Generalised Entropies and Asymptotic Complexities of Languages*

Yuri Kalnishkan, **Michael V. Vyugin**, and **Vladimir Vovk**
Computer Learning Research Centre and
Department of Computer Science,
Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, UK
{yura, misha, vovk}@cs.rhul.ac.uk

January 7, 2014

Abstract

The paper explores connections between asymptotic complexity and generalised entropy. Asymptotic complexity of a language (a language is a set of finite or infinite strings) is a way of formalising the complexity of predicting the next element in a sequence: it is the loss per element of a strategy asymptotically optimal for that language. Generalised entropy extends Shannon entropy to arbitrary loss functions; it is the optimal expected loss given a distribution on possible outcomes. It turns out that the set of tuples of asymptotic complexities of a language w.r.t. different loss functions can be described by means of the generalised entropies corresponding to the loss functions.

1 Introduction

We consider the following on-line learning scenario: given a sequence of previous outcomes $x_1, x_2, \ldots, x_{n-1}$, a prediction strategy is required to output a prediction γ_n for the next outcome x_n.

Let the outcomes belong to a finite set Ω; it may be thought of as an alphabet and sequences as words. We allow greater variation in predictions

*A previous version of this paper was published in Proceedings of the 20th Annual Conference on Learning Theory, COLT-2007, Lecture Notes in Computer Science, vol. 4539 Springer 2007.

though. Predictions may be drawn from a compact set. A loss function $\lambda(\omega, \gamma)$ is used to measure the discrepancy between predictions and actual outcomes. The performance of the strategy is measured by the cumulative loss $\sum_{i=1}^{n} \lambda(x_i, \gamma_i)$ and the learner's goal is to make the loss as small as possible. Different aspects of this prediction problem have been extensively studied; see [CBL06] for an overview.

One is tempted to define complexity of a string as the loss of an optimal strategy so that elements of "simple" strings $\boldsymbol{x}$ are easy to predict and elements of "complicated" strings are hard to predict and large loss is incurred. However this intuitive idea is difficult to implement formally because it is hard to define an optimal strategy. If $\boldsymbol{x}$ is fixed, the strategy can be tailored to suffer the minimum possible loss on $\boldsymbol{x}$ (0 for natural loss functions such as square, absolute, or logarithmic). If there is complete flexibility in the choice of $\boldsymbol{x}$, i.e., "anything can happen", then every strategy can be tricked into suffering large loss and being greatly outperformed by some other strategy on some sequences $\boldsymbol{x}$.

One approach to this problem is predictive complexity introduced in [VW98] and studied in [Kal02, KVV04, KVV05]. This approach replaces strategies by the class of semi-computable superloss processes. Under certain restrictions on the prediction space and loss function this class has a natural optimal element. Predictive complexity of a finite string is defined up to a constant and is similar in many respects to Kolmogorov complexity; predictive complexity w.r.t. the logarithmic loss function equals the negative logarithm of Levin's a priori semi-measure.

This paper takes a different approach and introduces asymptotic complexity, which is in some respects easier and more intuitive. It is defined for languages (infinite sets of finite strings and sets of infinite sequences) and it equals the asymptotically optimal loss per element. This idea leads to several versions of complexity that behave slightly differently. An important advantage of this approach is that asymptotic complexity exists for all loss functions λ thus eliminating the question of existence, still partly unsolved for predictive complexity. One can consider effective and polynomial-time versions of asymptotic complexity by restricting oneself to computable or polynomial-time computable strategies. The existence of corresponding asymptotic complexities follows trivially.

There are other approaches to universality in prediction, e.g., those assuming the existence of a probability law generating the outcomes (see, e.g., [Hut04, Rya11]). We do not make this assumption. While a distribution always lurks in the background, the main definitions are formulated in a worst-case fashion.

In this paper we study the following question. Consider K different prediction environments, i.e., several different loss functions. How does the asymptotic complexity of a language vary from environment to environment? If we know the complexity w.r.t. one environment, what can we say about the complexities w.r.t. others?

We answer this question by describing the set

$$(\mathrm{AC}_1(L), \mathrm{AC}_2(L), \ldots, \mathrm{AC}_K(L)) \subseteq \mathbb{R}^K \ ,$$

where AC_k is an asymptotic complexity w.r.t. the kth environment and L ranges over all non-trivial languages. The set turns out to have a simple geometric description in terms of generalised entropy also known as Bayes loss (see [GD04] for a discussion of this concept). The set depends on the type of asymptotic complexity and may be different for different complexities [1].

Connections between Shannon entropy and complexity have long been studied; see [Rya86] and Theorem 2.8.1 in [LV08]. This paper was inspired by [FL05] and directly generalises its main result. In Section 6 we show how the concepts of predictability and dimension from [FL05] relate to asymptotic complexities and derive the result of [FL05] from our main theorem. Note that we strengthen the result by showing that the pairs of predictabilities and dimensions actually fill in the respective set.

While [FL05] restricts itself to polynomial-time computable strategies, we formulate three parallel versions of the main result, "non-effective", effective, and polynomial-time computable. The computational aspects of on-line prediction in the framework we are interested in do not appear to be well developed, so we included a number of auxiliary statements about computability and polynomial-time computability in appendixes.

The structure of the paper is as follows. In Section 2 we introduce the main concepts including entropy hulls, lattices, and closures in $\mathbb{R}^K$. Section 3 defines asymptotic complexities. In starts with non-effective versions and then proceeds to effective and polynomial-time versions. The key definitions related to the computational model are given in Section 3.2.1.

In Section 4 we formulate the main theorem and discuss its statement. In Section 4.2 we present several two-dimensional illustrations.

The proof of the main result is given in Section 5. We start by discussing some geometric properties of lattices and entropies in Section 5.1. Then in Section 5.2 we formulate and prove the recalibration lemma, which is the main tool of the proof. In Section 5.3 we show that tuples of complexities

[1] Note that the statement of the main theorem in the conference version [KVV07] of this paper was inaccurate in this respect. Details are given in Section 4.

must belong to certain sets in $\mathbb{R}^K$ and in Section 5.4 we show that the tuples fill those sets.

In Section 6 we apply the main result to the concepts of predictability and dimension and their computable and polynomial-time counterparts.

2 Preliminaries

We use $\mathbb{Z}$ to denote the set of integers and $\mathbb{N}$ to denote the set of non-negative integers $\{0, 1, 2, \ldots\}$.

2.1 Games, Strategies, and Losses

The notation in the paper mostly follows [KV08].

A *game* $\mathfrak{G}$ is a triple $\langle \Omega, \Gamma, \lambda \rangle$, where Ω is an *outcome space*, Γ is a *prediction space*, and $\lambda : \Omega \times \Gamma \to [0, +\infty]$ is a *loss function*.

We assume that $\Omega = \{\omega^{(0)}, \omega^{(1)}, \ldots, \omega^{(M-1)}\}$ is a finite set of cardinality $M < +\infty$. If $M = 2$, then Ω may be identified with $\mathbb{B} = \{0, 1\}$; we will call this case *binary*. We denote the set of all finite sequences of elements of Ω by Ω^* and the set of all (one-sided) infinite sequences by Ω^∞. Sets of finite sequences and sets of infinite sequences, i.e., subsets of Ω^* and Ω^∞, will be sometimes referred to as *languages*.

Bold letters $\boldsymbol{x}$, $\boldsymbol{y}$ etc. are used to refer to both finite and infinite sequences. By $|\boldsymbol{x}|$ we denote the length of a finite sequence $\boldsymbol{x}$, i.e., the number of elements in it. The set of sequences of length n is denoted by Ω^n, $n = 0, 1, 2, \ldots$. We will also be using the notation $\sharp_i \boldsymbol{x}$ for the number of $\omega^{(i)}$s among elements of $\boldsymbol{x}$. Clearly, $\sum_{i=0}^{M-1} \sharp_i \boldsymbol{x} = |\boldsymbol{x}|$ for any finite sequence $\boldsymbol{x}$. We use $\boldsymbol{x}|_n$ to denote the prefix of length n of a (finite of infinite) sequence $\boldsymbol{x}$. If $\boldsymbol{x}$ is finite and $\boldsymbol{y}$ is finite or infinite, we denote the concatenation of $\boldsymbol{x}$ and $\boldsymbol{y}$ by $\boldsymbol{xy}$. If X is a set of finite strings and Y is a set of finite or infinite strings, $X \times Y$ denotes the set of all concatenations $\boldsymbol{xy}$, where $\boldsymbol{x} \in X$ and $\boldsymbol{y} \in Y$. We use ω^n to denote the string consisting of n identical elements equal to ω and ω^∞ to denote the infinite string consisting of ωs.

We also assume that Γ is a compact topological space and λ is continuous w.r.t. the topology of the extended half-line $[0, +\infty]$. We treat Ω as a discrete space and thus the continuity of λ in two arguments is the same as continuity in the second argument.

In order to take some important games into account we must allow λ to attain the value $+\infty$. However we assume that the set $\Gamma_{\mathrm{fin}} = \{\gamma \in \Gamma \mid \max_{\omega \in \Omega} \lambda(\omega, \gamma) < +\infty\}$ is dense in Γ. In other words, every prediction

γ_0 leading to infinite loss can be approximated by predictions giving finite losses[2].

The following are examples of binary games with $\Omega = \mathbb{B}$ and $\Gamma = [0,1]$: the *square-loss* game with the loss function $\lambda(\omega,\gamma) = (\omega-\gamma)^2$, the *absolute-loss* game with the loss function $\lambda(\omega,\gamma) = |\omega-\gamma|$, and the *logarithmic* game with

$$\lambda(\omega,\gamma) = \begin{cases} -\log_2(1-\gamma) & \text{if} \quad \omega = 0 \\ -\log_2 \gamma & \text{if} \quad \omega = 1 \end{cases} .$$

A *prediction strategy* $\mathfrak{A} : \Omega^* \to \Gamma$ maps a finite sequence of outcomes to a prediction. We say that on a finite sequence $\boldsymbol{x} = x_1 x_2 \dots x_n \in \Omega^n$ the strategy $\mathfrak{A}$ suffers loss $\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}}(\boldsymbol{x}) = \sum_{i=1}^n \lambda(x_i, \mathfrak{A}(x_1 x_2 \dots x_{i-1}))$. By definition, we let $\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}}(\Lambda) = 0$, where Λ is the sequence of length 0. The upper index $\mathfrak{G}$ will be omitted if it is clear from the context.

We need to define one important class of games. A game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ is *weakly mixable* if for every two prediction strategies $\mathfrak{A}_1$ and $\mathfrak{A}_2$ there is a strategy $\mathfrak{A}$ and a function $\alpha : \mathbb{N} \to \mathbb{R}$ such that $\alpha(n) = o(n)$ as $n \to +\infty$ and

$$\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}}(\boldsymbol{x}) \le \min\left(\mathrm{Loss}_{\mathfrak{A}_1}(\boldsymbol{x}), \mathrm{Loss}_{\mathfrak{A}_2}(\boldsymbol{x})\right) + \alpha(|\boldsymbol{x}|) \tag{1}$$

for all finite sequences $\boldsymbol{x} \in \Omega^*$.

It is easy to describe weakly mixable games in geometric terms. An M-tuple $\left(s^{(0)}, s^{(1)}, \dots, s^{(M-1)}\right) \in [0,+\infty]^M$ is a *superprediction* w.r.t. $\mathfrak{G}$ if there is a prediction $\gamma \in \Gamma$ such that $\lambda(\omega^{(i)}, \gamma) \le s^{(i)}$ for all $i = 0, 1, \dots, M-1$. We say that a game $\mathfrak{G}$ with the set of superpredictions S is *convex* if the finite part of the set of superpredictions, $S \cap \mathbb{R}^M$, is convex.

Proposition 1 ([KV08], Theorem 7)**.** *A game is weakly mixable if and only if it is convex.*

It is easy to check that the square-loss, absolute-loss, and logarithmic games are convex and therefore weakly mixable.

2.2 Generalised Entropies

The term "generalised entropy" was introduces in [GD04], where entropy was extensively studied in a context similar to ours. This concept has long

[2] In [KV08] and other earlier papers it was required that for every $\gamma_0 \in \Gamma$ such that $\lambda(\omega^*, \gamma_0) = +\infty$ for some $\omega^* \in \Omega$ there should be a sequence $\gamma_1, \gamma_2, \ldots \in \Gamma_{\mathrm{fin}}$ converging to γ_0. If Γ is a metric space, then the existence of converging sequences is clearly equivalent to the denseness of Γ_{fin} in Γ. However, if Γ is a topological space, then the denseness of Γ_{fin} does not always imply the existence of converging sequences; see, e.g., Section 12.11 in [GO03]. Fortunately, these sequences are not really needed for our purposes. We thus require that Γ_{fin} should be dense, which is more general.

been known in statistical decision theory under the name of Bayes loss or Bayes risk (see [DeG70], Section 8.2).

Fix a game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$. Let $\mathbb{P}(\Omega)$ be the set of probability distributions on Ω. Since Ω is finite, we can identify $\mathbb{P}(\Omega)$ with the standard $(M-1)$-simplex $\mathbb{P}_M = \left\{ \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)}\right) \in [0,1]^M \mid \sum_{i=0}^{M-1} p^{(i)} = 1 \right\}$.

Generalised entropy $H : \mathbb{P}(\Omega) \to \mathbb{R}$ is the infimum of expected loss over $\gamma \in \Gamma$, i.e., for $p^* = \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)}\right) \in \mathbb{P}(\Omega)$ we have

$$H(p^*) = \inf_{\gamma\in\Gamma} \mathbf{E}_{p^*} \lambda(\omega, \gamma) = \inf_{\gamma\in\Gamma} \sum_{i=0}^{M-1} p^{(i)} \lambda(\omega^{(i)}, \gamma) \ .$$

Since $p^{(i)}$ can be 0 and $\lambda(\omega^{(i)}, \gamma)$ can be $+\infty$, we need to resolve a possible ambiguity. Let us assume that in this definition $0 \times (+\infty) = 0$.

