Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Sequential Equilibrium in Computational Games

Joseph Y. Halpern and **Rafael Pass***
Cornell University
halpern|rafael@cs.cornell.edu

Abstract

We examine sequential equilibrium in the context of *computational games* [Halpern and Pass, 2011a], where agents are charged for computation. In such games, an agent can rationally choose to forget, so issues of imperfect recall arise. In this setting, we consider two notions of sequential equilibrium. One is an *ex ante* notion, where a player chooses his strategy before the game starts and is committed to it, but chooses it in such a way that it remains optimal even off the equilibrium path. The second is an *interim* notion, where a player can reconsider at each information set whether he is doing the "right" thing, and if not, can change his strategy. The two notions agree in games of perfect recall, but not in games of imperfect recall. Although the interim notion seems more appealing, in [Halpern and Pass, 2011b] it is argued that there are some deep conceptual problems with it in standard games of imperfect recall. We show that the conceptual problems largely disappear in the computational setting. Moreover, in this setting, under natural assumptions, the two notions coincide.

1 Introduction

In [Halpern and Pass, 2011a], we introduced a framework to capture the idea that doing costly computation affects an agent's utility in a game. The approach, a generalization of an approach taken in [Rubinstein, 1986], assumes that players choose a Turing machine (TM) to play for them. We consider Bayesian games, where each player has a *type*) (i.e., some private information); a player's type is viewed as the input to his TM. Associated with each TM M and input (type) t is its *complexity*. The complexity could represent the running time of or space used by M on input t. While this is perhaps the most natural interpretation of complexity, it could have other interpretations as well. For example, it can be used to capture the complexity of M itself (e.g., the number of states in M, which is essentially the complexity measure considered by Rubinstein, who assumed that players choose a finite automaton to play for them rather than a TM) or to model the cost of searching for a better strategy (so that there is no cost for using a particular TM M, intuitively, the strategy that the player has been using for years, but there is a cost to switching to a different TM M'). A player's utility depends both on the actions chosen by all the players' machines and the complexity of these machines.

This framework allows us to consider the tradeoff in a game like *Jeopardy* between choosing a strategy that spends longer thinking before pressing the buzzer and one that answers quickly but is more likely to be incorrect. Note that if we take "complexity" here to be running time, an agent's utility depends not only on the complexity of the TM that he chooses, but also on the complexity of the TMs chosen by other players. We defined a straightforward extension of Bayesian-Nash equilibrium in such machine games, and showed that it captured a number of phenomena of interest.

Although in Bayesian games players make only one move, a player's TM is doing some computation during the game. This means that solution concepts more traditionally associated with extensive-form games, specifically, *sequential equilibrium* [Kreps and Wilson, 1982], also turn out to be of interest, since we can ask whether an agent wants to switch to a different TM during the computation of the TM that he has chosen (even at points off the equilibrium path). We can certainly imagine that, at the beginning of the computation, an agent may have decided to invest in doing a lot of computation, but part-way through the computation, he may have already learned enough to realize that further computation is unnecessary. In a sequential equilibrium, intuitively, the TM he chose should already reflect this. It turns out that, even in this relatively simple setting, there are a number of subtleties.

The "moves" of the game that we consider are the outputs of the TM. But what are the information sets? We take them to be determined by the states of the TM. While this is a natural interpretation, since we can view the TM's state as characterizing the knowledge of the TM, it means that the information sets of the game are not given exogenously, as is

*Halpern supported in part by NSF grants IIS-0812045, IIS-0911036, and CCF-1214844, AFOSR grants FA9550-08-1-0266 and FA9550-12-1-0040, and ARO grant W911NF-09-1-0281. Pass supported in part by an Alfred P. Sloan Fellowship, a Microsoft Research Faculty Fellowship, NSF Awards CNS-1217821 and CCF-1214844, NSF CAREER Award CCF-0746990, AFOSR YIP Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2-0211. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

standard in game theory; rather, they are determined endogenously by the TM chosen by the agent.[1] Moreover, in general, the game is one of imperfect recall. An agent can quite rationally choose to forget (by choosing a TM with fewer states, that is thus not encoding the whole history) if there is a cost to remembering. There are well-known subtleties in defining sequential equilibrium in the presence of imperfect recall (see, for example, [Piccione and Rubinstein, 1997]). We give a definition of sequential equilibrium in a companion paper [Halpern and Pass, 2011b] for standard games of imperfect recall that we extend here to take computation into account. We show that, in general, sequential equilibrium does not exist, but give simple conditions that guarantee that it exists, as long as a NE (Nash equilibrium) exists. (As is shown by a simple example in [Halpern and Pass, 2011a], reviewed below, NE is not guaranteed to exist in machine games, although sufficient conditions are given to guarantee existence.)

