d_{max}}, \tag{3.34}$$

where a_{min} is a minimum number of actions which can be performed in any information set of player i, assuming that this number is at least 2. The number of states of an automaton in schema prescribing π_i is therefore polylogarithmic in the size the game tree. The schema is hence small. □

The inequality 3.31 states that player's perception of what is happening in the game is limited, so that a number of distinguishable situations he might face after taking each action is sufficiently bounded. When a number of observations a player can receive is small, an automaton grows according to condition 3.31 while maintaining a tree structure. A number of transitions hence grows at most at the same rate as a number of information sets. Consider a following example of class of games which satisfies condition 3.31.

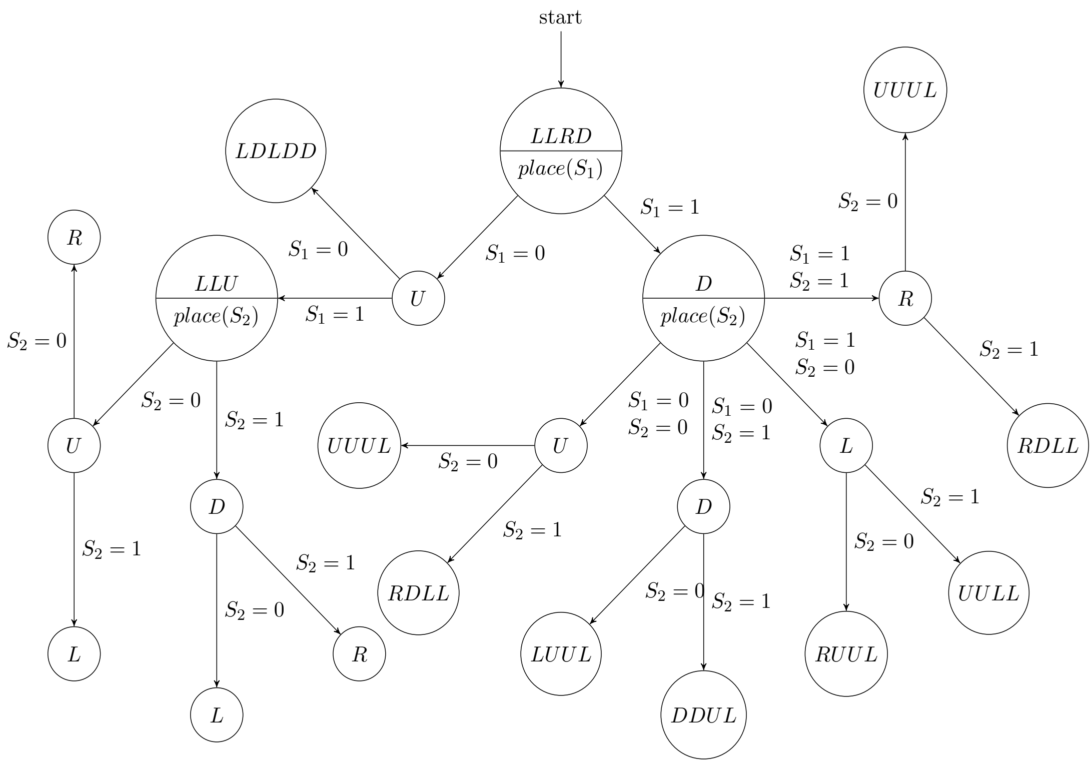

Figure 3.3: An automaton with size polynomial in the size of the described pursuit-evasion game tree. The game consists of one pursuer and one evader. The graph is a two-dimensional grid with size 11 and with available actions L, R, U and D, if applicable. The pursuer starts from the middle of the grid, he has two sensors S_1 and S_2 which collect binary data from the neighboring nodes, have infinite communication range, a 2-step battery life and are immediately activated after a drop. The initial location of the follower is chosen at random. The number of steps the game takes is 10. The figure follows a standard denotation of a game automaton, except that in case no observation can be provided, for better readability the chain of states is merged into one state prescribing a sequence of actions.