The infimum in the definition is actually achieved on a $\gamma_0 \in \Gamma$ and we can replace inf by min in the definition of H because λ is continuous and Γ is compact. One cannot use the standard theorem from analysis to prove this because λ is not a "standard" continuous function and can take infinite values. However it is easy to check directly. Fix $p^* = \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)}\right) \in \mathbb{P}_M$ and let $\gamma_1, \gamma_2, \ldots \in \Gamma$ be such that the expectations $\sum_{i=0}^{M-1} p^{(i)} \lambda(\omega^{(i)}, \gamma_n)$ converge to the infimum of the expectation for this p^* and $\gamma_n \to \gamma_0 \in \Gamma$ as $n \to \infty$. If $\lambda(\omega, \gamma_0)$ is finite for all $\omega \in \Omega$, then the infimum is achieved at γ_0 by the continuity of λ. Now suppose that $\lambda(\omega, \gamma_0)$ are infinite for some $\omega^{(i)}$. Clearly, the corresponding components $p^{(i)}$ must be zero, or otherwise the expectations $\sum_{i=0}^{M-1} p^{(i)} \lambda(\omega^{(i)}, \gamma_n)$ converge to infinity. Therefore we can drop the corresponding terms from the sum and obtain the desired result by the continuity of λ.

The assumption (made in Section 2.1) that any prediction leading to an infinite loss can be approximated by predictions with finite losses implies that the infimum in the definition of H can be taken over the values of $\gamma \in \Gamma$ such that $\lambda(\omega, \gamma) < +\infty$ for all $\omega \in \Omega$.

In the binary case $\Omega = \mathbb{B}$ the definition can be simplified. Let p be the probability of 1. Clearly, p fully specifies a distribution from $\mathbb{P}(\mathbb{B})$ and thus $\mathbb{P}(\mathbb{B})$ can be identified with the line segment $[0, 1]$. We get $H(p) = \min_{\gamma\in\Gamma}[(1-p)\lambda(0, \gamma) + p\lambda(1, \gamma)]$. For the logarithmic game this gives us Shannon entropy hence the term "generalised entropy" for arbitrary games.

Take $K \geq 1$ games $\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_K$ with the same finite set of outcomes Ω. Let H_k be the $\mathfrak{G}_k$-entropy for $k = 1, 2, \ldots, K$. The *$\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$-entropy set* is the set $\{(H_1(p), H_2(p), \ldots, H_K(p)) \mid p \in \mathbb{P}(\Omega)\} \subseteq \mathbb{R}^K$. The convex hull of the $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$-entropy set is called the *$\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$-entropy hull*.

Remark 2. If outcomes are i.i.d. (independent identically distributed) random values according to a distribution p^*, then $H(p^*)$ is the optimal average loss per element of a prediction strategy. That is why entropy plays such an important role in the study of optimal prediction strategies in general and asymptotic complexities in particular.

2.3 Lattices and Closures

A set $\mathcal{M}$ with a partial order is a *lattice* if every two elements from $\mathcal{M}$ have a least upper bound and a greatest lower bound in $\mathcal{M}$.

For $x = (x_1, x_2, \ldots, x_K)$ and $y = (y_1, y_2, \ldots, y_K)$ let $x \le y$ if $x_i \le y_i$ for all $i = 1, 2, \ldots, K$. This relation is a partial order. Let $\max(x_1, x_2, \ldots, x_n)$, where $x_i = \left(x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(K)}\right) \in \mathbb{R}^K$, $i = 1, 2, \ldots, n$, be the componentwise maximum

$$\left(\max\left(x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}\right), \ldots, \max\left(x_1^{(K)}, x_2^{(K)}, \ldots, x_n^{(K)}\right)\right)$$

and $\min(x_1, x_2, \ldots, x_n)$ be the componentwise minimum. We will also use the notation $\max \mathcal{M}$ for the componentwise maximum of a possibly infinite set $\mathcal{M} \subseteq \mathbb{R}^K$ provided each maximum

$$\begin{array}{l}
\max\left\{x^{(1)} \mid \left(x^{(1)}, x^{(2)}, \ldots, x^{(K)}\right) \in \mathcal{M}\right\} \\
\max\left\{x^{(2)} \mid \left(x^{(1)}, x^{(2)}, \ldots, x^{(K)}\right) \in \mathcal{M}\right\} \\
\quad \ldots \\
\max\left\{x^{(K)} \mid \left(x^{(1)}, x^{(2)}, \ldots, x^{(K)}\right) \in \mathcal{M}\right\}
\end{array}$$

exists; the same applies to the componentwise minimum $\min \mathcal{M}$. Clearly, $\max(x, y)$ is the least upper bound and $\min(x, y)$ is the greatest lower bound of x and y in $\mathbb{R}^K$ w.r.t. the partial order $\le$, i.e., $\mathbb{R}^K$ with $\le$ is a lattice.

A set $\mathcal{M} \subseteq \mathbb{R}^K$ is a *sublattice*[3] of $\mathbb{R}^K$ if for every $x, y \in \mathcal{M}$ it contains their greatest lower bound $\min(x, y)$ and least upper bound $\max(x, y)$. Clearly, a sublattice contains the maximum and minimum of any finite subset. (In what follows we will speak about sublattices without mentioning $\mathbb{R}^K$ or $\le$ as this set and this relation will always be assumed.)

Similarly, a set $\mathcal{M}$ with a partial order $\le$ is an *upper semilattice* if every two elements from $\mathcal{M}$ have a least upper bound in $\mathcal{M}$ and a *lower semilattice*

[3]Note that a set $\mathcal{M} \subseteq \mathbb{R}^K$ can be a lattice w.r.t. $\le$ without being a sublattice of $\mathbb{R}^K$; cf. [Bir48, II.4]. For example, $\mathcal{M} = \{(-1, -1), (1, 0), (0, 1), (2, 2)\} \subseteq \mathbb{R}^2$ is of this kind.

if every two points from $\mathcal{M}$ have a greatest lower bound in $\mathcal{M}$. A set $\mathcal{M} \subseteq \mathbb{R}^K$ is an *upper subsemilattice* (of $\mathbb{R}^K$ w.r.t. $\leq$) if for every $x, y \in \mathcal{M}$ it contains $\max(x, y)$. A set $\mathcal{M} \subseteq \mathbb{R}^K$ is a *lower subsemilattice* (of $\mathbb{R}^K$ w.r.t. $\leq$) if for every $x, y \in \mathcal{M}$ it contains their greatest lower bound $\min(x, y)$.

The $\leq$-*closure* of a set $\mathcal{M} \subseteq \mathbb{R}^K$ is the smallest sublattice containing $\mathcal{M}$. Respectively, the *upper* $\leq$-*closure* of a set $\mathcal{M} \subseteq \mathbb{R}^K$ is the smallest upper subsemilattice containing $\mathcal{M}$ and the *lower* $\leq$-*closure* of a set $\mathcal{M} \subseteq \mathbb{R}^K$ is the smallest lower subsemilattice containing $\mathcal{M}$. The $\leq$-closure of $\mathcal{M}$ exists and it is the intersection of all sublattices containing $\mathcal{M}$; the same applies to the upper and lower $\leq$-closures. The $\leq$-closure contains the upper and lower $\leq$-closures because each sublattice is a subsemilattice.

Note that the definitions of $\leq$ and all subsequent concepts are coordinate-dependent.

3 Asymptotic Complexities

In this section we define measures of complexity for *languages*, i.e., sets of sequences. The finite and infinite sequences should be considered separately. Let a game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ have a finite outcome space Ω.

3.1 Non-effective Case

We start by giving basic definitions with no regard to computability.

3.1.1 Finite Sequences

Consider $L \subseteq \Omega^*$. We will call the values

$$\overline{\mathrm{AC}}(L) = \inf_{\mathfrak{A}} \limsup_{n \to +\infty} \max_{\boldsymbol{x} \in L \cap \Omega^n} \frac{\mathrm{Loss}_{\mathfrak{A}}(\boldsymbol{x})}{n} \quad , \tag{2}$$

$$\underline{\mathrm{AC}}(L) = \inf_{\mathfrak{A}} \liminf_{n \to +\infty} \max_{\boldsymbol{x} \in L \cap \Omega^n} \frac{\mathrm{Loss}_{\mathfrak{A}}(\boldsymbol{x})}{n} \tag{3}$$

the *upper* and *lower asymptotic complexity* of L w.r.t. the game $\mathfrak{G}$. As with generalised entropies, we will use subscripts for AC to specify a particular game if it is not clear from the context.

In order to complete the definition, we must decide what to do if L contains no sequences of certain lengths at all. In this paper we are concerned only with infinite sets of finite sequences and asymptotic complexity of a finite or an empty language $L \subseteq \Omega^*$ is undefined. Thus by assumption there are strings of infinitely many lengths in L.

Still there may be no strings of a certain length in L. Let us assume that the limits in (2) and (3) are taken over all the values $n_1 < n_2 < \ldots$ such that $L \cap \Omega^{n_i} \neq \varnothing$.

3.1.2 Infinite Sequences

There are two natural ways to define complexities of nonempty languages $L \subseteq \Omega^\infty$.

First we can extend the notions we have just defined. Indeed, for a nonempty set of infinite sequences consider the set of all finite prefixes of all its sequences[4]. The language thus obtained is infinite and has upper and lower complexities. For the resulting complexities we will retain the notation $\overline{\mathrm{AC}}(L)$ and $\underline{\mathrm{AC}}(L)$. We will refer to these complexities as *uniform*.

The second way is the following. Let

$$\overline{\overline{\mathrm{AC}}}(L) = \inf_{\mathfrak{A}} \sup_{\boldsymbol{x} \in L} \limsup_{n \to +\infty} \frac{\mathrm{Loss}_{\mathfrak{A}}(\boldsymbol{x}|_n)}{n} \ , \tag{4}$$

$$\underline{\underline{\mathrm{AC}}}(L) = \inf_{\mathfrak{A}} \sup_{\boldsymbol{x} \in L} \liminf_{n \to +\infty} \frac{\mathrm{Loss}_{\mathfrak{A}}(\boldsymbol{x}|_n)}{n} \ . \tag{5}$$

We will refer to these complexities as *non-uniform*.

The concept of asymptotic complexity generalises certain complexity measures studied in the literature. The concepts of predictability and dimension studied in [FL05] can be easily reduced to asymptotic complexity: dimension is lower non-uniform complexity w.r.t. a multidimensional generalisation of the logarithmic game and predictability equals $1 - \underline{\underline{\mathrm{AC}}}$, where $\underline{\underline{\mathrm{AC}}}$ is lower non-uniform complexity w.r.t. a multidimensional generalisation of the absolute-loss game; see Section 6 for derivations.

3.2 Effective Versions of Complexities

In this section we define effective versions of games, strategies, and complexities.

3.2.1 Computability Model

While computability over the real domain is a relatively well-known area, polynomial-time computability with real numbers is less popular. We will

[4]As pointed out by an anonymous reviewer, this mapping of subsets of Ω^∞ to subsets of Ω^* is not injective. Indeed, $L_1 = \bigcup_{n=0}^{\infty} (0^n 1^\infty)$ and $L_2 = L_1 \cup \{0^\infty\}$ have the same sets of prefixes. Therefore subsets of Ω^∞ cannot be *identified* with their sets of prefixes.

describe a computational model along the lines of [Ko91] (see also [Wei00, Sections 7 and 9.4]) in detail to avoid ambiguity.

A *dyadic* rational number is a number of the form $n/2^m$, where n and m are integers. We call a triple $r = \langle b, \boldsymbol{x}, \boldsymbol{y} \rangle$, where $b \in \mathbb{B}$ is a bit and $\boldsymbol{x} = (x_1 x_2 \dots x_u), \boldsymbol{y} = (y_1 y_2 \dots y_v) \in \mathbb{B}^*$ are binary strings, a *representation of a dyadic number* d if $x_1 = 1$ and

$$d = s \left(\sum_{i=0}^{u-1} x_{u-i} 2^i + \sum_{i=1}^{v} y_i 2^{-i} \right) \quad , \tag{6}$$

where $s = 1$ if $b = 1$ and $s = -1$ if $b = 0$. Intuitively, b represents the sign of d and $\boldsymbol{x}.\boldsymbol{y}$ is a finite binary expansion of $|d|$. Let d map correctly formed triples into dyadic numbers according to (6). We will call v the *precision* of the triple r and write $v = \mathrm{prec}(r)$.

For every $x \in \mathbb{R}$ let CF_x be the set of sequences of triples, i.e., functions ϕ from non-negative integers to representations of dyadic numbers, such that $\mathrm{prec}(\phi(m)) = m$ and $|\,\mathrm{d}(\phi(m)) - x| \leq 2^{-m}$ for all $m = 1, 2, \dots$. Any element of CF_x can be thought of as a representation of x. A number $x \in \mathbb{R}$ is *computable* if CF_x contains a computable function ϕ. If there is a Turing machine taking a unary representation of m as input and finishing work in time polynomial in m and computing $\phi(m)$ for all $m \in \mathbb{N}$, then x is *polynomial-time computable* (we use unary notation in line with Definitions 2.6 and 2.7 from [Ko91] so that the running time of the machine is polynomial in the length of the input). A point $x = (x_1, x_2, \dots, x_K) \in \mathbb{R}^K$ is (polynomial-time) computable if all its coordinates $x_1, x_2, \dots, x_K$ are (polynomial-time) computable.

