Randomness, Predictability, and Complexity in Repeated Interactions

v0.02

Preliminary version, please do not cite.

Shuige Liu\textsuperscript{a}, Zsombor Z. M\'eder\textsuperscript{b,*}

\textsuperscript{a}Faculty of Political Science and Economics, Waseda University, 1-6-1, Nishi-Waseda, Shinjuku-Ku, 169-8050, Tokyo, Japan.
\textsuperscript{b}Humanities, Arts and Social Sciences, Singapore University of Technology and Design, 8 Sompah Road, 487372 Singapore.

Abstract

Nash equilibrium often requires players to adopt a mixed strategy, i.e., a randomized choice between pure strategies. Typically, the player is asked to use some \textit{randomizing device}, and the story usually ends here. In this paper, we will argue that: (1) Game theory needs to give an account of what counts as a random sequence (of behavior); (2) from a game-theoretic perspective, a plausible account of randomness is given by algorithmic complexity theory, and, in particular, the complexity measure proposed by Kolmogorov; (3) in certain contexts, strategic reasoning amounts to modelling the opponent’s mind as a Turing machine; (4) this account of random behavior also highlights some interesting aspects on the nature of strategic thinking. Namely, it indicates that it is an \textit{art}, in the sense that it cannot be reduced to following an algorithm.

Keywords: repeated games, mixed strategy, Kolmogorov complexity, Turing machine.

\textit{JEL: C72, C73.}

\textsuperscript{5}The authors gratefully acknowledge the support of Grant-in-Aids for Young Scientists (B) of JSPS No. 17K13707, Grant for Special Research Project No. 2017K-016 of Waseda University, and Start-up Research Grant SRHA 15-110 of the Singapore University of Technology and Design.

\textsuperscript{*}Corresponding author

Email addresses:
shuige_liu@aoni.waseda.jp (Shuige Liu),
zsombor_meder@sutd.edu.sg (Zsombor Z. Méder)
1. Introduction

Repeated interactions are a core topic for the social sciences. Two principal reasons might be cited to account for this fact. First, compared to one-shot interactions, repeated ones account for the dominant share of social behavior. Our family members, friends, sexual partners, work colleagues, or business partners are usually stable over time. Indeed, even on the competitive market, a large share of interactions take place not between anonymous buyers and sellers, but between partners who engage in a series of trades and who are acquainted with each other. Second, stable behavioral and epistemic patterns, such as customs, conventions, and culture are possible and meaningful only through repeated interactions. Conversely, if our lives consisted of a succession of one-shot interactions, discovering regularities in social behavior would likely be prohibitively difficult.

The goal of many a social scientist is thus to acquire an understanding of repeated interactions. This involves dealing with three, interconnected problems:

1. Modeling the rules of repeated interaction.
2. Developing a descriptive and normative account of individual decision-making in such situations.
3. Discovering and describing the stable behavioral and epistemic patterns which emerge from these interactions.

In this paper, we discuss these three problem from the perspective of game theory. Game theory’s standard solution for problem (1) is to model a repeated interactions as a stage game which is recurring over time. To problems (2) and (3), game theory provides several answers, including the notion of a mixed-strategy Nash-equilibrium, refinements of the Nash equilibrium concepts applied to extensive-form games (subgame-perfect equilibrium etc.), the Folk Theorem, etc. We will argue that the idea of mixed-strategy equilibrium fails to capture essential elements of strategic decision-making in repeated situation. Further, we will argue that there is conceptual inconsistency between the theory of repeated games and mixed-strategy equilibria. Based on these arguments, in Sections 3 and 4 we will give an outline of our approach and show that it can overcome those difficulties and provide insights for our understanding of individual’s behavior in repeated situations.

2. Repeated games, Nash equilibrium, and repeated Situations

2.1. The problem of learning in repeated games

The theory of repeated games was developed to model repeated situations, and individuals’ behavior in them (Mertens and Sorin, 2015). Here, a repeated game is formulated as a super-game constituted by a sequence of stage games. An individual’s behavior in a repeated situation is then formulated as a strategy in this super game, that is, a pre-determined plan recommending an action for a player at every possible node. Therefore, this approach gives little space for the study of learning and adjustment which are important and essential in an individual’s behavior in repeated situations; after all, what does it mean by learning and adjustment if one’s behavior is nothing but following a pre-determined plan of actions?