Consider a pursuit-evasion game, in which a group of players (called pursuers) attempts to track down members of another group (called evaders) while moving on a general graph. Each pursuer obtains a *constant and finite* number of sensors S of different types and with different properties, such as susceptibility range, communication range, battery life and the ability of remote activation. As the pursuers move across the graph, they place individual sensors in the nodes. In case the physical model of the sensor enables the pursuers to receive the information (the sensor is active, the pursuers are within range, etc.), they might use it to obtain the information and react accordingly. Since the technical attributes of the sensors are *constant and finite* (especially the battery life), a maximum branching factor of the game tree is $SR^{|S|}$, where SR is a maximum number of distinct data the sensor can obtain within a given susceptibility range and $|S|$ is a number of sensors. A number of information sets grows approximately

$$I^j \leq I^{j-1} + SR^{|S|^{BL}}, \tag{3.35}$$

where BL is a maximum battery life; even though the size of the game tree grows exponentially in a number of performed moves across the graph. Therefore, for every pure strategy of the pursuers, there is a small schema which prescribes it. For example, an automaton of a schema with trivial distinguishing functions playing such a pursuit-evasion game can have a structure depicted in Figure 3.3.

Moreover, the sensibility of the sensor network can grow with the number of sensors placed. In that case, the number of information sets is bounded by

$$I^j \leq I^{j-1} + SR^{|S|^{BL}}(s|P|j)^{BL}, \tag{3.36}$$

where $|P|$ is a number of pursuers and s is a suitable constant representing the increase in sensibility. The inequality 3.36 also satisfies the condition of inequality 3.31. Another possibility might be to introduce the ability of pursuers to collect some resources as the game progresses and invest it in a predetermined number of sensors since constant-size combinations also grow polynomially with a number of items from which it is possible to choose. For example, this ability might correspond to recharging some sensors during the game-play.

3.3 Games with a small number of situations

The previous section considered games in which every strategy is efficiently representable because the number of observations is small. In this section, this property is generalized for games with a small number of situations.

Definition 40. *(Game situation) Given an extensive-form game G = (N, H, Z, A, χ, ρ, σ, u, C, I), let P_I be a maximal partition of I_i into mutually disjoint subsets, such that all information sets belonging to the same part of P_I have isomorphic subtrees (including utilities). Then the parts of partition P_I form the set of game situations for player i, associated with P_I.*

A pure strategy prescribes the same action in every situation if and only if in each situation the actions taken in all information sets in this situation lead to the forest of isomorphic subtrees. Note that because the game trees are isomorphic for every situation, an optimal strategy is the same in all information sets belonging to the same game situation. In the following observation, S_i^j denotes the number of situations of player i in which he can find himself after taking any sequence of j actions according to a fixed pure strategy.

Observation 6. *Let G be an extensive-form game and π_i be a pure strategy of player i taking the same action in every situation and let b_1, b_2 be two constants, such that* *for every j the number of situations in which the player can find himself after taking a sequence of j actions according to* π_i *satisfies*

$$S_i^j \leq S_i^{j-1} + b_1 j^{b_2}, \quad b_1, b_2 \in \mathbb{N}. \tag{3.37}$$

Then there exists a small schema which implements π_i*.*

Proof. By the same argument as in the previous proof, there exists an automaton with size polylogarithmic in the size of the game tree prescribing π_i, because the number of situations is also polylogarithmic. The only problem is that in this case, the distinguishing functions might not be trivial because the actions leading to the forests of isomorphic subtrees in the information sets belonging to the same situation might not be the same. Let the abstracted action prescribed by the automaton in every situation be the certificate of the forest into which the action leads. The forests are the same in all information sets belonging to the same situation, because of the definition of π_i. In every information set, the distinguishing function g_i computes the certificate of a forest for every action which can be taken in this information set. Because the construction of a certificate is a LOGSPACE problem [61] and the maximum number of certificates which have to be computed before the action is identified is $|A|$, the complexity of finding the action which has to be played in the information set is LOGSPACE. The size of the automaton remains polylogarithmic because it has a polylogarithmic number of states and each state is assigned a certificate which size is also logarithmic. Therefore, the schema prescribing π_i is small. □

As an example of such a class of games, in which the number of situations is small, consider the games on hypercubes. The number of possible positions in the cube grows only polynomially with the depth. In case the utilities depend only on the position in the cube and not on the path leading to this position (which is a common property in a lot of games, e.g. in Tic-Tac-Toe), the number of situations in this game grows according to inequality 3.37.