Let Ω be a finite set. A function $f : \Omega^* \to \mathbb{R}$ is *computable* if there is a Turing machine that given a finite string $\boldsymbol{x} = x_1 x_2 \dots x_n \in \Omega^*$ and non-negative integer precision m outputs a representation r of a dyadic number such that $\mathrm{prec}(r) = m$ and $|f(\boldsymbol{x}) - \mathrm{d}(r)| \leq 2^{-m}$. In other words, for every $\boldsymbol{x} \in \Omega^*$ the machine calculates a function from $\mathrm{CF}_{f(\boldsymbol{x})}$. If the machine takes the unary representation of m as input and there is a polynomial $p(\cdot, \cdot)$ such that the machine always finishes work in time $p(n, m)$, we say that f is polynomial-time computable. A function $f = (f_1, f_2, \dots, f_K) : \Omega^* \to \mathbb{R}^K$ is (polynomial-time) computable if all its components $f_1, f_2, \dots, f_K$ are (polynomial-time) computable.

A function $f : M \to \mathbb{R}$, where $M \subseteq \mathbb{R}$, is computable if there is an oracle Turing machine that given a non-negative integer precision m and an oracle evaluating some $\phi \in \mathrm{CF}_x$ outputs a representation r of a dyadic number such that $\mathrm{prec}(r) = m$ and $|f(x) - \mathrm{d}(r)| \leq 2^{-m}$. Suppose that the machine takes the unary representation of m as input and works with the oracle in the following way. When it needs an approximation of x with precision 2^{-k} it prints the unary representation of k (say, on a special tape reserved for this purpose), requests an approximation, and gets it in one unit of time (say, on another special tape). If there is a polynomial $p(\cdot)$ such that the machine finishes work in time $p(m)$ for all $x \in M$ and $m \in \mathbb{N}$, we say that f is polynomial-time computable. It is easy to see that the classes of computable and polynomial-time computable functions are closed under composition (for the latter recall the unary input rule). Computable and polynomial-time computable functions on $M \subseteq \mathbb{R}^K$ and $M \times \Omega^*$ to $\mathbb{R}$ and $\mathbb{R}^K$ are defined in a similar fashion.

Some important properties of computable and polynomial-time computable functions are outlined in Appendix A.

3.2.2 Computable Games

We call a game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ *(polynomial-time) computable* if $\Gamma \subseteq \mathbb{R}^m$ for some positive integer m, (polynomial-time) computable points are dense in Γ and the function $e^{-\lambda(\cdot,\cdot)} : \Omega \times \Gamma \to [0,1]$ is (polynomial-time) computable. Note that we do not postulate computability of λ itself because if would have implied boundedness of λ.

A *(polynomial-time) computable strategy* w.r.t. $\mathfrak{G}$ is a (polynomial-time) computable function $\Omega^* \to \Gamma$.

We need an effective version of mixability. We will give a definition that is very weak but sufficient for the purposes of this paper. A game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ is *(polynomial-time) computably very weakly mixable* if for every two (polynomial-time) computable prediction strategies $\mathfrak{A}_1$ and $\mathfrak{A}_2$ and $\varepsilon > 0$ there is a (polynomial-time) computable strategy $\mathfrak{A}$ and a function $\alpha_\varepsilon : \mathbb{N} \to \mathbb{R}$ such that $\alpha_\varepsilon(n) = o(n)$ as $n \to +\infty$ and

$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}) \le \min\left(\mathrm{Loss}_{\mathfrak{A}_1}(\boldsymbol{x}), \mathrm{Loss}_{\mathfrak{A}_2}(\boldsymbol{x})\right) + \varepsilon|\boldsymbol{x}| + \alpha_\varepsilon(|\boldsymbol{x}|) \tag{7}$$

for all finite sequences $\boldsymbol{x} \in \Omega^*$.

In order to get an effective version of Proposition 1, one needs to restate results of [KV08] in an effective fashion. The procedures used in [KV08] are essentially effective (and efficient) but require certain properties of Γ and λ; otherwise the prediction space and the loss function can be distorted in such a way as to make the procedures from [KV08] unusable. Formalising these properties in a simple form appears to be a difficult task. Instead we will formulate a proposition with simple and concise conditions that are sufficient and rather general but by no means necessary.

Proposition 3. *If a (polynomial-time) computable game $\mathfrak{G}$ has a convex prediction space Γ and a convex in the second argument loss function λ, then $\mathfrak{G}$ is (polynomial-time) computably very weakly mixable.*

This statement is proved in Appendix B.

Remark 4. As demonstrated by Lemma 56 and Remark 58 in the appendix, for many games the conclusion of proposition can be strengthened so that $\varepsilon|\boldsymbol{x}|$ can be dropped from (7) but that is unnecessary for the purposes of this paper and will only make the statements of the theorems more complicated. The complete investigation of this question is outside of the scope of this paper.

Remark 5. In [KV08] the concept of weak mixability was introduced differently by means of a merging strategy (see Section 2.2 in [KV08]) that outputs the predictions of $\mathfrak{A}$ given the predictions of $\mathfrak{A}_1$ and $\mathfrak{A}_2$. While the two definitions are clearly equivalent for the uncomputable case, the effective versions present a problem. We leave out the complete investigation of the equivalence outside of this paper and use the simple definition of mixability as more consistent with the goal of the paper. However we would like to point out that the proofs in Appendix B essentially construct effective merging strategies.

3.2.3 Effective Complexities

The *effective asymptotic complexities* $\overline{\mathrm{ACE}}$, $\underline{\mathrm{ACE}}$, $\overline{\overline{\mathrm{ACE}}}$, and $\underline{\underline{\mathrm{ACE}}}$ w.r.t. a game $\mathfrak{G}$ are defined as in Section 3.1 except that the infimums in (2), (3), (4), and (5) are restricted to computable strategies $\mathfrak{A}$.

The *polynomial-time asymptotic complexities* $\overline{\mathrm{ACP}}$, $\underline{\mathrm{ACP}}$, $\overline{\overline{\mathrm{ACP}}}$, and $\underline{\underline{\mathrm{ACP}}}$ w.r.t. a game $\mathfrak{G}$ are defined as in Section 3.1 except that the infimums in (2), (3), (4), and (5) are restricted to polynomial-time computable strategies $\mathfrak{A}$.

3.3 Simple Relations between Complexities

3.3.1 Inequalities

Since for every sequence of real numbers α_n the lower limit does not exceed the upper limit, lower complexities never exceed the corresponding upper complexities. We get $\overline{\mathrm{AC}}(L) \geq \underline{\mathrm{AC}}(L)$ for all infinite languages $L \subseteq \Omega^*$ and $\overline{\overline{\mathrm{AC}}}(L) \geq \underline{\underline{\mathrm{AC}}}(L)$ for all infinite languages $L \subseteq \Omega^*$ and all nonempty languages $L \subseteq \Omega^\infty$.

For nonempty languages $L \subseteq \Omega^\infty$ we can compare upper uniform and nonuniform complexities. Since for every infinite string $\boldsymbol{y} \in L \subseteq \Omega^\infty$ and every positive integer n the inequality

$$\max_{\boldsymbol{x} \in L} \frac{\mathrm{Loss}(\boldsymbol{x}|_n)}{n} \geq \frac{\mathrm{Loss}(\boldsymbol{y}|_n)}{n}$$

holds, uniform complexities never exceed the corresponding non-uniform complexities; we get $\overline{\mathrm{AC}}(L) \geq \overline{\overline{\mathrm{AC}}}(L)$ and $\underline{\mathrm{AC}}(L) \geq \underline{\underline{\mathrm{AC}}}(L)$. Since for nonempty languages $L \subseteq \Omega^\infty$ all four complexities exist, we can draw the "quadrangle" of inequalities:

$$\begin{array}{ccccc} \overline{\mathrm{AC}}(L) & & \geq & \underline{\mathrm{AC}}(L) & \\ & \rotatebox{-45}{$\geq$} & & & \rotatebox{-45}{$\geq$} \\ & & \overline{\overline{\mathrm{AC}}}(L) & \geq & \underline{\underline{\mathrm{AC}}}(L) \end{array} \tag{8}$$

The same inequalities, including the quadrangle, hold for effective and polynomial-time complexities.

The set of all strategies includes computable strategies and the set of computable strategies includes all polynomial-time computable strategies. Therefore we get

$$\mathrm{AC}(L) \leq \mathrm{ACE}(L) \leq \mathrm{ACP}(L)$$

for every language L and complexity AC such that $\mathrm{AC}(L)$, $\mathrm{ACE}(L)$, and $\mathrm{ACP}(L)$ are defined.

3.3.2 Differences

Let us show that the complexities we have introduced are different and generally speaking the quadrangle does not collapse.

First let us show that upper and lower complexities differ. For example, consider the absolute-loss game. Recall that 0^n is the sequence of n zeros and let $\Xi_n = \{0^n\} \times \mathbb{B}^n$. Consider the language $L = \prod_{i=0}^\infty \Xi_{2^{2^i}} \subseteq \mathbb{B}^\infty$. Informally, L consists of sequences that have alternating constant and random segments of rapidly increasing lengths. At the end of a constant segment the optimal loss per element is low and at the end of a random segment it is high. It is easy to see that $\overline{\mathrm{AC}}(L) = \overline{\overline{\mathrm{AC}}}(L) = 1/2$, while $\underline{\mathrm{AC}}(L) = \underline{\underline{\mathrm{AC}}}(L) = 0$. This follows from $\sum_{j=1}^{i-1} 2^{2^j} \leq \sum_{k=1}^{2^{i-1}} 2^k = 2^{2^{i-1}+1} - 1 = o(2^{2^i})$ as $i \to +\infty$.

Secondly let us show that uniform complexities differ from non-uniform. Once again, consider the absolute-loss game. Let $L \subseteq \mathbb{B}^\infty$ be the set of all sequences that have only zeros from some position on. In other terms,

$L = \cup_{n=0}^{\infty}(\mathbb{B}^n \times \{0^\infty\})$, where 0^∞ is the infinite sequence of zeros. Informally, L consists of sequence that start from a random segment and then stabilise to 0. For every sequence the optimal loss per element goes to zero if we take sufficiently long chunks; however for every n there are sequences that have not stabilised yet so uniform optimal loss must be substantial. We have $\overline{\overline{\mathrm{AC}}}(L) = \underline{\underline{\mathrm{AC}}}(L) = 0$ while $\overline{\mathrm{AC}}(L) = \underline{\mathrm{AC}}(L) = 1/2$.

In Appendix C.1 we discuss these arguments in more details and show how they generalise.

Let us show that non-effective complexities AC differ from their effective counterparts ACE. Again consider the absolute-loss game. There are countably many computable strategies. It is easy to see that weak mixability can be extended to countable families and construct a strategy $\mathfrak{S}$ capturing the power of all computable strategies. The differentiating language $L \subseteq \Omega^\infty$ will consist of a single string $\boldsymbol{x} = \omega_1\omega_2\ldots$. We construct $\boldsymbol{x}$ by induction. Let γ be the prediction output by $\mathfrak{S}$ on $\omega_1\omega_2\ldots\omega_n$. If $\lambda(0,\gamma) \geq 1/2$ take $\omega_{n+1} = 0$ and otherwise take $\omega_{n+1} = 1$. For every computable strategy $\mathfrak{A}$ we get

$$\frac{n}{2} \leq \mathrm{Loss}_{\mathfrak{S}}(\boldsymbol{x}|_n) \leq \mathrm{Loss}_{\mathfrak{A}}(\boldsymbol{x}|_n) + o(n)$$

and thus $\mathrm{ACE}(L) \geq 1/2$ for all complexities ACE. On the other hand, our procedure is deterministic and there is a strategy that predicts elements of $\boldsymbol{x}$ exactly; thus $\mathrm{AC}(L) = 0$.

The same idea can be used to differentiate polynomial-time complexities from effective. This requires some technical steps though: we need an effective version of mixability for countable families and an effective enumeration of all Turing machines calculating polynomial-time computable strategies. Still there is a computable strategy $\mathfrak{S}$ capturing the power of all polynomial-time computable strategies.

There is a computable infinite sequence $\boldsymbol{x}$ such that $\mathrm{Loss}_{\mathfrak{S}}(\boldsymbol{x}|_n) \geq 0.4n$ (for every γ there is ω such that $\lambda(\omega,\gamma) \geq 1/2$ and by calculating $e^{-\lambda}$ with a sufficient precision we will find ω such that $\lambda(\omega,\gamma) \geq 0.4$). On the other hand, there is a computable strategy that precalculates the elements of $\boldsymbol{x}$ and suffers zero loss on prefixes $\boldsymbol{x}|_n$.

More details are given in Appendix C.2.

4 The Main Result and Discussion

4.1 The Main Theorem

We can now formulate the main result of this paper[5]

Theorem 6. *Suppose that games $\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_K$ ($K \geq 1$) have the same finite outcome space Ω and are weakly mixable. Then the $\leq$-closure of the $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_k$-entropy hull coincides with the following sets (here AC_k is asymptotic complexity w.r.t. $\mathfrak{G}_k$, $k = 1, 2, \ldots, K$):*

- $\{(\underline{\mathrm{AC}}_1(L), \underline{\mathrm{AC}}_2(L), \ldots, \underline{\mathrm{AC}}_K(L)) \mid L \subseteq \Omega^*$ *and* L *is infinite*$\}$;
- $\{(\underline{\mathrm{AC}}_1(L), \underline{\mathrm{AC}}_2(L), \ldots, \underline{\mathrm{AC}}_K(L)) \mid L \subseteq \Omega^\infty$ *and* $L \neq \varnothing\}$;
- $\left\{\left(\underline{\underline{\mathrm{AC}}}_1(L), \underline{\underline{\mathrm{AC}}}_2(L), \ldots, \underline{\underline{\mathrm{AC}}}_K(L)\right) \mid L \subseteq \Omega^\infty \text{ and } L \neq \varnothing\right\}$.