The definition of sequential equilibrium in [Halpern and Pass, 2011b] views sequential equilibrium as an *ex ante* notion. The idea is that a player chooses his strategy before the game starts and is committed to it, but he chooses it in such a way that it remains optimal even off the equilibrium path. This, unfortunately, does *not* correspond to the more standard intuitions behind sequential equilibrium, where players are reconsidering at each information set whether they are doing the "right" thing, and if not, can change their strategies. This *interim* notion of sequential rationality agrees with the ex ante notion in games of perfect recall, but the two notions differ in games of imperfect recall. We argue in [Halpern and Pass, 2011b] there are some deep conceptual problems with the interim notion in standard games of imperfect recall. We consider both an *ex ante* and interim notion of sequential equilibrium here. We show that the conceptual problems when the game tree is given (as it is in standard games) largely disappear when the game tree (and, in particular, the information sets) are determined by the TM chosen, as is the case in machine games. Moreover, we show that, under natural assumptions regarding the complexity function, the two notions coincide.

2 Computational games: a review

This review is largely taken from [Halpern and Pass, 2011a]. We model costly computation using *Bayesian machine games*. Formally, a Bayesian machine game is given by a tuple $([m], \mathcal{M}, T, \Pr, \mathscr{C}_1, \ldots, \mathscr{C}_m, u_1, \ldots, u_m)$, where

- $[m] = \{1, \ldots, m\}$ is the set of players;
- $\mathcal{M}$ is a set of TMs;
- $T \subseteq (\{0,1\}^*)^m$ is the set of *type profiles* (m-tuples consisting of one type for each of the m players);
- $\Pr$ is a distribution on T;
- $\mathscr{C}_i$ is a *complexity function* (see below);
- $u_i : T \times (\{0,1\}^*)^m \times I\!N^m \rightarrow I\!R$ is player i's utility function. Intuitively, $u_i(\vec{t}, \vec{a}, \vec{c})$ is the utility of player i if $\vec{t}$ is the type profile, $\vec{a}$ is the action profile (where we identify i's action with M_i's output), and $\vec{c}$ is the profile of machine complexities.

If we ignore the complexity function and drop the requirement that an agent's type is in $\{0,1\}^*$, then we have the standard definition of a Bayesian game. We assume that TMs take as input strings of 0s and 1s and output strings of 0s and 1s. Thus, we assume that both types and actions can be represented as elements of $\{0,1\}^*$. We allow machines to randomize, so given a type as input, we actually get a distribution over strings. To capture this, we take the input to a TM to be not only a type, but also a string chosen with uniform probability from $\{0,1\}^\infty$ (which we view as the outcome of an infinite sequence of coin tosses). The TM's output is then a deterministic function of its type and the infinite random string. We use the convention that the output of a machine that does not terminate is a fixed special symbol ω. We define a *view* to be a pair (t, r) of two bitstrings; we think of t as that part of the type that is read, and of r as the string of random bits used. A complexity function $\mathscr{C} : \mathbf{M} \times \{0,1\}^*; (\{0,1\}^* \cup \{0,1\}^\infty) \rightarrow I\!N$, where $\mathbf{M}$ denotes the set of Turing machines, gives the complexity of a (TM,view) pair.

We can now define player i's *expected utility* $U_i(\vec{M})$ if a profile $\vec{M}$ of TMs is played; we omit the standard details here. We then define a (computational) NE of a machine game in the usual way:

Definition 2.1 *Given a Bayesian machine game* $G = ([m], \mathcal{M}, T, \Pr, \vec{\mathscr{C}}, \vec{u})$*, a machine profile* $\vec{M} \in \mathcal{M}^m$ *is a* (computational) Nash equilibrium *if, for all players* i*,* $U_i(\vec{M}) \geq U_i(M_i', \vec{M}_{-i})$ *for all TMs* $M_i' \in \mathcal{M}$*.*

Although a NE always exists in standard games, a computational NE may not exist in machine games, as shown by the following example, taken from [Halpern and Pass, 2011a].