Conclusion

This thesis introduces a theory of models of strategies in extensive-form games called game-playing schemata. In this model, the strategies are represented by a structure consisting of a finite automaton and two computational functions. The automaton represents a structured memory of the players, while the functions represent their computational abilities. Together, the schema is a realization of a pure strategy which can be implemented by players in order to play a given game. The applications of related models can be found frequently in cognitive modeling of dynamic situations. For example, the IBL model which is proved to provide robust predictions about the strategies of players in repeated games is based on the bounded memory of the players and their propensity to forget past observations and actions.

The thesis first presented a model of automata playing repeated normal-form games and explained the main complications of extending this model to extensive-form games. It introduced and discussed a model of game-playing schemata which is able to play extensive-form games correctly. Then the thesis described methods for evaluating the structural complexity of schemata and provided an algorithm minimizing the memory of a given schema. The analysis of dynamic decision making and solution concepts using schemata provided the following result.

Propositions 4 and 5. *Every multi-player, perfect-recall extensive-form game has a schemata profile ϕ^{Nash} such that ϕ^{Nash} is a (balanced) Nash equilibrium in schemata strategies.*

However, the computation of Nash equilibrium using schemata is at least as hard as computing Nash equilibrium using traditional strategies – a PPAD-hard problem. The thesis subsequently introduced the class of small automata, which is a class of efficiently representable strategies. The investigation of the computational complexity led to the second main result.

Proposition 6. *Let G be a multi-player perfect-recall extensive-form game with a number of actions independent on the size of the game tree and a guaranteed maximum number of actions before every player acts again (or the game ends). With pure schemata strategies restricted to supersmall schemata, the problem MAXPAY-EFCE in schemata is polynomial.*

Finally, the thesis introduced two conditions under which the restriction to small strategies can be done without any loss on optimality. The first condition restricts a number of observations which a player can receive, while the second condition restricts a number of situations a player can encounter.

Future work

This thesis introduced a theory of schemata playing extensive-form games. It presented new concepts related to modeling bounded rationality and verified that the definitions provide novel approaches and results. However, the work serves mainly as a proof of concept. There are multiple directions into which the theory of schemata might be extended.

Cooperative outcomes. In repeated games, it was proved that restriction to small automata has significant consequences on cooperative outcomes of games being a part of equilibrial strategies. Similar effects on the structure of equilibrium might be shown also for schemata.

Relation to abstractions. Schemata seem to be closely related to imperfect recall abstractions for extensive-form games. Also imperfect recall abstractions model bounded memory of players through their ability to forget their past actions. The connection between these two concepts should be more precisely analyzed so that both theories can benefit from results originated in each other.

Inapproximability. Because schemata are realizations of pure strategies, the existence of small schemata is dependent on the ability of distinguishing functions to abstract the game tree. The approximability of a given strategy is hence related to the Kolmogorov complexity of game trees, which gives rise to questions about lower bounds for algorithms minimizing the complexity of schemata prescribing a pure strategy.

Balanced equilibria. So far the examples were given of classes of games with small schemata implementing pure strategies in these games; and the consequences on the computational complexity of solution concepts. The properties of balanced equilibria should be analyzed in a similar way, since balanced equilibria better reflect the original intention of schemata to balance the expected utility and structural complexity.