The upper $\leq$-closure of the $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_k$-entropy hull coincides with the following sets:

- $\left\{\left(\overline{\mathrm{AC}}_1(L), \overline{\mathrm{AC}}_2(L), \ldots, \overline{\mathrm{AC}}_K(L)\right) \mid L \subseteq \Omega^* \text{ and } L \text{ is infinite}\right\}$;
- $\left\{\left(\overline{\mathrm{AC}}_1(L), \overline{\mathrm{AC}}_2(L), \ldots, \overline{\mathrm{AC}}_K(L)\right) \mid L \subseteq \Omega^\infty \text{ and } L \neq \varnothing\right\}$;
- $\left\{\left(\overline{\overline{\mathrm{AC}}}_1(L), \overline{\overline{\mathrm{AC}}}_2(L), \ldots, \overline{\overline{\mathrm{AC}}}_K(L)\right) \mid L \subseteq \Omega^\infty \text{ and } L \neq \varnothing\right\}$.

If the games $\mathfrak{G}_1, \mathfrak{G}_2 \ldots, \mathfrak{G}_K$ are computable and computably very weakly mixable, the same holds for effective complexities. If the games $\mathfrak{G}_1, \mathfrak{G}_2 \ldots, \mathfrak{G}_K$ are polynomial-time computable and polynomial-time computably very weakly mixable, the same holds for polynomial-time complexities.

In Section 4.2 we illustrate the statement of the theorem by drawing entropy hulls and their closures in two dimensions where entropy hulls are easy to visualise. In most natural cases (e.g., if the games involved are the square-loss, absolute-loss, or logarithmic) the entropy hull is a sublattice. However this is not true of all games as shown by an example in Section 4.2 so the closures cannot be dropped from the statement of the theorem.

Section 5.1 discusses shapes of the entropy hull in the general case. According to Corollary 22, the $\leq$-closure and the upper $\leq$-closure of an entropy hull are compact and convex sets.

[5] The statement of the main theorem in the conference version of this paper was inaccurate: it claimed that all the sets coincided with the upper $\leq$-closure. Comments for Lemma 34 and Remark 31 uncover an omission that led to the inaccuracy.

Let us discuss some other requirements in the statement of the theorem. The requirement of weak mixability cannot be omitted. For example, consider the simple prediction game $\langle \mathbb{B}, \mathbb{B}, \lambda \rangle$, where $\lambda(\omega, \gamma)$ is 0 if $\omega = \gamma$ and 1 otherwise. The convex hull of the set of superpredictions w.r.t. the simple prediction game coincides with the set of superpredictions w.r.t. the absolute-loss game. Geometric considerations imply that their generalised entropies coincide. Thus the maximum of the generalised entropy w.r.t. the simple prediction game is 1/2 (see Section 4.2). On the other hand, it is easy to check that $\mathrm{AC}(\mathbb{B}^*) = 1$, where AC is any of the asymptotic complexities w.r.t. the simple prediction game.

The statement of the theorem does not apply to pairs $(\overline{\mathrm{AC}}_1(L), \underline{\mathrm{AC}}_2(L))$ or pairs $(\overline{\overline{\mathrm{AC}}}_1(L), \underline{\underline{\mathrm{AC}}}_2(L))$. Indeed, let $\mathfrak{G}_1 = \mathfrak{G}_2$. Then $H_1 = H_2$ and the entropy hull with its $\leq$-closure are subsets of the bisector of the first quadrant. However we know that upper and lower complexities differ and thus there will be pairs outside the bisector.

The theorem is proven in Section 5.

4.2 Entropy Hulls in Two Dimensions

In this section we consider entropy hulls of two games and construct an example showing that the entropy hull is not necessarily an upper subsemilattice.

For planar sets the following simple criterion holds.

Proposition 7. *A compact convex set $\mathcal{U} \subseteq \mathbb{R}^2$ is an upper subsemilattice if and only if $\max \mathcal{U} \in \mathcal{U}$. A compact convex set $\mathcal{S} \subseteq \mathbb{R}^2$ is*

- *an upper subsemilattice if and only if $\max \mathcal{S} \in \mathcal{S}$;*
- *a sublattice if and only if $\max \mathcal{S}, \min \mathcal{S} \in \mathcal{S}$.*

Proof. The only if part is trivial: the componentwise maximums and minimums are achieved on some points because of compactness and their least upper and greatest lower bounds belong to corresponding subsemilattices.

The if part follows from convexity. Let $x^* = \max \mathcal{U} \in \mathcal{U}$. Take $u, v \in \mathcal{U}$. The triangle with the vertices u, v, and x^* is a subset of $\mathcal{U}$ and $\max(u, v)$ is in the triangle. □

Remark 8. This criterion does not hold in higher dimensions. Let $\mathcal{M} \subseteq \mathbb{R}^3$ be the triangle with the vertices $(1, 0, 0)$, $(0, 1, 0)$ and $(1, 1, 1)$. The set $\mathcal{M}$ is convex and compact and it contains its maximum $(1, 1, 1)$. However

$$\max((1, 0, 0), (0, 1, 0)) = (1, 1, 0) \notin \mathcal{M} \ .$$

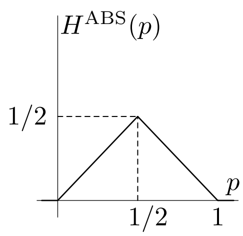

Figure 1: The graph of the ABS-entropy

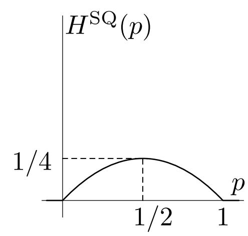

Figure 2: The graph of the SQ-entropy

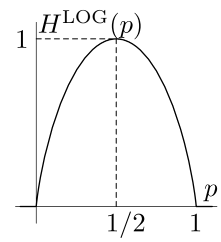

Figure 3: The graph of the LOG-entropy

Corollary 9. *Let $\mathfrak{G}_1$ and $\mathfrak{G}_2$ be two games with the same number of possible outcomes; let H_1 be the generalised entropy w.r.t. $\mathfrak{G}_1$ and H_2 be the generalised entropy w.r.t. $\mathfrak{G}_2$. Then the $\mathfrak{G}_1/\mathfrak{G}_2$-entropy hull is*

1. *an upper subsemilattice if and only if*

$$\arg\max H_1 \cap \arg\max H_2 \neq \varnothing \ ;$$

2. *a sublattice if and only if*

$$\arg\max H_1 \cap \arg\max H_2 \neq \varnothing \ ,$$
$$\arg\min H_1 \cap \arg\min H_2 \neq \varnothing \ .$$

The corollary relies on Corollary 22, which will be proven later.

The rest of this section contains examples of entropy sets and hulls in two dimensions.

It is easy to check by direct calculation that the entropy for the absolute-loss game is given by $H^{\mathrm{ABS}}(p) = \min(p, 1-p)$, the entropy for the square-loss game is given by $H^{\mathrm{SQ}}(p) = p(1-p)$, and the entropy for the logarithmic game is given by $H^{\mathrm{LOG}}(p) = -p\log_2 p - (1-p)\log_2(1-p)$, and thus it coincides with Shannon entropy. The graphs of the entropies are shown in Figures 1, 2, and 3. The entropy hulls for the pairs of games are shown on Figures 4, 5, and 6; the corresponding entropy sets are represented by bold lines. Since all the three games are symmetric, it should come as no surprise that the entropy hulls are sublattices in $\mathbb{R}^2$.

Let us construct an entropy hull that is a not an upper subsemilattice. It follows from Corollary 9 that the example must be rather artificial. Let $\mathfrak{G}_1 = \langle \mathbb{B}, [0,1], \lambda_1 \rangle$, where

$$\lambda_1(\omega, \gamma) = \begin{cases} \gamma, & \text{if } \omega = 0 \\ 1 - \frac{\gamma}{2}, & \text{if } \omega = 1, \end{cases}$$

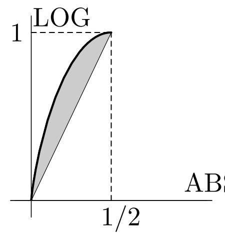

Figure 4: The ABS/LOG-entropy set and hull

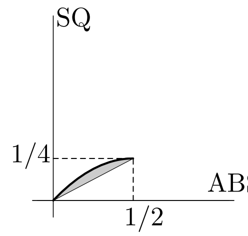

Figure 5: The ABS/SQ-entropy set and hull

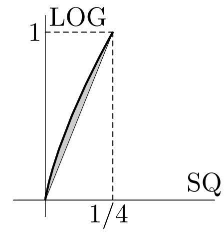

Figure 6: The SQ/LOG-entropy set and hull

and let $\mathfrak{G}_2 = \langle \mathbb{B}, [0,1], \lambda_2 \rangle$, where

$$\lambda_1(\omega,\gamma) = \begin{cases} 1 + \frac{\gamma}{2}, & \text{if } \omega = 0 \\ \frac{3}{2} - \gamma, & \text{if } \omega = 1. \end{cases}$$

The corresponding entropies are as follows:

$$\begin{aligned} H_1(p) &= \min_{\gamma\in[0,1]} \left(p\left(1 - \frac{\gamma}{2}\right) + (1-p)\gamma \right) \\ &= \min_{\gamma\in[0,1]} \left(\gamma\left(1 - \frac{3}{2}p\right) + p \right) \\ &= \min\left(p, 1 - \frac{p}{2}\right) \end{aligned}$$

(the last equality holds because the minimum of a linear function is achieved at either $\gamma = 0$ or $\gamma = 1$) and

$$\begin{aligned} H_2(p) &= \min_{\gamma\in[0,1]} \left(p\left(\frac{3}{2} - \gamma\right) + (1-p)\left(1 + \frac{\gamma}{2}\right) \right) \\ &= \min_{\gamma\in[0,1]} \left(\gamma\left(\frac{1}{2} - \frac{3}{2}p\right) + 1 + \frac{p}{2} \right) \\ &= \min\left(1 + \frac{p}{2}, \frac{3}{2} - p\right) \quad . \end{aligned}$$

The graph of H_1 consists of two line segments joining at $p = 2/3$ and the graph of H_2 consists of two line segments joining at $p = 1/3$. Thus the entropy set is the broken line passing through the points $(0,1)$, $\left(\frac{1}{3}, \frac{7}{6}\right)$, $\left(\frac{2}{3}, \frac{5}{6}\right)$, and $\left(\frac{1}{2}, \frac{1}{2}\right)$. Figure 7 shows the $\mathfrak{G}_1/\mathfrak{G}_2$-entropy hull, which is not an upper subsemilattice. Figure 8 shows the upper $\leq$-closure, and Figure 9 shows its $\leq$-closure.

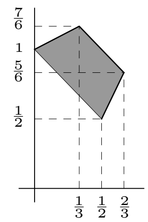

Figure 7: The $\mathfrak{G}_1/\mathfrak{G}_2$-entropy hull

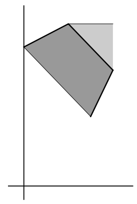

Figure 8: The upper $\leq$-closure

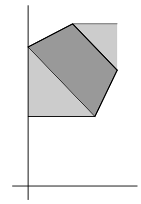

Figure 9: The $\leq$-closure

5 Proof of the Main Theorem

In this section we prove Theorem 6. We start with a discussion of entropy hulls and their geometric properties in Section 5.1. In Section 5.2 we obtain a lemma of independent interest about the optimisation of prediction strategies. Then in Section 5.3 we proceed to show that every tuple of complexities belongs to an appropriate closure of the entropy hull and in Section 5.4 we complete the proof by showing that every point in the closure corresponds to the tuple of complexities of some language.

5.1 Shapes of Entropy Hulls

In this subsection we discuss geometric aspects of entropy hulls and lattices. The results of this subsection clarify the statement of the main theorem and will be used in the proof.

It follows from our definition that every game has a prediction leading to finite losses. This implies that generalised entropy is finite and bounded. It can also be shown to be continuous.

Proposition 10. *Generalised entropy is a continuous function on its domain.*

Proof. Let $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ be a game with M outcomes. Let us show that the entropy $H : \mathbb{P}_M \to \mathbb{R}$ is a continuous function.

First let λ be bounded and $L > 0$ be such that $\lambda(\omega, \gamma) \leq L$ for all $\omega \in \Omega, \gamma \in \Gamma$. For all distributions $p_1 = \left(p_1^{(0)}, p_1^{(1)}, \ldots, , p_1^{(M-1)}\right)$ and $p_2 =$

$\left(p_2^{(0)}, p_2^{(1)}, \ldots, , p_2^{(M-1)}\right)$ from $\mathbb{P}_M$ we have

$$\begin{aligned}
H(p_2) &= \min_{\gamma\in\Gamma} \sum_{i=0}^{M-1} p_2^{(i)} \lambda(\omega^{(i)}, \gamma) \\
&= \min_{\gamma\in\Gamma} \sum_{i=0}^{M-1} \left(p_1^{(i)} \lambda(\omega^{(i)}, \gamma) + \left(p_2^{(i)} - p_1^{(i)}\right) \lambda(\omega^{(i)}, \gamma)\right) \\
&\leq H(p_1) + L \sum_{i=0}^{M-1} \left|p_2^{(i)} - p_1^{(i)}\right| \quad .
\end{aligned}$$

The continuity follows.

In order to obtain continuity for the unbounded case we need Lemma 15 from [KV08]. We include its statement for completeness.