Example 2.2 Consider rock-paper-scissors. As usual, rock beats scissors, scissors beats paper, and paper beats rock. A player gets a payoff of 1 if he wins, -1 if he loses, and 0 if it is a draw. But now there is a twist: since randomizing is cognitively difficult, we charge players $\epsilon > 0$ for using a randomized strategy (but do not charge for using a deterministic strategy). Thus, a player's payoff is $1 - \epsilon$ if he beats the other player but uses a randomized strategy. It is easy to see that every strategy has a deterministic best response (namely, playing a best response to whatever move of the other player has the highest probability); this is a strict best response, since we now charge for randomizing. It follows that, in any equilibrium, both players must play deterministic strategies (otherwise they would have a profitable deviation). But there is clearly no equilibrium where players use deterministic strategies. ∎

Interestingly, it can also be shown that, in a precise sense, if there is no cost for randomization, then a computational NE is guaranteed to exist in a computable Bayesian machine game (i.e., one where all the relevant probabilities are computable); see [Halpern and Pass, 2011a] for details.

3 Computation as an extensive-form game

Recall that a deterministic TM $M = (\tau, Q, q_0, \mathcal{H})$ consists of a read-only input tape, a write-only output tape, a read-write

[1] We could instead consider a "supergame", where at the first step the agent chooses a TM, and then the TM plays for the agent. In this supergame, the information can be viewed as exogenous, but this seems to us a less natural model.

work tape, 3 machine heads (one reading each tape), a set Q of machine states, a *transition function* $\tau : Q \times \{0,1,b\}^2 \rightarrow Q \times \{0,1,b\}^2 \times \{L,R,S\}^3$, an initial state $q_0 \in Q$, and a set $\mathcal{H} \subseteq Q$ of "halt" states. We assume that all the tapes are infinite, and that only 0s, 1s and blanks (denoted b) are written on the tapes. We think of the input to a TM as a string in $\{0,1\}^*$ followed by blanks. Intuitively, the transition function says what the TM will do if it is in a state s and reads i on the input tape and j on the work tape. Specifically, τ describes what the new state is, what symbol is written on the work tape and the output tape, and which way each of the heads moves (L for one step left, R for one step right, or S for staying in the same place). The TM starts in state q_0, with the input written on the input tape, the other tapes blank, the input head at the beginning of the input, and the other heads at some canonical position on the tapes. The TM then continues computing according to τ. The computation ends if and when the machine reaches a halt state $q \in \mathcal{H}$. To simplify the presentation of our results, we restrict attention to TMs that include only states $q \in Q$ that can be reached from q_0 on some input. We also consider *randomized* TMs, which are identical to deterministic TMs except that the transition function τ now maps $Q \times \{0,1,b\}^2$ to a probability distribution over $Q \times \{0,1,b\}^2 \times \{L,R,S\}^3$. As is standard in the computer science literature, we restrict attention to probability distributions that can be generated by tossing a fair coin; that is, the probability of all outcomes has the form $c/2^k$ for $c,k \in I\!N$.

In a standard extensive-form game, a strategy is a function from information sets to actions. Intuitively, in a state s in an information set I for player i, the states in I are the ones that i considers possible, given his information at s; moreover, at all the states in I, i has the same information. In the "runs-and-systems" framework of [Fagin *et al.*, 1995], each agent is in some *local state* at each point in time. A *protocol* for agent i is a function from i's local states to actions. We can associate with a local state ℓ for agent i all the histories of computation that end in that local state; this can be thought of as the information set associated with ℓ. With this identification, a protocol can also be viewed as a function from information sets to actions, just like a strategy.

In our setting, the closest analogue to a local state in the runs-and-systems framework is the state of the TM. Intuitively, the TM's state describes what the TM knows. The transition function τ of the TM determines a protocol, that is, a function from the TM's state to a "generalized action", consisting of reading the symbols on its read and work tapes, then moving to a new state and writing some symbols on the write and work tapes (perhaps depending on what was read). We can associate with a state q of player i's TM the information set I_q consisting of all histories h where player i's is in state q at the end of the history. (Here, a *history* is just a sequence of *extended state profiles*, consisting of one extended state for each player, and an *extended state* for player i consists of the TM that i is using, the TM's state, and the content and head position of each of i's tapes.) Thus, player i implicitly *chooses* his information sets (by choosing the TM), rather than the information set being given exogenously.