This problem seems have long been noticed among game theorists, and, correspondingly, substantial work has been devoted to it. For example, learning theory (Fudenberg and Levine, 1998) and evolutionary game theory (Weibull, 1997) use various statistical methods to formulate players’ learning process in a repeated game. However, the core of this statistical method is set-theoretical model (Aumann, 1976), which itself has some conceptual problems (Kaneko, 2002; Liu, 2016). In addition, the statistical method cannot capture an individual’s initiative learning, which is thought by many psychologists to be an essential part of learning process (Levine, 1975). It can hardly be said that the problem of learning in repeated games has been fully solved.

The notion of Nash equilibrium was originally developed to describe some necessary condition for an outcome in a one-shot non-cooperative game (Nash, 1950, 1951). Like the notion of the market equilibrium, the Nash equilibrium concept can

[1] Recently, the statistical approach has been adopted and extended by computer scientists on researches of artificial intelligence, such as pattern recognition and machine learning and has achieved some significant breakthroughs (Bishop, 2010; Murphy, 2012). However, the problems mentioned above persist.
be interpreted as a stable pattern of behavior in a repeated situation (Kaneko, 2004, Chapter 2). Further, as Nash equilibrium is a vector of strategies in which no one can improve his payoff by a unilaterally deviation, it is conceptually more convenient to apply the Nash equilibrium concept in repeated, rather than one-shot situations, since in the latter achieving a Nash equilibrium is more demanding on each player’s epistemic abilities (Aumann and Brandenburger, 1995; Kaneko and Hu, 2013). There is a literature using pure-strategy Nash equilibrium to study social behavior patterns, (e.g. Kaneko and Matsui, 1999). On the other hand, there are some critical problems with the notion of mixed-strategy Nash equilibrium.

2.2. Interpretations of probability for mixed strategies

Let us start from the notion of a mixed strategy. Mathematically, a mixed strategy is a probability distribution over the set of pure strategies of a player. The first question concerns the meaning of “probability” in this context. Among various interpretations of probability ( Hájek, 2012), only the subjective and frequentist interpretation seem applicable to the game-theoretic setting. The subjective interpretation is a standard one for mixed strategies in one-shot games. According to this view, a mixed strategy represents the uncertainty about the opponents’ choices (cf. Aumann and Brandenburger, 1995; Binmore, 1994). One difficulty that appears here is that this interpretation is incompatible with the original meaning behind the concept of strategy, that is, a strategy of a player is, first of all, some plan of behavior carried out by that player, rather than some belief in other players’ mind (von Neumann and Morgenstern, 1947; Luce and Raiffa, 1957). Second, a player is meant to choose a strategy from a strategy set, that is, he is assumed to have causal control over the strategy he would follow. However, it is not immediately clear that a player can have such a direct influence about his opponent’s beliefs.

These difficulties disappear if we adopt the frequentist interpretation of probability. According to this, in a repeated game, the probability vector which represents a mixed strategy indicates the relative frequency of the use of each pure strategy. However, besides the problems inherent to the frequentist interpretation in general (cf. ?), a basic difficulty here is that setting merely the frequencies of strategies to a particular value is neither necessary, nor sufficient as a proper guide to a player’s behavior. To see that it is not necessary, consider a game with a Nash equilibrium for the stage game which prescribes a particular pure strategy to be used with probability 50%. If the game is played an odd number of times, it is impossible for the strategy to be played exactly with this relative frequency, even assuming the player has access to a proper randomizing device. To see that having the proper frequency is not a sufficiently good guide for behavior, consider a player who follows a simple pattern in his behavior, while following the desired frequency. If this behavioral pattern is detected by his partner, who adjusts his behavior accordingly, the game might be out of equilibrium soon. We will discuss this problem in more detail in Section 3.

We emphasize that these problems have long been noticed by game theorists (cf. von Neumann and Morgenstern, 1947; Dixit and Nalebuff, 1993). But most of them stops at informal and casual discussion. A few exceptions are found in Chatterjee and Sabourian (2012) and Hu (2010). We will discuss them briefly in the following.