Lemma 11. *For every game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ and $\varepsilon > 0$ there is a number L_ε with the following property. For every $\gamma \in \Gamma$ there is $\gamma^* \in \Gamma$ such that $\lambda(\omega, \gamma^*) \leq L_\varepsilon$ and $\lambda(\omega, \gamma^*) \leq \lambda(\omega, \gamma) + \varepsilon$ for all $\omega \in \Omega$.*

For $\varepsilon > 0$ let $\Gamma_\varepsilon = \{\Omega, \Gamma, \lambda_\varepsilon\}$, where $\lambda_\varepsilon(\omega, \gamma) = \min(\lambda(\omega, \gamma), L_\varepsilon)$, and H_ε be the generalised entropy for $\mathfrak{G}_\varepsilon$. Since $\lambda_\varepsilon \leq \lambda$ everywhere, we get $H_\varepsilon \leq H$. On the other hand, $H \leq H_\varepsilon + \varepsilon$ by the definition of L_ε. The functions H_ε thus uniformly converge to H as $\varepsilon \to 0$ and the continuity of each H_ε implies the continuity of H. □

Since the simplex $\mathbb{P}_M$ is compact, every entropy set is a compact set (i.e., bounded and closed) w.r.t. the standard Euclidean topology. The same holds for the entropy hull:

Corollary 12. *The entropy hull is a compact set w.r.t. the standard Euclidean topology.*

Proof. The convex hull of a compact set is compact (see, e.g., [Egg58], Theorem 10). □

We need to show that the same applies to its upper and lower $\leq$-closures. We will obtain a number of properties of $\leq$-closures first.

Proposition 13. *Let $\mathcal{S}$ be the $\leq$-closure and $\mathcal{U}$ be the upper $\leq$-closure of a set $\mathcal{M} \subseteq \mathbb{R}^K$. Then*

1. *if $\mathcal{M}$ is bounded, then $\mathcal{S}$ and $\mathcal{U}$ are bounded;*

2. *if $\mathcal{M}$ is a lower subsemilattice, then $\mathcal{U}$ is a lower subsemilattice;*

3. if $\mathcal{M}$ is compact in $\mathbb{R}^K$, then $\mathcal{S}$ and $\mathcal{U}$ are compact in $\mathbb{R}^K$;

4. if $\mathcal{M}$ is convex, then $\mathcal{S}$ and $\mathcal{U}$ are convex.

Similar statements can be formulated for the lower $\leq$-closure but we skip them for brevity.

Proof of Proposition 13. Part 1 of the proposition is trivial. If a coordinate of points of $\mathcal{M}$ is bounded, the corresponding coordinate of points in $\mathcal{S}$ and $\mathcal{U}$ has the same bound.

For the rest of the proof we need the following technical statements.

Lemma 14. *Let $\mathcal{M} \subseteq \mathbb{R}^K$. Then its upper $\leq$-closure coincides with the set $U = \{\max(x_1, x_2, \dots, x_K) \mid x_1, x_2, \dots, x_K \in \mathcal{M}\}$.*

Proof of Lemma 14. The set U is contained in the upper $\leq$-closure of $\mathcal{M}$. Let us show that it is an upper subsemilattice. Take two points $u, v \in U$. There are $y_1, y_2, \dots, y_K \in \mathcal{M}$ and $z_1, z_2, \dots, z_K \in \mathcal{M}$ such that $u = \max(y_1, y_2, \dots, y_K)$ and $v = \max(z_1, z_2, \dots, z_K)$. The maximum $\max(u, v)$ has K components and each one is contributed by one of the vectors y_i or z_i. Therefore we can select K vectors from the set $\{y_1, y_2, \dots, y_K, z_1, z_2, \dots, z_K\}$ such that their componentwise maximum coincides with $\max(u, v)$. □

Remark 15. It is not sufficient to take a number less than K. Indeed, consider $\mathcal{M}$ consisting of K unit vectors $e_1, e_2, \dots, e_K$ (all coordinates of the vector e_i equal zero except the i-th, which equals 1). Their least upper bound $(1, 1, \dots, 1)$ can only be obtained as the maximum of all K of them.

Lemma 16. *For all vectors $x_{i,j}$ ($i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$) of the same dimension*

$$\min(\max(x_{1,1}, x_{1,2} \dots, x_{1,m}), \dots, \max(x_{n,1}, x_{n,2} \dots, x_{n,m})) = \\ \max\{\min(x_{1,i_1}, x_{2,i_2} \dots, x_{n,i_n}) \mid i_1, i_2, \dots, i_n = 1, 2, \dots, m\} \ .$$

Proof of Lemma 16. It is sufficient to prove the lemma for one-dimensional vectors, i.e., real numbers. A number x can be identified with the half-line $(-\infty, x]$ so that the half-line corresponding to $\max(x, y)$ is the union and the half-line corresponding to $\min(x, y)$ is the intersection of the half-lines corresponding to x and y. The statement of the lemma thus follows from the distributivity of $\cup$ and $\cap$. □

Remark 17. This lemma essentially states that the lattice $\mathbb{R}^K$ with the relation $\leq$ is distributive (see [Bir48], Chapter IX).

Now we can prove Part 2. Let $\mathcal{M}$ be a lower subsemilattice and $\mathcal{U}$ be its upper $\leq$-closure. Let $u, v \in \mathcal{U}$. In order to show that $\mathcal{U}$ is a lower subsemilattice, we need to prove that $\min(u, v) \in \mathcal{U}$. Lemma 14 implies that $u = \max(y_1, y_2, \dots, y_K)$ and $v = \max(z_1, z_2, \dots, z_K)$ for some $y_1, y_2, \dots, y_k, z_1, z_2, \dots, z_K \in \mathcal{M}$. It follows from Lemma 16 that $\min(u, v)$ can be represented as the maximum of minimums of some y_i and z_i. Since $\mathcal{M}$ is a lower subsemilattice, the minimums belong to $\mathcal{M}$ and since $\mathcal{U}$ is an upper subsemilattice, their maximum belongs to $\mathcal{U}$. Part 2 follows.

Corollary 18. *The $\leq$-closure of a set equals each of the following:*

- *the upper $\leq$-closure of its lower $\leq$-closure;*
- *the lower $\leq$-closure of its upper $\leq$-closure.*

Proof. The lower $\leq$-closure $\mathcal{L}$ of a set $\mathcal{M}$ is a subset of $\mathcal{S}$, the $\leq$-closure of $\mathcal{M}$. The upper $\leq$-closure $\mathcal{U}$ of $\mathcal{L}$ is a subset of $\mathcal{S}$ too. We have shown that $\mathcal{U}$ is a lower subsemilattice, and therefore a sublattice. Thus it coincides with $\mathcal{S}$. $\square$

We need another technical lemma.

Lemma 19. *Let vectors $x_{i,j}$ ($i = 1, 2, \dots, n$ and $j = 1, 2, \dots, \infty$) be of the same dimension and such that $\lim_{j\to\infty} x_{i,j}$ exists for all $i = 1, 2, \dots, n$. Then*

$$\lim_{j\to\infty} \max\{x_{i,j} \mid i = 1, 2, \dots, n\} = \max\{\lim_{j\to\infty} x_{i,j} \mid i = 1, 2, \dots, n\} \ .$$

Proof of Lemma 19. It is sufficient to prove the lemma for one-dimensional vectors. Let $M = \max\{\lim_{j\to\infty} x_{i,j} \mid i = 1, 2, \dots, n\}$. Those sequences $x_{i,j}, j = 1, 2, \dots$ that converge to numbers less than M will go below M from some j_0 on and will not contribute to $\max\{x_{i,j} \mid i = 1, 2, \dots, n\}$ for large j. The sequences converging to M will be within an ε-vicinity of M from some j_ε on and so will be $\max\{x_{i,j} \mid i = 1, 2, \dots, n\}$. Thus the maximum will converge to M. $\square$

Let us prove Part 3 of Proposition 13. Let $u_1, u_2, \dots \in \mathcal{U}$. We need to show that u_j has a subsequence converging to a limit in $\mathcal{U}$.

For each u_j there are points $x_{i,j} \in \mathcal{M}$, $i = 1, 2, \dots, K$, such that $u_j = \max\{x_{i,j} \mid i = 1, 2, \dots, K\}$. Since $\mathcal{M}$ is compact, the sequence $x_{1,m}$, $m = 1, 2, \dots$, has a converging subsequence x_{1,m_k}, $k = 1, 2, \dots$. The sequence x_{2,m_k}, $k = 1, 2, \dots$, in turn has a converging subsequence etc. Arguing in this way, we obtain a sequence of indices $j_1 < j_2 < \dots$ such that for all

$i = 1, 2, \ldots, K$ the sequence x_{i,j_n}, $n = 1, 2, \ldots$, converges to a limit in $\mathcal{M}$. Lemma 19 implies that $u_{j_n} = \max\{x_{i,j_n} \mid i = 1, 2, \ldots, K\}$ converges to a limit in $\mathcal{M}$.

We have shown that the set $\mathcal{U}$ is compact. The set $\mathcal{S}$ is compact as the lower $\leq$-closure of the compact $\mathcal{U}$.

Remark 20. Note that topological closeness of $\mathcal{M}$ in $\mathbb{R}^k$ does not imply closeness of $\mathcal{U}$. Indeed, let $\mathcal{M} = \{(x, y) \mid x, y < 0, xy \geq 1\} \subseteq \mathbb{R}^2$. It is a closed subset of $\mathbb{R}^2$. However its upper $\leq$-closure $\mathcal{U} = \{(x, y) \mid x, y < 0\}$ is not closed.

We need another technical lemma.

Lemma 21. *For all $x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n$ of the same dimension*

$$\max(x_1, x_2, \ldots, x_n) + \max(y_1, y_2, \ldots, y_n) = \max\{x_i + y_j \mid i, j = 1, 2, \ldots, n\} \ .$$

Proof of Lemma 21. It is sufficient to prove the lemma for one-dimensional vectors. The expression on the left-hand side is greater than or equal to each of the sums $x_i + y_j$ and equals one of them. □

In order to prove Part 4 of Proposition 13 let $u_1, u_2 \in \mathcal{U}$ and $\alpha \in (0, 1)$. We need to show that $\alpha u_1 + (1 - \alpha)u_2 \in \mathcal{U}$. There are points $x_1, x_2, \ldots, x_K$, $y_1, y_2, \ldots, y_K \in \mathcal{M}$ such that

$$u_1 = \max(x_1, x_2, \ldots, x_K) \ ,$$
$$u_2 = \max(y_1, y_2, \ldots, y_K) \ .$$

We have

$$\begin{aligned}\alpha u_1 + (1 - \alpha)u_2 = \alpha \max(x_1, x_2, \ldots, x_K) + (1 - \alpha) \max(y_1, y_2, \ldots, y_K) = \\ \max(\alpha x_1, \alpha x_2, \ldots, \alpha x_K) + \max((1 - \alpha)y_1, (1 - \alpha)y_2, \ldots, (1 - \alpha)y_K) = \\ \max\{\alpha x_i + (1 - \alpha)y_j \mid i, j = 1, 2, \ldots, K\} \ .\end{aligned}$$

Since all convex combinations of x_i and y_j belong to $\mathcal{M}$, this belongs to $\mathcal{U}$. Therefore $\mathcal{U}$ is convex and $\mathcal{S}$ is convex as the lower $\leq$-closure of the convex $\mathcal{U}$. □

Corollary 22. *The $\leq$-closure, the lower $\leq$-closure, and the upper $\leq$-closure of the entropy hull are compact and convex sets.*

The following statements describe relations of closures and projections.

For a set $\mathcal{M} \subseteq \mathbb{R}^K$ and pairwise different indices i_n, $1 \le i_n \le K$, $n = 1, 2, \ldots, N$, *the projection of* $\mathcal{M}$ *onto the coordinate hyperplane* corresponding to coordinates $i_1, i_2, \ldots, i_N$ is the set of points $x = \left(x^{(1)}, x^{(2)}, \ldots, x^{(N)}\right)$ such that there is $y = \left(y^{(1)}, y^{(2)}, \ldots, y^{(K)}\right) \in \mathcal{M}$ with $x^{(n)}$ in position i_n, i.e., such that $y^{(i_n)} = x^{(n)}$, $n = 1, 2, \ldots$.

Sometimes we will be speaking of the projection as of the set of $x = \left(x^{(1)}, x^{(2)}, \ldots, x^{(K)}\right)$ such that $x^{(i_n)} = y^{(i_n)}$, $n = 1, 2, \ldots, N$, and $x^{(i)} = 0$ for other indices.

The following simple lemma contains important properties of projections.

Lemma 23. *For every set* $\mathcal{M} \subseteq \mathbb{R}^K$ *and its projection* $\widetilde{\mathcal{M}}$ *onto a coordinate hyperplane*

1. *the convex hull of* $\widetilde{\mathcal{M}}$ *is the projection of the convex hull of* $\mathcal{M}$*;*
2. *the upper* $\le$*-closure of* $\widetilde{\mathcal{M}}$ *is the projection of the upper* $\le$*-closure of* $\widetilde{\mathcal{M}}$*;*
3. *the* $\le$*-closure of* $\widetilde{\mathcal{M}}$ *is the projection of the* $\le$*-closure of* $\widetilde{\mathcal{M}}$*.*

Proof. Part 1 follows from a representation of the convex hull of $\mathcal{M}$ as the set of linear combinations $\sum_{m=1}^{M} \alpha_m x_m$, where $x_m \in \mathcal{M}$ and $\alpha_m \in [0, 1]$, $m = 1, 2, \ldots, M$, and $\sum_{m=1}^{M} \alpha_m = 1$. (Indeed, each combination belongs to the convex hull and they all form a convex set.)

Part 2 follows from the representation of the upper closure provided by Lemma 14. Clearly, a similar statement holds for the lower closure.