The extensive-form game defined by computation in a Bayesian machine game is really just a collection of m single-agent decision problems. Of course, sequential equilibrium becomes even more interesting if we consider computational extensive-form games, where there is computation going on during the game, and we allow for interaction between the agents. We discuss sequential equilibrium in (computational) extensive-form games in the full paper, which also includes all proofs.[2] It turns out that many of the issues of interest already arise in our setting.

4 Beliefs

Using the view of machine games as extensive-form games, we can define sequential equilibrium. The first step to doing so involves defining a player's beliefs at an information set. But now we have to take into account that the information set is determined by the TM chosen. In the spirit of [Kreps and Wilson, 1982], define a *belief system* μ for a game G to be a function that associates with each player i, TM M_i for player i, and a state q for M_i a probability on the histories in the information set I_q. Following [Halpern and Pass, 2011b], we interpret $\mu_{q,M_i}(x)$ as the probability of going through history x conditional on reaching the local state q.

We do not expect $\sum_{x \in I_q} \mu_{q,M_i}(x)$ to be 1; in general, it is greater than 1. This point is perhaps best explained in the context of games of imperfect recall. Let the *upper frontier* of an information set X in a game of imperfect recall, denoted $\hat{X}$, consist of all those histories $h \in X$ such that there is no history $h' \in X$ that is a prefix of h. In [Halpern and Pass, 2011b], we consider belief systems that associate with each information set X a probability μ_X on the histories in X. Again, we do not require $\sum_{h \in X} \mu_X(h) = 1$. For example, if all the histories in X are prefixes of same complete history h^*, then we might have $\mu_X(h) = 1$ for all histories $h \in X$. However, we do require that $\Sigma_{h \in \hat{X}} \mu_X(h) = 1$. We make the analogous requirement here. Let $\hat{I}_q$ denote the upper frontier of I_q. The following lemma, essentially already proved in [Halpern and Pass, 2011b], justifies the requirement.

Lemma 4.1 *If q is a local state for player i that is reached by $\vec{M}$ with positive probability, and $\mu'_q(h)$ is the probability of going through history h when running $\vec{M}$ conditional on reaching q, then $\sum_{h \in \hat{I}_q} \mu'_q(h) = 1$.*

Given a belief system μ and a machine profile $\vec{M}$, define a probability distribution $\mu_q^{\vec{M}}$ over terminal histories in the obvious way: for each terminal history z, let h_z be the history in $\hat{I}_q$ generated by $\vec{M}$ that is a prefix of z if there is one (there is clearly at most one), and define $\mu_q^{\vec{M}}(z)$ as the product of $\mu_{q,M_i}(h_z)$ and the probability that $\vec{M}$ leads to the terminal history z when started in h_z; if there is no prefix of z in I_q, then $\mu_q^{\vec{M}}(z) = 0$. Following [Kreps and Wilson, 1982], let $U_i(\vec{M} \mid q, \mu)$ denote the expected utility for player i, where the expectation is taken with respect to $\mu_q^{\vec{M}}$. Note that utility is well-defined, since a terminal history determines both the input and the random-coin flips of each player's TM, and thus determines both its output and complexity.

[2] See http://www.cs.cornell.edu/home/halpern/papers/algseqeq.pdf.

5 Defining sequential equilibrium

If q is a state of TM M_i, we want to capture the intuition that a TM M_i for player i is a best response to a machine profile $\vec{M}_{-i}$ for the remaining players at q (given beliefs μ). Roughly speaking, we capture this by requiring that the expected utility of using another TM M_i' starting at a node where the TM's state is q to be no greater than that of using M_i. "Using a TM M_i' starting from q" means using the TM (M_i, q, M_i'), which, roughly speaking, is the TM that runs like M_i up to q, and then runs like M_i'.[3]

In the standard setting, all the subtleties in the definition of sequential equilibrium involve dealing with what happens at information sets that are reached with probability 0. When we consider machine games, as we shall see, under some reasonable assumptions, all states are reached with positive probability in equilibrium, so dealing with probability 0 is not a major concern (and, in any case, the same techniques that are used in the standard setting can be applied). But there are several new issues that must be addressed in making precise what it means to "switch" from M_i to M_i' at q.