Additional difficulties appear when we apply Nash equilibrium to repeated games. As pointed by Kaneko (1982), the Folk Theorem leads to so many Nash equilibria that it becomes impossible to select a reasonable one (or a subset of reasonable ones), or to recommend any strategy to the players. Even though there exists a literature on refining Nash equilibrium in repeated games (Mertens and Sorin, 2015 cf.), this difficulty has not been solved in principle. Thus, there are too many Nash equilibria in repeated games.

2.3. Act rationality and process rationality

In fact, there is a conceptual problem in how a mixed-strategy Nash equilibrium is to be used even for single-shot situations (and, by extension, to repeated ones), which is highlighted by the frequentist interpretation. For example, consider the Matching Pennies game. In the only Nash equilibrium, each player chooses one of his pure strategies with probability 0.5. But, crucially, each player then has to actually choose that pure strategy, and after doing so, this choice will not constitute an equilibrium: In particular, one player will lose, and so would like to change his strategy. Thus, when it comes to a repeated Matching Pennies, according to the frequentist interpretation, this “0.5” means that players use each of their actions in the
stage game with frequency 0.5. This then amounts to choosing a particular sequence of actions at the outset. But the profile of such sequences of action cannot be an equilibrium, as each round, one player will lose, and the other will win.

This dilemma arises from the fact that the notion of equilibrium is burdened with two, distinct meanings. On the one hand, being in a Nash equilibrium is a statement about which strategies players use – strategies which are mutual best responses. If an equilibrium profile is chosen, it will be in player’s self-interest to behave accordingly. Thus, being in an equilibrium is indicative of the rationality of behavior, and in this sense, is a notion of act rationality. On the other hand, the most common interpretation of a mixed-strategy equilibrium is for players to use a randomizing device (where this “random” is afterwards left unspecified). The behavior generated by this randomizing device does not seem to matter. This sense of equilibrium makes a claim about the processes through which players generate their behavior, and not the behavior – the sequence of actions chosen – in itself. Correspondingly, the idea of mixed strategies, interpreted as the use of a randomizing device, relies on a notion of process rationality.

2.4. The limits of ‘strategy’, and modeling repeated interaction

Part of the confusion is generated by the notion of “strategy”. As one of the most basic and classical concept in game theory (cf. von Neumann and Morgenstern [1947] Luce and Raiffa [1957]), a strategy is defined as a detailed plan which provides or prescribes a player some action at each possible node. A strategy implies that every possibility, every node in the game tree has been considered and an action assigned to it ex ante. Consequently, there is not much space for a player to learn, interpret the other’s behavior, guess, or adjust. Obviously, this assumption leads to very helpful simplifications for discussing Nash equilibrium etc. However, if the game tree is very large, computing or even storing such a strategy in any one mind becomes implausible, or in many cases, outright impossible. For example, the Shannon number of $10^{120}$ provides a conservative estimate on the number of possible chess board configurations. Clearly, even the most proficient chess players are nowhere close to holding a “strategy” in the demanding sense required by game theory.

To answer the difficulties presented above, we need to return to the problem of individual decision-making in a repeated situation. This process is depicted in [1].

The left-hand side of Figure [1] is an individual’s one-shot decision-making process in his own mind: He first forms some knowledge/beliefs about the situation; he then, through logical inference, reaches some decision and behaves according to it. The right-hand side shows the individual’s interaction with the social situation, which is the essential part of a repeated situation: Individuals’ behaviors determine the social situation, and each individual is influenced by that outcome. To be specific, one takes others’ actions, and tries to interpret or decipher them. Based on his interpretation, he modifies and updates his knowledge and beliefs, as well as adjusts his behavior; further, he will often choose his actions in a way to influence others. As the stage situation repeated, such a influence-inference cycle process also continues.
Here, two things are important. First, the starting point of an individual’s decision-making is his understanding of the stage situation, i.e., an individual needs to form some basic knowledge/beliefs of this situation. Therefore, we need first to analyze the stage situation, which constitutes the first part of our approach. In Section 3 we introduce several types of stage situations: the Matching-Pennies (MP)-type, the Battle-of-the-Sexes (BS)-type, and combined ones. In MP-type, each player aims to make his behavior unpredictable, while in BS, they aim at the opposite. Hence it is reasonable to expect that an individual’s decision-making is quite different in these two situations. The combined type situation will bear aspects of both MP- and BS-types, so that a player will aim to be partially, but not fully predictable.