Part 3 follows from Part 2 and Corollary 18. □

Corollary 24. *Take* $K \ge 2$ *games* $\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_K$ *with the same set of outcomes* Ω*. Then*

1. *the* $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_{K-1}$*-entropy set is the projection of the* $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$*-entropy set,*
2. *the* $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_{K-1}$*-entropy hull is the projection of the* $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$*-entropy hull,*
3. *the upper* $\le$*-closure of the* $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_{K-1}$*-entropy hull is the projection of the upper* $\le$*-closure of the* $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$*-entropy hull, and*
4. *the* $\le$*-closure of the* $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_{K-1}$*-entropy hull is the projection of the* $\le$*-closure of the* $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$*-entropy hull,*

where all projections are taken onto the coordinate hyperplane corresponding to the first $K-1$ coordinates.

Proposition 7 implies that constructing the $\le$-closures and upper $\le$-closures of compact convex two-dimensional sets is easy: one should add the coordinate-wise maximum and minimum (or just the maximum) to the set and take the convex hull.

Construction of $\le$-closures in more than two dimensions can be reduced to the two-dimensional case.

Proposition 25. *Let $\mathcal{M} \subseteq \mathbb{R}^K$ and $K \ge 2$. Then its $\le$-closure $\mathcal{S}$ equals $\bigcap_{1\le i<j\le K} C_{ij}$, where $C_{ij} \subseteq \mathbb{R}^K$ is the cylinder over the $\le$-closure of the projection of $\mathcal{M}$ onto the coordinate plane corresponding to coordinates i and j.*

Proof. Let $C = \bigcap_{1\le i<j\le K} C_{ij}$. First, note that each C_{ij} is a lattice w.r.t. $\le$ and so is their intersection. Thus $\mathcal{S} \subseteq C$.

We will prove that $C \subseteq \mathcal{S}$ using induction in K. The case $K = 2$ is trivial. Suppose that the statement is true in dimension $K-1$ and take $x = \left(x^{(1)}, x^{(2)}, \ldots, x^{(K)}\right) \in C$. Let $\widetilde{\mathcal{M}} \subseteq \mathbb{R}^{K-1}$ be the projection of $\mathcal{M}$ onto the hyperplane corresponding to the first $K-1$ coordinates and let $\tilde{x}$ be the projection of x (i.e., x without the last coordinate). The point $\tilde{x}$ belongs to the intersection of cylinders over closures of two-dimensional projections of $\widetilde{\mathcal{M}}$. By the inductive hypotheses it belongs to the $\le$-closure $\widetilde{\mathcal{S}}$ of $\widetilde{\mathcal{M}}$. By Lemma 23, $\widetilde{\mathcal{S}}$ is the projection of $\mathcal{S}$ and hence $\tilde{x}$ is the projection of some $u \in \mathcal{S}$. The first $K-1$ coordinates of u coincide with those of x. Let the last coordinate of u be $u^{(K)}$. If $u^{(K)} = x^{(K)}$, there is nothing more to prove. Let $u^{(K)} \ne x^{(K)}$.

Since $x \in C$, we get $x \in C_{iK}$ for $i = 1, 2, \ldots, K-1$. Therefore the projection $\bar{x}_i$ of x onto the coordinate plane corresponding to coordinates i and K belongs to the $\le$-closure of the corresponding projection of $\mathcal{M}$, i.e., to the projection of $\mathcal{S}$. There is $x_i \in \mathcal{S}$ that coincides with x in coordinates i and K.

Now we need to consider two cases.

Case $u^{(K)} > x^{(K)}$. Take $y = \max(x_1, x_2, \ldots, x_{K-1}) \in \mathcal{S}$. The last coordinate of y equals $x^{(K)}$ and all other coordinates are greater than or equal to corresponding coordinates of x.

The point $\min(y, u)$ belongs to $\mathcal{S}$ and equals x. Thus $x \in \mathcal{S}$.

Case $u^{(K)} < x^{(K)}$. Similarly, $x = \max(\min(x_1, x_2, \ldots, x_{K-1}), u) \in \mathcal{S}$.

□

Remark 26. This proposition cannot be extended to one-sided closures. Take $K = 3$ and let $\mathcal{M}$ be the triangle (with its interior) having the vertices $(1,1,0)$, $(1,0,1)$, and $(0,1,1)$. The upper $\leq$-closure of $\mathcal{M}$ is the set of points $(x,y,z) \in [0,1]^3$ such that $x+y+z \geq 2$. The projection of $\mathcal{M}$ onto each two-dimensional coordinate plane is the triangle with the vertices $(0,1)$, $(1,0)$, and $(1,1)$. It is easy to see that the intersection of cylinders over these triangles contains the point $(0.5, 0.5, 0.5)$. However the point does not belong to the upper $\leq$-closure of $\mathcal{M}$.

Let $\mathcal{U} \subseteq \mathbb{R}^K$ be a compact convex upper subsemilattice. Its projection onto any coordinate hyperplane is a compact convex upper subsemilattice too. Let $\mathcal{U}_1 \subseteq \mathbb{R}^{K-1}$ be the projection of $\mathcal{U}$ onto the hyperplane corresponding to the first $K-1$ coordinates, i.e.,

$$\mathcal{U}_1 = \left\{ \left(x^{(1)}, x^{(2)}, \ldots, x^{(K-1)} \right) \mid \left(x^{(1)}, x^{(2)}, \ldots, x^{(K-1)}, x^{(k)} \right) \in \mathcal{U} \text{ for some } x^{(k)} \right\} \ . \quad (9)$$

Define the function $\varphi : \mathcal{U}_1 \to \mathbb{R}$ by

$$\varphi\left(x^{(1)}, x^{(2)}, \ldots, x^{(K-1)} \right) = \max\left\{ x^{(k)} \mid \left(x^{(1)}, x^{(2)}, \ldots, x^{(K-1)}, x^{(k)} \right) \in \mathcal{U} \right\} \ ; \quad (10)$$

this function parametrises the upper part of the boundary of $\mathcal{U}$.

Proposition 27. *For every compact convex upper subsemilattice $\mathcal{U}$ let the set $\mathcal{U}_1$ and the function $\varphi : \mathcal{U}_1 \to \mathbb{R}$ be defined by (9) and (10). Then φ has the following properties:*

1. *φ is concave on $\mathcal{U}_1$;*
2. *φ is monotone, i.e., for all $u, v \in \mathcal{U}_1$ if $u \leq v$ then $\varphi(u) \leq \varphi(v)$;*
3. *φ achieves its maximum at the point*
$$x^* = \max \mathcal{U}_1 = \left(x_*^{(1)}, x_*^{(2)}, \ldots, x_*^{(K-1)} \right) \ ,$$
where $x_*^{(i)}$ is the maximum of the i-th coordinate of points in $\mathcal{U}_1$, $i = 1, 2, \ldots, K-1$;
4. *for every sequence $u_1, u_2, \ldots \in \mathcal{U}_1$ converging to u_0 such that $u_0 \leq u_i$ for all $i = 1, 2, \ldots$ we have $\lim_{i\to\infty} \varphi(u_i) = \varphi(u_0)$.*

Proof. Part 1 is trivial. To prove Part 2 consider the two points $(u, \varphi(u))$ and $(v, \varphi(v)) \in \mathcal{U}$. Since $\mathcal{U}$ is an upper subsemilattice,

$$\max((u, \varphi(u)), (v, \varphi(v))) = (v, \max(\varphi(u), \varphi(v))) \in \mathcal{U}$$

and the definition of φ implies that

$$\varphi(v) \geq \max(\varphi(u), \varphi(v)) \geq \varphi(u) \ .$$

In order to prove Part 3 we first need to show that $x^* \in \mathcal{U}_1$. Since $\mathcal{U}_1$ is compact, the maximum of the i-th coordinate is achieved on some $u_i \in \mathcal{U}$ and since $\mathcal{U}$ is an upper subsemilattice, $\max(u_1, u_2, \ldots, u_{K-1}) = x^*$ belongs to $\mathcal{U}_1$. For every $u \in \mathcal{U}_1$ we have $u \leq x^*$ and therefore $\varphi(u) \leq \varphi(x^*)$.

Let us prove Part 4. Part 2 implies that $\varphi(u_0) \leq \varphi(u_i)$ for all $i = 1, 2, \ldots$. If $\varphi(u_i)$ do not converge to $\varphi(u_0)$, there is $\delta > 0$ such that $\varphi(u_{i_k}) \geq \varphi(u_0) + \delta$ for an infinite sequence $i_1 < i_2 < i_3 < \ldots$. However since $\mathcal{U}$ is compact, the sequence $(u_{i_k}, \varphi(u_{i_k}))$, $k = 1, 2, \ldots$, has a converging subsequence. It must converge to $(u_0, r) \in \mathcal{U}$ such that $r \geq \varphi(u_0) + \delta$. This contradicts the definition of $\varphi(u_0)$. □

Remark 28. Part 4 states a weak form of continuity for φ. By a classical theorem of convex analysis (e.g., Theorem 24 in [Egg58]) concavity of φ implies its continuity on the relative interior of $\mathcal{U}_1$. However the authors do not know if a form of continuity stronger than that claimed by Part 4 actually holds on the relative boundary of $\mathcal{U}_1$.

The following two simple axillary lemmas relate to closed (not necessarily bounded) semilattices.

Lemma 29. *Let $\mathcal{U}$ be a closed upper subsemilattice. Then for any bounded $\mathcal{M} \subseteq S$ the componentwise supremum* $\sup \mathcal{M}$ *belongs to S.*

Proof. Let $\sup \mathcal{M} = u = \left(u^{(1)}, u^{(2)}, \ldots, u^{(K)}\right)$. Take $\varepsilon > 0$. For every $k = 1, 2, \ldots, K$ there is $x_k \in \mathcal{M}$ with the k-th coordinate greater than or equal to $u^{(k)} - \varepsilon$. The maximum $x = \max(x_1, x_2, \ldots, x_K)$ belongs to $\mathcal{U}$ and $\|x - u\|^2 \leq K\varepsilon^2$, where $\|\cdot\|$ is the Euclidean norm. Since $\mathcal{U}$ is closed, we conclude that $u \in \mathcal{U}$. □

Lemma 30. *Let $\mathcal{U} \subseteq \mathbb{R}^K$ be a closed upper subsemilattice and*

$$u_i = \left(u_i^{(1)}, u_i^{(2)}, \ldots, u_i^{(K)}\right) \ ,$$

$i = 1, 2, \ldots$, *be a bounded sequence of points from* $\mathcal{U}$. *Then the componentwise upper limit*

$$u = \left(u^{(1)}, u^{(2)}, \ldots, u^{(K)}\right) \quad ,$$

where $u^{(k)} = \limsup_{i\to\infty} u_i^{(k)}$, $k = 1, 2, \ldots, K$, *belongs to* S.

Proof. Take $\varepsilon > 0$ and let $\mathcal{M}_\varepsilon$ be the set of all u_i such that $u_i^{(k)} \le u^{(k)} + \varepsilon$, $k = 1, 2, \ldots, K$. It contains all but finitely many points from the sequence. We have $u_\varepsilon = \sup \mathcal{M}_\varepsilon \ge u$. On the other hand,

$$u_\varepsilon \le \left(u^{(1)} + \varepsilon, u^{(2)} + \varepsilon, \ldots, u^{(K)} + \varepsilon\right)$$

and thus u_ε converges to u. By Lemma 29 u_ε belongs to $\mathcal{U}$ and since $\mathcal{U}$ is closed, $u \in \mathcal{U}$. □

Remark 31. Lemma 30 does not hold for lower limits. Indeed, let $\mathcal{U} = \{(x, y) \mid x, y \ge 0, x + y \ge 1\}$. It is a closed upper sublattice. Let $u_{2n} = (0, 1)$ and $u_{2n-1} = (1, 0)$, $n = 1, 2, \ldots$. The componentwise lower limit $(0, 0)$ does not belong to $\mathcal{U}$.

5.2 Recalibration Lemma

The following lemma allows us to "optimise" the performance of a strategy w.r.t. several games. We will call it the *recalibration lemma*.

Lemma 32. *Let* $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_K$ *(*$K \ge 1$*) be prediction strategies for games* $\mathfrak{G}_1, \mathfrak{G}_2 \ldots, \mathfrak{G}_K$*, respectively, with the same outcome space* Ω *of size* M *and let* $\varepsilon > 0$*. Then for every finite string* $\boldsymbol{x} \in \Omega^*$ *there are distributions* $p_1, p_2, \ldots, p_N \in \mathbb{P}_M$ *and* $q = (q_1, q_2, \ldots, q_N) \in \mathbb{P}_N$ *(*$N = N(\boldsymbol{x}, \varepsilon)$*,* $p_i = p_i(\boldsymbol{x}, \varepsilon)$*,* $i = 1, 2, \ldots, N$*, and* $q = q(\boldsymbol{x}, \varepsilon)$*) such that*

- *for all* $k = 1, 2, \ldots, K$

  $$\sum_{i=1}^{N} q_i H_k(p_i) \le \frac{\mathrm{Loss}_{\mathfrak{A}_k}^{\mathfrak{G}_k}(\boldsymbol{x})}{|\boldsymbol{x}|} + \varepsilon \quad ,$$

  where H_k *is the generalised entropy w.r.t.* $\mathfrak{G}_k$*;*

- *for every weakly mixable game* $\mathfrak{G}$ *there is a prediction strategy* $\mathfrak{S}$ *and a function* $f : \mathbb{N} \to \mathbb{R}$ *(*$\mathfrak{S} = \mathfrak{S}_{\mathfrak{G},\varepsilon}$ *and* $f = f_{\mathfrak{G},\varepsilon}$ *so they are independent of* $\boldsymbol{x}$*) such that* $f(n) = o(1)$ *as* $n \to \infty$ *and*

  $$\frac{\mathrm{Loss}_{\mathfrak{S}}^{\mathfrak{G}}(\boldsymbol{x})}{|\boldsymbol{x}|} \le \sum_{i=1}^{N} q_i H(p_i) + \varepsilon + f(|\boldsymbol{x}|) \quad ,$$

where H is the generalised entropy w.r.t. $\mathfrak{G}$.