Given a TM $M_i = (\tau, Q, q_0, \mathcal{H})$ and $q \in Q$, let Q_{q,M_i} consist of all states $q' \in Q$ such that, for all views v, if the computation of M_i given view v reaches q', then it also reaches q. We can think of Q_{q,M_i} as consisting of the states q' that are "necessarily" below q, given that M_i is used. Note that $q \in Q_{q,M_i}$. Say that $M_i' = (\tau', Q', q', \mathcal{H}')$ is *compatible with* M_i *given* q if $q' = q$ (so that q is the start state of M_i') and τ and τ' agree on all states in $Q - Q_{q,M_i}$. If M_i' is not compatible with M_i given q, then (M_i, q, M_i') is not well defined. If M_i' is compatible with M_i given q, then (M_i, q, M_i') is the TM $(Q'', [\tau, q, \tau'], q_0, \mathcal{H}'')$, where $Q'' = (Q - Q_{q,M_i}) \cup Q'$; $[\tau, q, \tau']$ is the transition function that agrees with τ on states in $Q - Q_{q,M_i}$, and agrees with τ' on the remaining states; and $\mathcal{H}'' = (\mathcal{H} \cap (Q - Q_{q,M_i})) \cup \mathcal{H}'$.

Since (M_i, q, M_i') is a TM, its complexity given a view is well defined. In general, the complexity of (M_i, q, M_i') may be different from that of M_i even on histories that do not go through q. For example, consider a one-person decision problem where the agent has an input (i.e., type) of either 0 or 1. Consider four TMs: M, M_0, M_1, and M^*. Suppose that M embodies a simple heuristic that works well on both inputs, M_0 and M_1 give better results than M if the inputs are 0 and 1, respectively, and M^* acts like M_0 if the input is 0 and like M_1 if the input is 1. Clearly, if we do not take computational costs into account, M^* is a better choice than M; however, suppose that with computational costs considered, M is better than M_0, M_1, and M^*. Specifically, suppose that M_0, M_1, and M^* all use more states than M, and the complexity function charges for the number of states used. Now suppose that the agent moves to state q if he gets an input of 0. In state q, using M_0 is better than continuing with M: the extra charge for complexity is outweighed by the improvement in performance. Should we say that using M is then not a sequential equilibrium? The TM $(M, \{q\}, M_0)$ acts just like M_0 if the input is 0. From the *ex ante* point of view, M is a better choice than M_0. However, having reached q, the agent arguably does not care about the complexity of M_0 on input 1. Our definition of *ex ante* sequential equilibrium restricts the agent to making changes that leave unchanged the complexity of paths that do not go through q (and thus would not allow a change to M_0 at q). Our definition of *interim* sequential equilibrium does not make this restriction; this is the only way that the two definitions differ. Note that this makes it easier for a strategy to be an ex ante sequential equilibrium (since fewer deviations are considered).

(M_i, q, M_i') is a *local variant of* M_i if the complexity of (M_i, q, M_i') is the same as that of M_i on views that do not go through q; that is, if for every view v such that the computation of $M_i(v)$ does not reach q, $\mathscr{C}(M_i, v) = \mathscr{C}((M_i, q, M_i'), v)$. A complexity function $\mathscr{C}$ is *local* if $\mathscr{C}(M_i, v) = \mathscr{C}((M_i, q, M_i'), v)$ for all TMs M_i and M_i', states q, and views v that do not reach q. Clearly a complexity function that considers only running time, space used, and the transitions undertaken on a path is local. If $\mathscr{C}$ also takes the number of states into account, then it is local as long as M_i and (M_i, q, M_i') have the same number of states. Indeed, if we think of the state space as "hardware" and the transition function as the "software" of a TM, then restricting to changes M_i' that have the same state space as M_i seems reasonable: when the agent contemplates making a change at a non-initial state, he cannot acquire new hardware, so he must work with his current hardware.