Second, as mentioned before, we need to formulate an individual’s interpretation of others’ repeated behavior, as well as his own response to them, including any attempts to influence others. Repeated game theory tries to model this by defining a strategy as a function on histories, i.e., every possible sequence of action. As discussed before, this approach gives little space for understanding the process of deciphering other players in real social interaction. Our approach, as will be expounded in Section 4, is to model an individual as a Turing machine. We rely on the Turing-Church thesis in assuming that there is a match between Turing machines and human capacities – logical, inferential, computational, and, ultimately, behavioral. Actually, modeling a player as an automaton or a Turing machine is not a new idea in game theory (Abreu and Rubinstein, 1988; Ben-Porath, 1993; Hu, 2010). However, in our view, this literatures has not succeeded in delivering a full account of decision-making in repeated interaction.

In our approach, at each step, an individual’s choices at all previous steps are formulated as a sequence of symbols (e.g., a sequence of 0-1 in repeated MP). Such sequences of all individuals are an input for an individual (i.e., a Turing machine), who uses these as inputs to find patterns in others’ behavior. We study such a process through the lens of Kolmogorov complexity (Li and Vitányi, 2009), that is, the mathematical theory which aims to capture how hard it is to describe a sequences. By this approach, we can discuss both randomness of a mixed strategy, some special patterns, and interrelation between players’ actions. Based on his understanding of others, an individual determine and adjust his own actions.

We will show in Section 4 that, as an implication of the undecidability of the halting problem, there is no golden rule to guide an individual’s behavior in repeated situations. An individual needs to do some ad hoc inference. This does not imply that nothing interesting can be said about how player’s can and should behave in such situation. Rather, it means that there is no universal rule, no analogon of a Nash strategy to guarantee that a player always “wins”, or acquires his maximin value, or the like. Informally speaking, this means that decision-making in (a repeated or one-shot) situation is an art rather than a routine technology. This has, of course, long been known for practitioners of actual strategic decision-making, whether in a military, political, or business context.

3. Categories of Stage Situations

3.1. Predictable and unpredictable situations

In the following, we will refer to a “situation” or “stage situation” instead of the standard notion of a “game” – a set of players, strategies, and payoffs. A “situation” is obtained from a game by replacing the strategy set with an action set for each player. Thus, in a situation, an individual’s payoff is defined on the product set of actions rather than strategies. A repeated situation is repetition of some stage situation. For the most part, we will deal with 2-person situations. Most of the discussions here can be extended into n-person situations after some elaboration. We assume common knowledge of the stage situation.

We start with two paradigmatic situations, a zero-sum and a non-zero-sum situation. The following is the well-known situation of Matching Pennies:

When the Matching Pennies situation is repeated, it is in the interest of each player to (1.a) make his behavior unpredictable to the other, and (1.b) try, through his behavior, to make the other player believe that his behavior is, in fact, predictable. The

---

2 It should be noted that there are two main concepts of “complexity” in computer science. The perhaps more well-known is computational complexity, related to the number of steps required to carry out a computation. The other is descriptional complexity, that is, the hardness to describe something (e.g., a sequence of symbols). Kolmogorov complexity deals with the later. See (Garey and Johnson, 1979; Durand and Zvonkin, 2007).
first can be regarded as a principle of unpredictability, while the second is a principle of deceptive predictability. We call a situation where it is in the interest of each player to behave according to (1.a) and (1.b) an MP-type situation.

In contrast, consider the Battle of the Sexes:

<table>
<thead>
<tr>
<th>Player 1</th>
<th>a_{11}</th>
<th>a_{12}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player 2</td>
<td>a_{21}</td>
<td>a_{22}</td>
</tr>
<tr>
<td>a_{11}</td>
<td>1, −1</td>
<td>−1, 1</td>
</tr>
<tr>
<td>a_{12}</td>
<td>−1, 1</td>
<td>1, −1</td>
</tr>
</tbody>
</table>

Table 1: Matching pennies

<table>
<thead>
<tr>
<th>Player 1</th>
<th>a_{11}</th>
<th>a_{12}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player 2</td>
<td>a_{21}</td>
<td>a_{22}</td>
</tr>
<tr>
<td>a_{11}</td>
<td>2, 1</td>
<td>0, 0</td>
</tr>
<tr>
<td>a_{12}</td>
<td>0, 0</td>
<td>1, 2</td>
</tr>
</tbody>
</table>