Behavioral validity. The model of schemata was proposed as a behavioral model of strategizing in large extensive-form games, which has well-defined theoretical properties. The experiments with human subjects might be conducted in order to verify the hypotheses related to the behavioral nature of this model. Especially, whether the equilibria in games in which all pure strategies are implementable by small schemata better predict the behavior of human subjects; and moreover, whether the restriction to small schemata helps to provide better predictions even in games, in which a lot of pure strategies are not efficiently representable. Verifying these hypotheses would require designing specific game scenarios in order to draw defensible conclusions.

Bibliography

[1] John F. Nash. Non-Cooperative Games. *Annals of Mathematics*, 54(2), 1951.

[2] Robert Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, 1974.

[3] Thomas C. Schelling. *The Strategy of Conflict.* Harvard University Press, 1980.

[4] Milind Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned.* Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[5] K. Vela Velupillai. Foundations of boundedly rational choice and satisficing decisions. *Advances in Decision Sciences*, 2010, 2010.

[6] Cassey Lee. Bounded rationality and the emergence of simplicity amidst complexity. *Journal of Economic Surveys*, 25(3):507–526, 2011.

[7] Herbert A. Simon. Bounded rationality in social science: Today and tomorrow. *Mind & Society*, 1(1):25–39, 2000.

[8] Thanh H. Nguyen. *Combating Adversaries under Uncertainties in Real-world Security Problems: Advanced Game-theoretic Behavioral Models and Robust Algorithms.* PhD thesis, University of Southern California, 5 2016.

[9] Yasaman D. Abbasi. *Modeling Human Bounded Rationality in Opportunistic Security Games.* PhD thesis, University of Southern California, 2016.

[10] Ariel Rubinstein. *Modeling bounded rationality.* MIT press, 1998.

[11] Michael W. Eysenck and Mark T. Keane. *Cognitive psychology: A student's handbook.* Taylor & Francis, 2000.

[12] Herbert A. Simon. Theories of bounded rationality. *Decision and organization*, 1(1):161–176, 1972.

[13] Reinhard Selten. What is bounded rationality. *Bounded rationality: The adaptive toolbox*, 13:13–36, 2001.

[14] Colin Camerer. *Behavioral game theory: Experiments in strategic interaction.* Princeton University Press, 2003.

[15] Richard D. McKelvey and Thomas R. Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.

[16] Daniel L. McFadden. Quantal choice analaysis: A survey. In *Annals of Economic and Social Measurement, Volume 5, number 4*, pages 363–390. NBER, 1976.

[17] Tomás Lejarraga, Varun Dutt, and Cleotilde Gonzalez. Instance-based learning: A general model of repeated binary choice. *Journal of Behavioral Decision Making*, 25(2):143–153, 2012.

[18] Cleotilde Gonzalez and Varun Dutt. Instance-based learning: Integrating sampling and repeated decisions from experience. *Psychological review*, 118(4):523, 2011.

[19] Cleotilde Gonzalez, Javier F. Lerch, and Christian Lebiere. Instance-based learning in dynamic decision making. *Cognitive Science*, 27(4):591–635, 2003.

[20] Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5(4):297–323, 1992.

[21] Robert Aumann. Perspectives on bounded rationality. In *Proceedings of the 4th conference on Theoretical aspects of reasoning about knowledge*, pages 108–117. Morgan Kaufmann Publishers Inc., 1992.

[22] Nimrod Megiddo. *Remarks on bounded rationality.* IBM Corporation, 1986.

[23] Ariel Rubinstein. Finite automata play the repeated prisoner's dilemma. *Journal of economic theory*, 39(1):83–96, 1986.

[24] Robert E. Marks. *Repeated games and finite automata.* Australian Graduate School of Management, University of New South Wales, 1990.

[25] Ehud Kalai. Bounded rationality and strategic complexity in repeated games. *Game theory and applications*, pages 131–157, 1990.

[26] Itzhak Gilboa. The complexity of computing best-response automata in repeated games. *Journal of economic theory*, 45(2):342–352, 1988.