A TM $M_i = (\tau, Q, q_0, \mathcal{H})$ for player i is *completely mixed* if, for all states $q \in Q - \mathcal{H}$, $q' \in Q$, and bits $k, k' \in \{0, 1\}$, $\tau(q, k, k')$ assigns positive probability to making a transition to q'. A machine profile $\vec{M}$ is completely mixed if for each player i, M_i is completely mixed. Following [Kreps and Wilson, 1982], we would like to say that a belief system μ is *compatible* with a machine profile $\vec{M}$ if there exists a sequence of completely-mixed machine profiles $\vec{M}^1, \vec{M}^2, \ldots$ converging to $\vec{M}$ such that if q is a local state for player i that is reached with positive probability by $\vec{M}$ (that is, there is a type profile $\vec{t}$ that has positive probability according to the type distribution in G and a profile $\vec{r}$ of random strings such that $\vec{M}(\vec{t}, \vec{r})$ reaches q), then $\mu_{q,M_i}(h)$ is just the probability of $\vec{M}$ going through h conditional on $\vec{M}$ reaching q (denoted $\pi_{\vec{M}}(h \mid q)$); and if q is a local state that is reached with probability 0 by $\vec{M}$, then $\mu_{q,M_i}(h)$ is $\lim_{n\to\infty} \pi_{\vec{M}^n}(h \mid q)$. To make this precise, we have to define convergence. We say that $\vec{M}^1, \vec{M}^2, \ldots$ *converges to* $\vec{M}$ if, for each player i, all the TMs $M_i^1, M_i^2, \ldots, M_i$ have the same state space, and the transition function of each TM in the sequence to converge to that of M_i. Note that we place no requirement on the complexity functions. We could require the complexity function of M_i^k to converge to that of M_i in some reasonable sense. However, this seems to us unreasonable. For example, if we have a complexity function that charges for randomization, as in Example 2.2, then the complexity functions of M_i^n may not converge to the complexity function of M_i. Thus, if we require the complexity functions to converge, there will not be a sequence of completely mixed strategy profiles converging

[3] We remark that, in the definition of sequential equilibrium in [Halpern and Pass, 2011b], the agent was allowed to change strategy not just at a single information set, but at a collection of information sets. Allowing changes at a set of information sets seems reasonable for an ex ante notion of sequential equilibrium, but not for an interim notion; thus, we consider changes only at a single state here.

to a deterministic strategy profile $\vec{M}$. If we think of the sequence of TMs as arising from "trembles" in the operation of some fixed TM (e.g., due to machine failure), then requiring that the complexity functions converge seems unreasonable.

Definition 5.1 *A pair $(\vec{M}, \mu)$ consisting of a machine profile $\vec{M}$ and a belief system μ is called a* belief assessment*. A belief assessment $(\vec{M}, \mu)$ is an* interim sequential equilibrium *(resp.,* ex ante sequential equilibrium*) in a machine game $G = ([m], \mathcal{M}, \ldots)$ if μ is compatible with $\vec{M}$ and for all players i, states q of M_i, and TMs M'_i compatible with M_i and q such that $(M_i, q, M'_i) \in \mathcal{M}$ (resp., and (M_i, q, M'_i) is a local variant of M_i), we have*

$$U_i(\vec{M} \mid q, \mu) \geq U_i(((M_i, q, M'_i), \vec{M}_{-i}) \mid q, \mu)).$$

In Definition 5.1 we consider only switches (M_i, q, M'_i) that result in a TM that is in the set $\mathcal{M}$ of possible TMs in the game. That is, we require that the TM we switch to is "legal", and has a well-defined complexity.

As we said, upon reaching a state q, an agent may well want to switch to a TM (M_i, q, M'_i) that is not a local variant of M_i. This is why we drop this requirement in the definition of interim sequential equilibrium. But it is a reasonable requirement ex ante. It means that, at the *planning stage of the game*, there is no TM M'_i and state q such that agent i prefers to use M'_i in the event that q is reached. That is, M_i is "optimal" at the planning stage, even if the agent considers the possibility of reaching states that are off the equilibrium path.

The following result is immediate from the definitions, and shows that, in many cases of interest, the two notions of sequential equilibrium coincide.

Proposition 5.2 *Every interim sequential equilibrium is an ex ante sequential equilibrium. In a machine game with a local complexity function, the interim and ex ante sequential equilibria coincide.*

6 Relating Nash equilibrium and sequential equilibrium

In this section, we relate NE and sequential equilibrium.

First note that is easy to see that every (computational) sequential equilibrium is a NE, since if $q = q_0$, the start state, $(M_i, q_0, M'_i) = M'_i$. That is, by taking $q = q_0$, we can consider arbitrary modifications of the TM M_i.

Proposition 6.1 *Every ex ante sequential equilibrium is a NE.*

Since every interim sequential equilibrium is an ex ante sequential equilibrium, it follows that every interim sequential equilibrium is a NE as well.