Table 2: Battle of the sexes

This is non-zero-sum situation. While the interest of players is not perfectly aligned, this situation encourages coordinated behavior between participators, that is, either (a_{11}, a_{21}) or (a_{12}, a_{22}). In the Battle of the Sexes, it is in the interest of each player to (2.a) make his behavior predictable to the other, and (2.b) try, through his behavior, to make the other player believe that his own behavior, is, in fact, predictable. The first principle is that of predictability, while the second is a principle of truthful predictability. We call a situation where the goal of each player is to behave according to (2.a) and (2.b) an BS-type situation.

We have thus distilled four principles for generating actions in repeated situations, which we group in two according to the type of the situation.

1. Principle of unpredictability and deceptive predictability.
2. Principle of predictability and truthful predictability.

To see these criteria in operation, let 0 represent actions a_{11}/a_{21} and 1 represent a_{12}/a_{22} in each of the two situations presented above. Consider the following three 0-1 sequences (borrowed from Fortnow (2001)):

010101010101010101010101
110100110010110100101100
10011110111011001100100110

Each sequence represents a sequence of 24 actions of an individual in the repeated MP situation. The frequency of 0s and 1s is 12 for each one of these sequences (i.e., a relative frequency of 0.5). However, the first sequence follows a pattern - the individual switches between 0 and 1 at every period. Assuming this pattern can be detected by his opponent, his opponent can exploit this fact. The second sequence seems more complicated, but still follows a pattern: Position i is 1 if and only if there are an odd number of 1’s in the binary expansion of number i. The third sequence has been generated by random coin tosses. It has no pattern; the only way to describe it is to recite it. In Section 4, we will give a detailed account of this idea.

Behaving unpredictably requires choosing a sequence of actions like in the third sequence. Predictability would suggest a sequence such as the first one. The second sequence appears as in in-between case: it might or might not be predictable, depending on the computational abilities of one’s opponent.
3.2. Mixed-strategy equilibria

The standard answer on how to behave in situations such as MP makes use of the idea of a proper mixed-strategy Nash equilibrium. In the unique equilibrium, player 1 uses $a_{11}$ and $a_{12}$ with probability $\frac{1}{2}, \frac{1}{2}$ and player 2 uses $a_{21}$ and $a_{22}$ with probability $\frac{1}{2}, \frac{1}{2}$. Similarly, the BS situation has a mixed-strategy equilibrium with probabilities $\frac{1}{3}, \frac{2}{3}$ being used for $a_{11}$ and $a_{12}$ (and similarly for player 2).

However there are several issues raised by playing a proper mixed strategy.

- Behavioral evidence indicates we are rather ineffective in generating random numbers.
- In real-life situations, More often than not, we do not have access to any such device.
- Even if a device appears random, players would also have to make sure that the randomizing device has the desired ontological and statistical properties of randomness.
- It seems even when we do, in fact, have access to such devices (say, random number generators from the Internet), we do not seem to making much use of them.
- Even if we do have access to such a device, we have to make sure that the other party does not have access to the (sequence of) actions chosen by it.
- Suppose the random device generates the sequence 111111.... Even knowing that the device satisfied all criteria regarding randomizing devices, does that give a player sufficient reason actually play it?

Originally, von Neumann developed the notion of a mixed strategy to capture behavior aiming to be unpredictable (von Neumann, 1928; von Neumann and Morgenstern, 1947). Since von Neumann focused on zero-sum games (i.e, MP-type situations), a mixed-strategy equilibrium describing a balance between players’ unpredictable behaviors is compatible with the situation he intended to study. When the notion of mixed strategies was extended to nonzero-sum games, its roots in the principle of unpredictability were forgotten.

Thus, we argue that for stage games with mixed-strategy Nash equilibria, we should discard Nash equilibrium as the notion we appeal to for either descriptive or normative purposes. Instead, we should reinstate predictability and unpredictability (i.e., our principles 1 and 2 above), and develop a theory for social situations based on them.