[27] Soumya Paul, R. Ramanujam, and Sunil Simon. Automata and compositional strategies in extensive form games. In *Models of Strategic Reasoning*, pages 174–201. Springer, 2015.

[28] Nimrod Megiddo and Avi Wigderson. On play by means of computing machines: preliminary version. In *Proceedings of the 1986 Conference on Theoretical aspects of reasoning about knowledge*, pages 259–274. Morgan Kaufmann Publishers Inc., 1986.

[29] Joseph Y. Halpern, Rafael Pass, and Lior Seeman. Computational extensive-form games. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 681–698. ACM, 2016.

[30] James Pita, Manish Jain, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. Robust solutions to stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence*, 174(15):1142–1171, 2010.

[31] Rong Yang, Christopher Kiekintveld, Fernando Ordonez, Milind Tambe, and Richard John. Improving resource allocation strategy against human adversaries in security games. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 22, pages 458–466, 2011.

[32] Albert X. Jiang, Thanh H. Nguyen, Milind Tambe, and Ariel D. Procaccia. *Decision and Game Theory for Security.* Springer International Publishing, 11 2013.

[33] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations.* Cambridge University Press, 2009.

[34] Eric Rasmusen. *Games and Information: An Introduction to Game Theory*. Blackwell, 2001.

[35] John von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

[36] Harold W. Kuhn. Extensive games and the problem of information. *Annals of Mathematics Studies*, 28, 1953.

[37] Robert Aumann. Survey of repeated games. *Essays in game theory and mathematical economics in honor of Oskar Morgenstern*, 1981.

[38] Jeffrey S. Banks and Rangarajan K. Sundaram. Repeated games, finite automata, and complexity. *Games and Economic Behavior*, 2(2):97–117, 1990.

[39] Christos H. Papadimitriou and Mihalis Yannakakis. On complexity as bounded rationality. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 726–733. ACM, 1994.

[40] Leonid G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53 – 72, 1980.

[41] Narendra K. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, December 1984.

[42] Victor Klee and George J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities*, volume III, pages 159–175. Academic Press, New York, 1972.

[43] Françoise Forges. An approach to communication equilibria. *Econometrica*, 54(6):1375–85, 1986.

[44] Christos H. Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM (JACM)*, 55(3):14–23, 2008.

[45] Albert X. Jiang and Kevin Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 119–126. ACM, 2011.

[46] Bernhard von Stengel and Françoise Forges. Extensive-form correlated equilibrium: Definition and computational complexity. *Math. Oper. Res.*, 33(4):1002–1022, 2008.

[47] Michael Spence. Job market signaling. *The Quarterly Journal of Economics*, 87(3):355–374, 1973.

[48] Wan Huang. *Equilibrium Computation for Extensive Games*. PhD thesis, London School of Economics, 2011.

[49] Abraham Neyman. Bounded complexity justifies cooperation in the finitely repeated prisoners' dilemma. *Economics letters*, 19(3):227–229, 1985.

[50] Herbert A. Simon. Rational choice and the structure of the environment. *Psychological review*, 63(2):129–141, 1956.

[51] Nathaniel Love, Timothy Hinrichs, David Haley, Eric Schkufza, and Michael Genesereth. General game playing: Game description language specification. Technical report, 2008.

[52] Juris Hartmanis and Richard E. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[53] Leonard Pitt. Inductive inference, DFAs, and computational complexity. *Analogical and inductive inference*, pages 18–44, 1989.

[54] Martin Hühne. Linear speed-up does not hold on Turing machines with tree storages. *Information processing letters*, 47(6):313–318, 1993.

[55] Ivan H. Sudborough and A Zalcberg. On families of languages defined by time-bounded random access machines. *SIAM Journal on Computing*, 5(2):217–230, 1976.

[56] Amir M. Ben-Amram, Niels H. Christensen, and Jakob G. Simonsen. Computational models with no linear speedup. *Chicago Journal of Theoretical Computer Science*, 7:1–24, 2012.