If all the games $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_K$ and all the strategies $\mathfrak{G}_1, \mathfrak{G}_2 \ldots, \mathfrak{G}_K$ are (polynomial-time) computable, and $\mathfrak{G}$ is (polynomial-time) computable and (polynomial-time) computably very weakly mixable then $\mathfrak{S}$ can be chosen to be (polynomial-time) computable.

The idea behind the lemma can be described informally as follows. Consider a predictor outputting, say, the likelihood of rain. Suppose that by analysing its past performance we have found a pattern of the following kind. Whenever the predictor outputs the value of 70%, it actually rains in 90% of cases. We can thus improve the predictor by *recalibrating* it: if we see the prognosis of 70%, we replace it by 90%. Generally speaking, we may observe that whenever a predictor outputs a prediction γ_1, a more appropriate choice would be γ_2. By outputting γ_1, the predictor signals us about a specific state of the nature; however γ_2 is a better prediction for this state. The loss per element of the optimised strategy on outcomes preceded by γ_1 is close to the generalised entropy w.r.t. some distribution and the overall loss per element is their convex combination. This gives us the lower bound from the first part of the lemma. To obtain the second part of the lemma observe that we can use the data output by our predictor to predict other seemingly unrelated things. Suppose that whenever the predictor outputs the value of 70%, the stock market goes down in 60% of cases. We can make use of this fact and construct a stock market prediction strategy.

Another interpretation of the lemma[6] is as follows. Predictions of discretised strategies allow us to split a string into several (generally speaking, not contiguous) substrings. The discretised strategies tell us nothing of the behaviour of outcomes within the substrings so we can assume that inside each substring the outcomes are i.i.d. (independent identically distributed) and construct a new strategy exploiting this. The loss per element of the new strategy will be a convex combination of entropies w.r.t. the distributions of outcomes from the substrings and this gives us the second part of the lemma. By observing that the new strategy performs better or nearly as well as the original strategies we get the first part of the lemma.

These ideas will be implemented as follows. Given a pool of prediction strategies, we will discretise them and consider all possible mappings from the tuples of their possible outputs to predictions from a discrete set. We will then make use of weak mixability and merge the strategies generated by all the mappings.

[6] Along the lines suggested by a reviewer.

Proof. Let $\Omega = \{\omega^{(0)}, \omega^{(1)}, \ldots, \omega^{(M-1)}\}$. For every $k = 1, 2, \ldots, K$ let $\mathfrak{G}_k = \langle \Omega, \Gamma_k, \lambda_k \rangle$ and let H_k be the generalised entropy w.r.t. $\mathfrak{G}_k$; let $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ be a weakly mixable game.

The first part of this proof is the construction of $\mathfrak{S}$. First let us perform an ε-*quantisation* of $\mathfrak{A}_k$.

Lemma 33. *For any $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ and $\varepsilon > 0$ there is a finite set $\Gamma^{(\varepsilon)}$ such that for any $\gamma \in \Gamma$ there is $\gamma^* \in \Gamma^{(\varepsilon)}$ such that $\lambda(\omega, \gamma^*) \leq \lambda(\omega, \gamma) + \varepsilon$ for every $\omega \in \Omega$.*

Proof of Lemma 33. Lemma 11 (which is Lemma 15 from [KV08]) implies that it is sufficient to consider bounded loss functions λ. If λ is bounded, the lemma follows from continuity of λ and compactness of Γ. $\square$

Lemma 57 provides an effective version of this statement.

Let $\Gamma^{(\varepsilon)} \subseteq \Gamma$ and $\Gamma_k^{(\varepsilon)} \subseteq \Gamma_k$, $k = 1, 2, \ldots, K$, be such subsets. There are strategies $\mathfrak{A}_k^{(\varepsilon)}$ that output only predictions from $\Gamma_k^{(\varepsilon)}$ and such that $\mathrm{Loss}_{\mathfrak{A}_k^{(\varepsilon)}}^{\mathfrak{G}_k}(\boldsymbol{x}) \leq \mathrm{Loss}_{\mathfrak{A}_k}^{\mathfrak{G}_k}(\boldsymbol{x}) + \varepsilon|\boldsymbol{x}|$ for all $\boldsymbol{x} \in \Omega^*$, $k = 1, 2, \ldots, K$. By Lemma 57 if the games and strategies are (polynomial-time) computable, then $\mathfrak{A}_k^{(\varepsilon)}$ can be chosen to be (polynomial-time) computable.

The set $\Gamma_\varepsilon = \Gamma_1^{(\varepsilon)} \times \Gamma_2^{(\varepsilon)} \times \ldots \times \Gamma_K^{(\varepsilon)}$ is finite. Let $|\Gamma_\varepsilon| = L$. As we run the strategies $\mathfrak{A}_1^{(\varepsilon)}, \mathfrak{A}_2^{(\varepsilon)}, \ldots, \mathfrak{A}_K^{(\varepsilon)}$ on the same sequence of outcomes, we can say that on every step they collectively output one of the tuples $\gamma \in \Gamma_\varepsilon$.

For a mapping $\sigma : \Gamma_\varepsilon \to \Gamma^{(\varepsilon)}$ consider a strategy $\mathfrak{S}_\sigma$ that works as follows. It runs $\mathfrak{A}_1^{(\varepsilon)}, \mathfrak{A}_2^{(\varepsilon)}, \ldots, \mathfrak{A}_K^{(\varepsilon)}$ and when they produce $\gamma \in \Gamma_\varepsilon$ it outputs $\sigma(\gamma)$. There are finitely many ($\left|\Gamma^{(\varepsilon)}\right|^L$, to be precise) such mappings σ and strategies $\mathfrak{S}_\sigma$. Since $\mathfrak{G}$ is weakly mixable, there is a strategy $\mathfrak{S}$ that performs as well as any of them up to small o in the string length. If $\mathfrak{G}$ is (polynomial-time) computably very weakly mixable, then $\mathfrak{S}$ can be chosen to be (polynomial-time) computable at the cost of suffering possible extra loss not exceeding $\varepsilon|\boldsymbol{x}|$.

The second part of the proof consists of analysing the properties of $\mathfrak{S}$ and obtaining the desired loss bounds.

For a finite sequence $\boldsymbol{x}$ of length n, let $\gamma_1, \gamma_2, \ldots, \gamma_N$ ($N \leq L$ and $N \leq |\boldsymbol{x}|$), be an enumeration (without repetitions) of all elements of Γ_ε that are output at least once as strategies $\mathfrak{A}_k$ predict elements of $\boldsymbol{x}$. Let $n_j^{(m)}$ ($j = 1, 2, \ldots, N$ and $m = 0, 1, \ldots, M-1$) be the number of times when, while predicting elements of $\boldsymbol{x}$, the strategies $\mathfrak{A}_k$ output the tuple γ_j on a step

Table 1: Predictions and outcomes for a given sequence $\boldsymbol{x}$

Predictions	Number of $\omega^{(0)}$s	$\ldots$	Number of $\omega^{(M-1)}$s
γ_1	$n_1^{(0)}$	$\ldots$	$n_1^{(M-1)}$
γ_2	$n_2^{(0)}$	$\ldots$	$n_2^{(M-1)}$
$\vdots$	$\vdots$	$\ldots$	$\vdots$
γ_j	$n_j^{(0)}$	$\ldots$	$n_j^{(M-1)}$
$\vdots$	$\vdots$	$\ldots$	$\vdots$
γ_N	$n_N^{(0)}$	$\ldots$	$n_N^{(M-1)}$

when the outcome $\omega^{(m)}$ occurs. We get Table 1, where the j-th row contains the numbers $n_j^{(m)}$. Note that $\sum_{j=1}^N n_j^{(m)} = \sharp_m \boldsymbol{x}$ for all $m = 0, 1, \ldots, M-1$.

Let us construct an auxiliary strategy $\mathfrak{S}_{\boldsymbol{x}}$ for the game $\mathfrak{G}$. The strategy "knows" Table 1 and aims to predicts elements of $\boldsymbol{x}$ as well as it can on the basis of this information. It runs the strategies $\mathfrak{A}_k^{(\varepsilon)}$, $k = 1, 2, \ldots, K$, and then uses Table 1 to calibrate their output to predict $\boldsymbol{x}$.

If on some step they output $\gamma_j \in \Gamma_\varepsilon$, we know that we are on the j-th line of the table. We can use this information to optimise our performance. Let γ_j^* be an element of $\Gamma^{(\varepsilon)}$ where the minimum

$$\min_{\gamma\in\Gamma^{(\varepsilon)}} \sum_{m=0}^{M-1} n_j^{(m)} \lambda(\omega^{(m)}, \gamma) \tag{11}$$

is attained. The strategy $\mathfrak{S}_{\boldsymbol{x}}$ outputs γ_j^* each time $\mathfrak{A}_k^{(\varepsilon)}$ output γ_j. If $\mathfrak{A}_k^{(\varepsilon)}$ output some $\gamma \in \Gamma_\varepsilon$ not in the table (this cannot happen while predicting elements of $\boldsymbol{x}$ and therefore $\mathfrak{S}_{\boldsymbol{x}}$ is not particularly concerned), let $\mathfrak{S}_{\boldsymbol{x}}$ predict some fixed element of $\Gamma^{(\varepsilon)}$, say, γ_1.

Minimum (11) can be approximated using the generalised entropy H. Line j of the table specified a distribution on Ω. Put $p_j^{(m)} = n_j^{(m)} / \sum_{r=0}^{M-1} n_j^{(r)}$ (since $\mathfrak{A}_k^{(\varepsilon)}$ output γ_j at least once while predicting elements of $\boldsymbol{x}$, the denominator is not 0); the M-tuple $p_j = \left(p_j^{(0)}, p_j^{(1)}, \ldots, p_j^{(M-1)}\right)$ is a distribution on Ω. We have

$$\sum_{m=0}^{M-1} n_j^{(m)} \lambda_k(\omega^{(m)}, \gamma) = \left(\sum_{m=0}^{M-1} n_j^{(m)}\right) \sum_{m=0}^{M-1} p_j^{(m)} \lambda_k(\omega^{(m)}, \gamma)$$

and thus

$$\left(\sum_{m=0}^{M-1} n_j^{(m)}\right) H(p_j) \le \sum_{m=0}^{M-1} n_j^{(m)} \lambda_k(\omega^{(m)}, \gamma_j^*) \le \left(\sum_{m=0}^{M-1} n_j^{(m)}\right) (H(p_j) + \varepsilon) \tag{12}$$

(the former inequality holds because $H(p_j)$ is the minimum of the expectation and the latter inequality holds because the prediction where $H(p_j)$ is attained can be closely approximated by an element from $\Gamma^{(\varepsilon)}$).

The total loss of $\mathfrak{S}_{\boldsymbol{x}}$ on $\boldsymbol{x}$ is $\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{S}_{\boldsymbol{x}}}(\boldsymbol{x}) = \sum_{j=1}^{N} \sum_{m=0}^{M-1} n_j^{(m)} \lambda_k(\omega^{(m)}, \gamma_j^*)$. Put $q_j = \left(\sum_{m=0}^{M-1} n_j^{(m)}\right)/n$ so that $\sum_{j=1}^{N} q_j = 1$. Summing (12) over j yields

$$n \sum_{j=1}^{N} q_j H(p_j) \le \mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{S}_{\boldsymbol{x}}}(\boldsymbol{x}) \le n \sum_{j=1}^{N} q_j H(p_j) + n\varepsilon \ . \tag{13}$$

To obtain the second part of the statement of the lemma, note that the behaviour of $\mathfrak{S}_{\boldsymbol{x}}$ is identical to $\mathfrak{S}_\sigma$ for some mapping $\sigma : \Gamma_\varepsilon \to \Gamma^{(\varepsilon)}$ and use the second inequality in (13). For the (polynomial-time) computable case we need to replace ε by $\varepsilon/2$ from the start.

To get the first part of the statement of the theorem consider $\mathfrak{S}_{\boldsymbol{x}}$ for the game $\mathfrak{G} = \mathfrak{G}_k$. Both $\mathfrak{A}_k^{(\varepsilon)}$ and $\mathfrak{S}_{\boldsymbol{x}}$ use predictions from $\Gamma_k^{(\varepsilon)}$ but $\mathfrak{S}_{\boldsymbol{x}}$ does it in an optimal way on $\boldsymbol{x}$. Therefore

$$\mathrm{Loss}^{\mathfrak{G}_k}_{\mathfrak{S}_{\boldsymbol{x}}}(\boldsymbol{x}) \le \mathrm{Loss}^{\mathfrak{G}_k}_{\mathfrak{A}_k^{(\varepsilon)}}(\boldsymbol{x}) \le \mathrm{Loss}^{\mathfrak{G}_k}_{\mathfrak{A}_k}(\boldsymbol{x}) + |\boldsymbol{x}|\varepsilon \ .$$

Combining this with the first inequality in (13) completes the proof. □

5.3 Bounds for the Tuples of Complexities

The proofs in this subsection hold without changes for non-effective and effective complexities.