In general, not every NE is a sequential equilibrium. However, under some natural assumptions on the complexity function, the statement holds. A strategy profile $\vec{M}$ in a machine game G is *lean* if, for all players i and local states q of M_i, q is reached with positive probability when playing $\vec{M}$.

Proposition 6.2 *If $\vec{M}$ is a lean NE for machine game G and μ is a belief system compatible with $\vec{M}$, then $(\vec{M}, \mu)$ is an ex ante sequential equilibrium.*

The restriction to local variants (M_i, q, M'_i) of M_i in the definition of ex ante sequential equilibrium is critical here. Proposition 6.2 does not hold for interim sequential equilibrium, as the following example, which illustrates the role of locality, shows.

Example 6.3 Let x be an n-bit string whose Kolmogorov complexity is n (i.e., x is incompressible—there is no shorter description of x). Consider a single-agent game G_x (so the x is built into the game; it is not part of the input) where the agent's type is a string of length $\log n$, chosen uniformly at random, and the utility function is defined as follows, for an agent with type t:

- The agent "wins" if it outputs $(1, y)$, where $y = x_t$ (i.e., it manages to guess the tth bit of x, where t is its type). In this case, it receives a utility of 10, as long as its complexity is at most 2.
- The agent can also "give up": if it outputs t_0 (i.e., the first bit of its type) and its complexity is 1, then it receives a utility of 0.
- Otherwise, its utility is $-\infty$.

Consider the 4-state TM M that just "gives up". Formally, $M = (\tau, \{q_0, b_0, b_1, H\}, q_0, \{H\})$, where τ is such that, in q_0, M reads the first bit t_0 of the type, and transitions to b_i if it is i; and in state b_i, it outputs i and transitions to H, the halt state. Take the complexity of M to be 1 (on all inputs); the complexity of any TM $M' \neq M$ that has at most $0.9n$ states is 2; and all other TMs have complexity 3.

Note that M is the unique NE in G_x. Since x is incompressible, no TM M^* with fewer than $0.9n$ states can correctly guess x_t for all t (for otherwise M^* would provide a description of x shorter than $|x|$). It follows that no TM with complexity greater than 1 does better than M. Thus, M is the unique NE. It is also a lean NE, and thus, by Proposition 6.2, an ex ante sequential equilibrium. However, there exists a non-local variant of M at b_0 that gives higher utility than M. Notice that if the first bit is 0 (i.e., if the TM is in state b_0), then x_t is one of the first $n/2$ bits of x. Thus, at b_0, we can switch to the TM M' that reads the whole type t and the first $n/2$ bits of x, outputs $(1, x_t)$. It is easy to see that M' can be constructed using $0.5n + O(1)$ states. Thus, M is not an interim sequential equilibrium (in fact, none exists in G_x). (M, b_0, M') is not a local variant of M, since M' has higher complexity than M at q_0. ∎

We now show that for a natural class of games, every NE is lean, and thus also an ex ante sequential equilibrium. Intuitively, we consider games where there is a strictly positive cost for having more states. Our argument is similar in spirit to that of [Rubinstein, 1986], where it is shown that in games with automata, there is always a NE with no "wasted" states; all states are reached in equilibrium. Roughly speaking, a machine game G has *positive state cost* if (a) a state q is not reached in TM M, and M^{-q} is the TM that results from removing q from M, then $\mathscr{C}_i(M^{-q}, v) < \mathscr{C}_i(M, v)$; and (b) utilities are monotone decreasing in complexity; that is, $u_i(\vec{t}, \vec{a}, (c'_i, \vec{c}_{-i})) < u_i(\vec{t}, \vec{a}, (c_i, \vec{c}_i))$ if $c'_i > c_i$. (See the full paper for the precise definition.)

Lemma 6.4 *Every NE $\vec{M}$ for machine game G with positive state cost is lean.*

Combining Proposition 6.2 and Lemma 6.4, we immediately get the following result.

Theorem 6.5 *If $\vec{M}$ is a NE for a machine game G with positive state cost, and μ is a belief system compatible with $\vec{M}$, then $(\vec{M}, \mu)$ is an ex ante sequential equilibrium.*

One might be tempted to conclude from Theorem 6.5 that sequential equilibria are not interesting, since every NE is a sequential equilibrium. But this result depends on the assumption of positive state cost in a critical way, as the following simple example shows.