3.3. Combined situations

There are situations which combine the aspects of both MP- and BS-type. Here are two examples:

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_{11}$</td>
<td>$a_{21}$</td>
</tr>
<tr>
<td>1, -1</td>
<td>-1, 1</td>
</tr>
<tr>
<td>-1, 1</td>
<td>1, -1</td>
</tr>
<tr>
<td>-9, -9</td>
<td>-9, -9</td>
</tr>
<tr>
<td>-9, -9</td>
<td>-9, -9</td>
</tr>
</tbody>
</table>

Table 3: Matching pennies in the boxing house/opera.

The situation is a combination of MP- and BS-type. It encourages participators to cooperate to prevent any outcome in $\{a_{13}, a_{14}\} \times \{a_{21}, a_{22}\}$ and $\{a_{11}, a_{12}\} \times \{a_{23}, a_{24}\}$, while it requires unpredictable behavior on the remaining choices. Therefore,
it is reasonable for an individual to adopt a combined criterion here: on one hand, he need to make sure that he does not reach the \((-9, -9)\) areas, that is, shows some predictability; on the other hand, having reached the upper left/lower right blocks, he needs to be unpredictable.

In this game, if Players 1 and 2 successfully coordinate, they win 2 from Player 3, which they split equally. In addition, Player 3 guesses where Player 1 and 2 will coordinate. If Player 3 guesses incorrectly, Players 1 and 2 gain an additional 2.

Let us take Player 1’s viewpoint. Here, he needs to coordinate with player 2, but (together with player 2), needs to make their locus of coordination is unpredictable to player 3. This situation requires a more delicate behavior. If player 1 can not prevent players 2 and 3 receive the same information about their behavior each round, can he still choose a behavior which is predictable to player 2, while unpredictable to player 3?

Understanding the stage situation is the starting point of an individual’s decision-making in a repeated situation. Based on two basic principles of behavior criteria, we introduced three types of situations: MP-type, BS-type, and combined type. In next section, we will move the proper problem of predictability and behavior adjustment work in repeated situations.

4. Understanding repeated interactions through Kolmogorov complexity

In this section, we formulate learning, signaling, and behavior adjustment in repeated situations. We first will give a survey of the preliminaries in the theory of Turing machines, and show how to model each individual as a Turing machine. At each step, an individual’s choices at all previous steps are formulated as a sequence of symbols (in particular a 0–1-sequence). Such sequences of all other players are the input for player (i.e., a Turing machine), who uses this input to decipher the patterns or rules governing others’ behavior. We show that, as an implication of the undecidability of the halting problem, that there is no universal golden rule to guide an individual’s behavior. Then we will give a survey of the preliminaries of Kolmogorov complexity, which allows us to study the process of deciphering the other’s behavior. We discuss both randomness of symbol sequences (i.e., a sequence of actions), patterns, and interrelation between players’ actions. We finally specify how an individual makes decisions and adjusts his behavior based on his understanding (i.e., deciphering of a Turing machine).

4.1. Prelimiaries

4.1.1. Binary strings

Our alphabet will consist of the set \(\mathcal{B} = \{0, 1\}\). The set of finite strings (or, equivalently, sequences) over this alphabet is denoted by \(\mathcal{B}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}\). The concatenation of two strings \(x, y \in \mathcal{B}^*\) is just the string obtained by writing \(x\) first, and \(y\) after. As a convention, \(0^k = 00\ldots0\) and \(1^k = 11\ldots1\). A language based on our alphabet is defined as some \(L \subseteq \mathcal{B}^*\).

In the following, we denote the cardinality of a set by \(|X|\).

Note that we can easily define a bijective function \(f\) between the natural numbers \(\mathbb{N}\) and the set of strings \(\mathcal{B}^*\) through:

\[
f(0) = \epsilon, f(1) = 0, f(2) = 1, f(3) = 00, f(4) = 01, \ldots
\]
Consider a string $x \in \mathcal{B}^*$. The length of $x$ is denoted by $l(x)$. We define the prefix-free encoding $\overline{x}$ of $x$ as follows:

$$\overline{x} = 1^{l(x)}0x$$

Note that $l(\overline{x}) = 2l(x) + 1$. The idea behind prefix-free encoding is that by concatenating two prefix-free encoded strings, we will be able to tell where the first string ends, and where the second one begins. Consider for instance the strings $x = 01$ and $y = 110$. By concatenating them, we get $xy = 01110$. But by reading 01110, even knowing it encodes two strings, we are unable to tell whether $x = \epsilon$ and $y = 01110$, or $x = 0$ and $y = 1110$ or so forth. However, by taking $\overline{x} = 11001$, and $\overline{y} = 1110110$, and then concatenating them: $\overline{x}\overline{y} = 11001110110$, we can get back $x$ and $y$, knowing the prefix-free encoding.