[57] Anil Nerode. Linear automaton transformations. *Proceedings of the American Mathematical Society*, 9(4):541–544, 1958.

[58] Edward F. Moore. Gedanken-experiments on sequential machines. *Automata studies*, 34:129–153, 1956.

[59] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. Introduction to automata theory, languages, and computation. *ACM SIGACT News*, 32(1):60–65, 2001.

[60] Ariel Rubinstein. Comments on the interpretation of game theory. *Econometrica: Journal of the Econometric Society*, pages 909–924, 1991.

[61] Steven Lindell. A logspace algorithm for tree canonization. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 400–404. ACM, 1992.

List of Figures

Acronyms

CE	Correlated equilibrium.
EFCE	Extensive-form correlated equilibrium.
EFG	Extensive-form game.
FRG	Finitely repeated games.
GGP	General game playing.
IBL	Instance-based learning.
MAID	Multi-agent influence diagram.
NE	Nash equilibrium.
NFG	Normal-form game.
NP	Nondeterministic polynomial time.
PD	Prisoner's dilemma.
PPAD	Polynomial parity argument, directed version.
PT	Prospect theory.
QR	Quantal response.
TM	Turing machine.

Functions and Symbols

(M, f, g)	A game-playing schema.
$-i$	A set of opponents of player i.
A	A set of (a) sets of actions for each player; or (b) actions of an automaton.
$A(h)$	A set of actions in node h.
B	A behavioral function of an automaton.
$C(a)$	A probability function for performing a chance action a.
$C(h)$	A probability of reaching node h due to chance.
$Ext(\sigma_i)$	A set of extensions of a sequence σ_i.
H	A set of nodes in a game tree.
M	A game automaton.
M_π	A canonical automaton associated with a pure strategy π.
N	A set of players.
O	A set of observations of an automaton.
Q	A set of states of an automaton.
S^μ	A set of minimal G-consistent non-reducible schemata with respect to measure μ.
Z	A set of terminal nodes in a game tree.
Δ	A set of mixed strategies.
Φ^μ	A set of probability distributions over S^μ.
Π	A set of pure strategies.
Π^*	A set of reduced pure strategies.
Σ	A set of sequences in a sequence-form game representation.
β	A behavioral strategy.
δ	(A) a mixed strategy; or (b) a transition function of an automaton.
$\mathcal{L}$	A size-parametric class of games.
μ	A measure of (a) automata or (b) schemata complexity.
μ_p	A complexity measure associated with property p, where p is (a) Q for number of states; (b) δ for number of transitions; (c) T for number of time steps; or (d) S for size of memory.
$\mathbb{N}$	A set of natural numbers.
$\overline{\Sigma}$	A set of sets of extensive-form-game observations for each player.
ϕ	A schemata profile.
π	A pure strategy.
$\preceq_\mu$	A partial order generated by measure μ.
ψ^μ	A utility/complexity balancing function with respect to measure μ.
$\mathbb{R}$	A set of real numbers.
$\rho(h)$	A player function for node h.
σ	A sequence in a sequence-form game representation.
$\sqsubseteq$	A prefix relation on sequences.
$agr(\sigma)$	A set of agreeing strategies for sequence σ.

$agr(h)$	A set of (possibly partial) agreeing strategy profiles for node h.
f, g	The distinguishing functions.
$g_i(\sigma_1, ..., \sigma_n)$	An extended utility function for player i.
$inf_i(\sigma_i)$	An information set in which the last action of σ_i is taken.
$p(\sigma_1, ..., \sigma_n)$	A correlation plan of sequences $\sigma_1, ..., \sigma_n$.
q_0	An initial state of an automaton.
$r_i(\sigma_i)$	A realization plan of sequence σ_i for player i.
$rel(\sigma_i)$	A set of sequences of $-i$ which form a relevant pair with σ_i.
$seq_i(I)$	A set of sequences leading to information set I for player i.
$seq_i(h)$	A set of sequences leading to node h for player i.
$u_i(a_1, ..., a_n)$	A utility function for player i.