Example 6.6 Consider a single-agent game where the type space is $\{0, 1\}$, and the agent gets payoff 1 if he outputs his type, and otherwise gets 0. Suppose that all TMs have complexity 0 on all inputs (so that the game does not have positive state cost), and that the type distribution assigns probability 1 to the type being 0. Let M be the 4-state TM that reads the input and then outputs 0. Formally, $M = (\tau, \{q_0, b^0, b^1, H\}, q_0, \{H\})$, where τ is such that in q_0, M reads the type t and transitions to b_i if the type is i; and in state b_i, it outputs 0 and transitions to H, the halt state. M is clearly a NE, since $b = 0$ with probability 1. However, M is not an ex ante sequential equilibrium, since conditioned on reaching b_1, outputting 1 and transitioning to H yields higher utility; furthermore, note that this change is a local variant of M since all TMs have complexity 0. ∎

7 Existence

We cannot hope to prove a general existence theorem for sequential equilibrium, since not every game has even a NE, and by Proposition 6.1, every ex ante sequential equilibrium is a NE. Nonetheless, we show that for any Bayesian machine game G where the set $\mathcal{M}$ of possible TMs that can be chosen is finite, if G has a NE, then it has a sequential equilibrium. More precisely, we show that in every game where the set $\mathcal{M}$ of possible TMs that can be chosen is finite, every NE can be converted to an ex ante sequential equilibrium with the same distribution over outcomes. As illustrated in Example 6.6, not every NE is an ex ante sequential equilibrium; thus, in general, we must modify the original equilibrium.

Theorem 7.1 *Let G be a machine game where the set $\mathcal{M}$ of TMs is finite. If G has a NE, then it has an ex ante sequential equilibrium with the same distribution over outcomes.*

We end this section by providing some existence results for games with infinite machine spaces. As shown in Theorem 6.5, in games with positive state cost, every NE is an ex ante sequential equilibrium. Although positive state cost is a reasonable requirement in many settings, it is certainly a nontrivial requirement. A game G has *non-negative state cost* if the two conditions in the definition of "positive state cost" hold when replacing the strict inequalities with non-strict inequalities. That is, roughly speaking, G has non-negative state cost if adding machine states (without changing the functionality of the TM) can never improve the utility. It is hard to imagine natural games with negative state cost. In particular, a complexity function that assigns complexity 0 to all TMs and inputs has non-negative state cost. Say that G is *complexity-local* if, for each player i, i's utility does not depend on the complexity of players $-i$. (Note that all single-player games are trivially complexity local.) Although non-negative state cost combined with complexity-locality is not enough to guarantee that every NE is an ex ante sequential equilibrium (as illustrated by Example 6.6), it is enough to guarantee the existence of an ex ante sequential equilibrium.

Proposition 7.2 *If G is a complexity-local machine game with non-negative state cost that has a NE, then it has a lean NE with the same distribution over outcomes.*

Corollary 7.3 *If G is a complexity-local machine game with non-negative state cost, and G has a NE, then G has an ex ante sequential equilibrium.*

References

[Fagin *et al.*, 1995] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, Cambridge, Mass., 1995. A slightly revised paperback version was published in 2003.

[Halpern and Pass, 2011a] J. Y. Halpern and R. Pass. Algorithmic rationality: Game theory with costly computation. 2011. Available at www.cs.cornell.edu/home/halpern/papers/algrationality.pdf; to appear, *Journal of Economic Theory*. A preliminary version with the title "Game theory with costly computation" appears in *Proc. First Symposium on Innovations in Computer Science*, 2010.

[Halpern and Pass, 2011b] J. Y. Halpern and R. Pass. Sequential equilibrium and perfect equilibrium in games of imperfect recall. Unpublished manuscript; available at www.cs.cornell.edu/home/halpern/papers/imperfect.pdf., 2011.

[Kreps and Wilson, 1982] D. M. Kreps and R. B. Wilson. Sequential equilibria. *Econometrica*, 50:863–894, 1982.

[Piccione and Rubinstein, 1997] M. Piccione and A. Rubinstein. On the interpretation of decision problems with imperfect recall. *Games and Economic Behavior*, 20(1):3–24, 1997.

[Rubinstein, 1986] A. Rubinstein. Finite automata play the repeated prisoner's dilemma. *Journal of Economic Theory*, 39:83–96, 1986.