Following this idea, it is easy to see that the iterated concatenation of any number of prefix-free strings allow us to identify and decode them, knowing the prefix-free encoding defined above. Also, note that shorter prefix-free encodings also exist, which achieve the same purpose. Sometimes we will refer to a prefix-free encoding more simply as just an encoding.

4.1.2. The stage game

The stage situation, as we mentioned in Section 3 is defined as:

$$\Gamma = \langle N, (A_i)_{i \in N}, (u_i)_{i \in N} \rangle$$

where

(1) $N = \{1, 2\}$, the set of players;

(2) $A_i$ a finite set of actions for player $i \in N$. For simplicity, we consider an (arbitrary) ordering of strategies within each set, so we denote the elements of $S_i$ by $a_{i1}, a_{i2}, \ldots$. We let $A = \Pi_{i \in N} A_i$, the set of strategy profiles. This can also be ordered lexicographically, for example, for the two-player case $A = \{(a_{11}, a_{21}), (a_{11}, a_{22}), \ldots, (a_{21}, a_{22}), \ldots\}$;

(3) $u_i : S \rightarrow \mathbb{Q}$, the payoff functions for all players $i \in N$.

Two important example of stage situations are MP situations and BS situations, as shown in Section 3.

4.1.3. Encoding games

The repeated situation $\Gamma_T$ consists of the stage situation $\Gamma$ played sequentially $T \geq 1$ times. We will encode the repeated game $\Gamma_T$ defined above as a string. To do that, we need map each element of $\Gamma_T$ to a string, take their prefix-free encoding, and concatenate these encodings. Note that $|\mathbb{Q}| = |\mathbb{N}|$, i.e., there exists a bijection $g : \mathbb{Q} \leftrightarrow \mathbb{N}$. We will use an arbitrary such $g$, as well as, $f$, the bijection between strings and natural numbers. Thus, we set:

(1) $x_T = f(T)$;

(2) $x_n = f(|N|)$. For two-player-games, as in our example, $x_n = 1$;

(3) $(x_s)_{s \in N} = f(|S_i|)$. The encodings of the sizes of the strategy sets for each player will be concatenated by player order;

(4) $x_{u(s)} = g(u_i(s))$, for $s \in S$. The encoding of the payoffs will be concatenated in the lexicographic order of the strategy profiles mentioned above, and then by player order.

Based on the above, we can set:

$$x_{\Gamma_T} = \overline{x_T}, \overline{x_n}, x_{s_1}, \ldots, x_{s_{|N|}}, x_{u_1(s_1)}, \ldots, x_{u_2(s_{|S_2|})}, \ldots$$

This achieves the desired encoding of a finitely repeated simultaneous-move game through a string. Given such a string, the game this string can be reconstructed by first finding out the number of periods from $x_T$, the number players from $x_n$, the

---

1Specifically, we read until we reach the first 0, and count how many 1s we find. This gives us the length of $x$. Then, we read 01 for $x$. We then continue to the next 0, allowing us to know the length of $y$. Finally, we read 110 for $y$. 

10
number of available strategies for each player, and finally, the payoffs of all strategy profiles for each player.\footnote{\textsuperscript{4}}

Note that many strings will not encode any games. For example, the string $x = 111$ will not encode a game. However, it is easy to construct a procedure for deciding whether a string encodes a game or not.\footnote{\textsuperscript{5}} Given such a procedure, one could enumerate the elements of $B^*$, and check for each string whether it encodes a game. Within this enumeration, the first string encoding a game can be associated with the number 0, the second such string with number 1, and so forth. This creates a bijection between set of possible games and $\mathbb{N}$. Compounding this with the bijection $f$, we have established a bijection between the set of strings, and finitely repeated simultaneous-move games with rational payoffs.

### 4.2. Turing machines

Time is assumed to be discrete. The tape is a linear list of cells, infinite on both ends. Each cell can contain a 0, a 1, or a ‘blank symbol’ $B$. The tape has a head, which, at any point of time, is located over one of the cells (the ‘scanned’ cell). The head can move towards either end of the tape (‘left’ and ‘right’). At period 0, the head is over a particular cell (‘start cell’). At time 0 all cells contain Bs, except for a contiguous finite sequence of cells, extending from the start cell to the right, which contain 0s and 1s. This binary sequence is called the input. Note that the input can contain several strings if they are self-delimiting. At any point in time, the head can take one of the following actions: write one of $\{0, 1, B\}$ on the current cell, and move to the right/left. The set of possible actions is $A = \{0, 1, B, L, R\}$.

A Turing machine consists of:

1. A set of states $Q$, in which the machine can be in. One of these state is a ‘start state’ $q_0$;
2. A number of rules of the form: if the current state is $p$, and the symbol under the head is $s$, then take action $a$, and change the state to $q$. The rules of the Turing Machine thus have the form $(p,s,a,q)$.

A Turing machine is thus defined by a mapping $T : Q \times S \rightarrow A \times Q$.

Any two distinct quadruples cannot have their first two elements identical: the device is deterministic.

Not every possible combination of states and actions has to be a rule. If there is no rule associated to a particular state-symbol combination, we say that the device ‘halts’. The ‘output’ is the longest 0−1 sequence bordered by blanks starting from the symbol being scanned when it halts (or 0 if a blank is scanned). A Turing machine carries out a definite sequence of steps, and it may or may not halt.

As a first step, we need a binary code for the states and possible actions. This allows us to encode to encode $\overline{q}, \overline{s}, \overline{a}$. A Turing machine can thus be defined by the number of states, the number of bits and the rules. By using the encoding $\overline{x}$, for a given Turing machine we can define:

$$\overline{T} = \overline{b} \overline{r} \overline{q_1,Start} \overline{s_1} \overline{a_1} \overline{q_1,End} \ldots \overline{q_r,Start} \overline{s_r} \overline{a_r} \overline{q_r,End}$$

Intuitively, $b$ gives the number of bits used to encode states/actions, $r$ gives number of rules, and then the rules are described successively. We demand that the rules are lexicographically ordered by $q_1,Start, q_1,End$. By encoding Turing machines as a binary string, we created a mapping from the set of Turing machines to the set of sequences. This mapping can be made one-to-one (consider all Turing machines. Assign 0 the one which comes first lexicographically, 1 to the second, …). Call this the Gödel number of a Turing machine.

We could create a machine that starts listing the sequences lexicographically, and checks for each one of them whether it corresponds to a Turing machine.

A universal Turing machine $U$ is a Turing machine that can imitate the behavior of any other Turing machine $T$. It takes two inputs: a number $k$, corresponding to the number of the Turing machine to be imitated, and an input $y$. It then goes on...
to imitate the operation of Turing machine $k$ on input $y$. Universal Turing machines can be constructed effectively. Universal Turing machines can be surprisingly simple.

4.2.1. The halting problem

Turing showed the following statement:

**Theorem 4.1 (Halting Problem).** There is no Turing machine $g$ such that for all $k, y$ we have $g(k, y) = 1$ if $T_k$ stops on input $y$, and $g(k, y) = 0$ otherwise.

**Proof.** Suppose there exists such a $g$. Then, construct the Turing machine $\psi$ by $\psi(x) = 1$ if $g(x, x) = 0$, and $\psi(x)$ does not stop otherwise. Let $\psi$ have Gödel number $y$, so $\psi = T_y$. Then, $T_y(y) = \psi(y)$ stops if and only if $g(y, y) = 0$, according to $\psi$’s definition. However, $g(y, y) = 0$ if $T_y(y)$ does not stop, according to $g$’s definition. This is a contradiction.

4.3. Kolmogorov complexity

Let $T$ be a (one-argument) Turing machine, and $x$ a sequence. The complexity of $x$ with respect to $T$ is defined by:

$$C_T(x) = \min \{l(p) : T(p) = x\}$$

$C_T(x) = \infty$ if there exists no such $p$.

It would seem that this makes complexity depend on $T$. However, we can show that there exists a $U$ Universal Turing machine such that for every $T$, there exists a $c$ such that

$$C(x) = C_U(x) = C_T(x) + c$$

A sequence is called compressible if $C(x) \ll l(x)$. It is incompressible otherwise. Note that most strings will be incompressible.

References