5.3.1 Tuples of Lower Complexities Belong to the Closure of the Entropy Hull

For every $k = 1, 2, \ldots, K$ let $\mathfrak{G}_k = \langle \Omega, \Gamma_k, \lambda_k \rangle$ and let H_k be the generalised entropy w.r.t. G_k; let $|\Omega| = M$ and $\Omega = \left\{\omega^{(0)}, \omega^{(1)}, \ldots, \omega^{(M-1)}\right\}$. Let AC_k be an asymptotic complexity w.r.t. $\mathfrak{G}_k$, $k = 1, 2, \ldots, K$. We will start show that the tuples of lower complexities belong to the $\le$-closure of the $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$-entropy hull.

The proof relies on Proposition 25.

We start from the case $K = 1$. The $\mathfrak{G}_1$-entropy hull coincides with its $\leq$-closure and equals the interval $[\min_{p\in\mathbb{P}_M} H_1(p), \max_{p\in\mathbb{P}_M} H_1(p)]$. Lemma 32 implies that for every $\varepsilon > 0$ there is a strategy $\mathfrak{S}_1$ such that for every finite sequence $\boldsymbol{x}$ we have

$$\mathrm{Loss}^{\mathfrak{G}_1}_{\mathfrak{S}_1}(\boldsymbol{x}) \leq |\boldsymbol{x}| \max_{p\in\mathbb{P}_M} H_1(p) + \varepsilon|\boldsymbol{x}| + o(|\boldsymbol{x}|)$$

and therefore $\mathrm{AC}_1(L) \leq \max_{p\in\mathbb{P}_M} H_1(p)$ for all languages L and complexities AC_1. On the other hand, by taking $p = e_i$, i.e., the vector of all zeroes with 1 at the i-th position, we get

$$H_1(e_i) = \min_{\gamma\in\Gamma_1} \lambda_1(\omega^{(i)}, \gamma)$$

and therefore $\min_{p\in\mathbb{P}_M} H_1(p) \leq \lambda_1(\omega, \gamma)$ for all $\omega \in \Omega, \gamma \in \Gamma_1$. Hence for every strategy $\mathfrak{S}$ and every finite sequence $\boldsymbol{x}$

$$\mathrm{Loss}^{\mathfrak{G}_1}_{\mathfrak{S}}(\boldsymbol{x}) \geq |\boldsymbol{x}| \min_{p\in\mathbb{P}_M} H_1(p)$$

and $\mathrm{AC}_1(L) \geq \min_{p\in\mathbb{P}_M} H_1(p)$ for all languages L and complexities AC_1.

Now consider the case $K = 2$.

Let $\mathcal{S}$ be the $\leq$-closure of the $\mathfrak{G}_1/\mathfrak{G}_2$-entropy hull $\mathcal{H}$ and L be a language of a type for which a lower complexity AC is defined. Our goal is to show that the point

$$(\mathrm{AC}_1(L), \mathrm{AC}_2(L)) = s = \left(s^{(1)}, s^{(2)}\right)$$

belongs to $\mathcal{S}$.

The projection $\mathcal{S}_1$ of $\mathcal{S}$ onto the first coordinate line is the $\leq$-closure of the $\mathfrak{G}_1$-entropy hull by Corollary 24. The case $K = 1$ considered above implies that $s_1 \in \mathcal{S}_1$.

Let $\varphi : \mathcal{S}_1 \to \mathbb{R}$ parametrise the upper part of the boundary of $\mathcal{S}$ as in (10). The sublattice $\mathcal{S}$ is an upper subsemilattice and Proposition 27 holds for $\mathcal{S}$, $\mathcal{S}_1$, and φ. We will show that $s^{(2)} \leq \varphi\left(s^{(1)}\right)$. If $s^{(1)} = \max(S_1)$, then it is sufficient to notice that $s^{(2)} \leq \max_{s\in S_1} \varphi(s) = \varphi(s_1)$. Let $s^{(1)} < \max(S_1)$.

Let $\mathrm{AC} = \underline{\mathrm{AC}}$ be uniform complexity. The equality $\underline{\mathrm{AC}}_1(L) = s^{(1)}$ implies that for every $\varepsilon > 0$ there a strategy $\mathfrak{S}_1$ w.r.t. $\mathfrak{G}_1$ such that for infinitely many positive integers n the inequality

$$\frac{\mathrm{Loss}^{\mathfrak{G}_1}_{\mathfrak{S}_1}(\boldsymbol{x})}{n} \leq s^{(1)} + \varepsilon$$

holds for all $\boldsymbol{x} \in L \cap \Omega^n$ if $L \subseteq \Omega^*$ or for all $\boldsymbol{x}$ that are prefixes of length n of elements of L if $L \subseteq \Omega^\infty$. If complexity is effective or polynomial, then $\mathfrak{S}_1$ can be chosen to be computable or polynomial-time computable, respectively.

Let $\mathrm{AC} = \underline{\underline{\mathrm{AC}}}$ be non-uniform complexity. Now $\underline{\underline{\mathrm{AC}}}_1(L) = s^{(1)}$ implies that for every $\boldsymbol{x} \in L$ there are infinitely many positive integers n such that the inequality

$$\frac{\mathrm{Loss}^{\mathfrak{G}_1}_{\mathfrak{S}_1}(\boldsymbol{x}|_n)}{n} \le s^{(1)} + \varepsilon$$

holds. Again, if complexity is effective or polynomial, then $\mathfrak{S}_1$ can be chosen to be computable or polynomial-time computable, respectively.

Each one of these cases can be reformulated as follows. For every $\varepsilon > 0$ there is a strategy $\mathfrak{S}_1$ such that for all $\boldsymbol{x}$ from a certain infinite set $L' \subseteq \Omega^*$ the inequality

$$\mathrm{Loss}^{\mathfrak{G}_1}_{\mathfrak{S}_1}(\boldsymbol{x}) \le \left(s^{(1)} + \varepsilon\right)|\boldsymbol{x}| \tag{14}$$

holds. The exact form of the set L' depends on the type of complexity AC.

By Lemma 32 there is a strategy $\mathfrak{S}_2$ (which can be chosen to be computable or polynomial-time computable if $\mathfrak{S}_1$ is) such that for every $\boldsymbol{x} \in L'$ there is

$$v = \left(v^{(1)}, v^{(2)}\right) \in \mathcal{H} \subseteq \mathcal{S}$$

such that

$$\mathrm{Loss}^{\mathfrak{G}_2}_{\mathfrak{S}_2}(\boldsymbol{x}) \le (v^{(2)} + \varepsilon)|\boldsymbol{x}| + o(|\boldsymbol{x}|)$$

and

$$v^{(1)}|\boldsymbol{x}| \le \mathrm{Loss}^{\mathfrak{G}_1}_{\mathfrak{S}_1}(\boldsymbol{x}) + \varepsilon|\boldsymbol{x}| \le \left(s^{(1)} + 2\varepsilon\right)|\boldsymbol{x}| \ .$$

We have $v^{(2)} \le \varphi\left(v^{(1)}\right)$. Since φ is non-decreasing, $\varphi\left(v^{(1)}\right) \le \varphi\left(s^{(1)} + 2\varepsilon\right)$ (provided $\varepsilon > 0$ is sufficiently small for $\varphi\left(s^{(1)} + 2\varepsilon\right)$ to be defined). Thus for every $\boldsymbol{x} \in L'$ the inequality

$$\mathrm{Loss}^{\mathfrak{G}_2}_{\mathfrak{S}_2}(\boldsymbol{x}) \le \left(\varphi\left(s^{(1)} + 2\varepsilon\right) + \varepsilon\right)|\boldsymbol{x}| + o(|\boldsymbol{x}|) \tag{15}$$

holds. By Proposition 27 if we take a sequence of positive numbers ε_n converging to 0, we get $\varphi\left(s^{(1)} + 2\varepsilon_n\right) \to \varphi\left(s^{(1)}\right)$ as $n \to \infty$. Thus

$$s^{(2)} = \mathrm{AC}(L) \le \varphi(s_1) \ . \tag{16}$$

A similar statement holds if we swap $\mathfrak{G}_1$ and $\mathfrak{G}_2$. Inequality (16) implies that there are $a_1, a_2 \ge 0$ such that $s + a_i e_i \in \mathcal{S}$, $i = 1, 2$. But $s = \min(s + a_1 e_1, s + a_2 e_2)$ and therefore $s \in \mathcal{S}$.

Now consider an arbitrary $K > 2$. Let $\mathcal{S}$ be the $\le$-closure of the $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$-entropy hull $\mathcal{H}$ and L be a language of a type for which a lower complexity AC is defined. The proof for the case of two games can be applied to any two games $\mathfrak{G}_i$ and $\mathfrak{G}_j$ with $1 \leq i < j \leq K$. We see that the tuple of complexities of L belongs to the cylinder over the $\leq$-closure of the projection of $\mathcal{S}$ onto the coordinate plane corresponding to coordinates i and j. By Proposition 25 the intersection of cylinders equals $\mathcal{S}$.

5.3.2 Tuples of Upper Complexities Belong to the Upper Closure of the Entropy Hull

In this subsection we show that the tuples of upper complexities belong to the upper $\leq$-closure of the $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$-entropy hull.

Let $\mathcal{U}$ be the upper $\leq$-closure of the $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$-entropy hull $\mathcal{H}$ and L be a language of a type for which an upper complexity AC is defined. We need to show that the point

$$(\mathrm{AC}_1(L), \mathrm{AC}_2(L), \ldots, \mathrm{AC}_K(L)) = s = \left(s^{(1)}, s^{(2)}, \ldots, s^{(K)}\right)$$

belongs to $\mathcal{U}$. The proof is by induction in K.

The base case of $K = 1$ is identical to Section 5.3.1.

Now we will assume the desired result holds for $K-1$ games and prove it for K games, $K \geq 2$.

Let us single out one coordinate of s, e.g., the last one and put $s = \left(s_1, s^{(K)}\right)$, where $s_1 \in \mathbb{R}^{K-1}$. By the induction hypothesis, s_1 belongs to the upper $\leq$-closure of the $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_{K-1}$-entropy hull. Corollary 24 implies that s_1 belongs to the projection $\mathcal{U}_1$ of $\mathcal{U}$ onto the first $K-1$ coordinates hyperplane.

Let $\varphi : \mathcal{U}_1 \to \mathbb{R}$ parametrises the upper part of the boundary of $\mathcal{U}$ as in (10). We will show that $s^{(K)} \leq \varphi(s_1)$; see Figure 10 for an illustration with $K = 3$ and $\mathcal{U}_1 \subseteq \mathbb{R}^2$.

If $\mathrm{AC} = \overline{\mathrm{AC}}$ is uniform complexity, then the equality $\overline{\mathrm{AC}}_k(L) = s^{(k)}$ implies that for every $\varepsilon > 0$ there a strategy $\mathfrak{A}_k$ w.r.t. $\mathfrak{G}_k$ such that for all sufficiently large integers n the inequality

$$\frac{\mathrm{Loss}_{\mathfrak{A}_k}^{\mathfrak{G}_k}(\boldsymbol{x})}{n} \leq s^{(k)} + \varepsilon$$

holds for all $\boldsymbol{x} \in L \cap \Omega^n$ if $L \subseteq \Omega^*$ or for all $\boldsymbol{x}$ that are prefixes of length n of elements of L if $L \subseteq \Omega^\infty$.

If $\mathrm{AC} = \overline{\overline{\mathrm{AC}}}$ is non-uniform complexity then $\overline{\overline{\mathrm{AC}}}_k(L) = s^{(k)}$ implies that for every $\boldsymbol{x} \in L$ there is a positive integer N such that for all $n \geq N$ the inequality

$$\frac{\mathrm{Loss}_{\mathfrak{A}_k}^{\mathfrak{G}_k}(\boldsymbol{x}|_n)}{n} \le s^{(k)} + \varepsilon$$

holds. In both the cases if complexity is effective or polynomial, then $\mathfrak{A}_k$ can be chosen to be computable or polynomial-time computable, respectively.

The sets of finite strings for which the respective inequalities hold can be intersected and the result is a set of the same type. Thus for every $\varepsilon > 0$ there are strategies $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_{K-1}$ such that for all $\boldsymbol{x}$ from a certain infinite set $L' \subseteq \Omega^*$ the inequalities

$$\mathrm{Loss}_{\mathfrak{A}_k}^{\mathfrak{G}_k}(\boldsymbol{x}) \le \left(s^{(k)} + \varepsilon\right)|\boldsymbol{x}| \tag{17}$$

hold for $k = 1, 2, \ldots, K-1$. The exact form of the set L' depends on the type of complexity AC.

By Lemma 32 there is a strategy $\mathfrak{A}_K$ (which can be chosen to be computable or polynomial-time computable if $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_{K-1}$ are) such that for every $\boldsymbol{x} \in L'$ there is

$$v = \left(v^{(1)}, v^{(2)}, \ldots, v^{(K)}\right) \in \mathcal{H} \subseteq \mathcal{U}$$

(again let $v = \left(v_1, v^{(K)}\right)$, where $v_1 \in \mathbb{R}^{K-1}$) such that

$$\mathrm{Loss}_{\mathfrak{A}_K}^{\mathfrak{G}_K}(\boldsymbol{x}) \le (v^{(K)} + \varepsilon)|\boldsymbol{x}| + o(|\boldsymbol{x}|)$$

and

$$v^{(k)}|\boldsymbol{x}| \le \mathrm{Loss}_{\mathfrak{A}_k}^{\mathfrak{G}_k}(\boldsymbol{x}) + \varepsilon|\boldsymbol{x}| \le \left(s^{(k)} + 2\varepsilon\right)|\boldsymbol{x}| \enspace ,$$

for $k = 1, 2, \ldots, K-1$. We get

$$\mathrm{Loss}_{\mathfrak{A}_K}^{\mathfrak{G}_K}(\boldsymbol{x}) \le (\varphi(v_1) + \varepsilon)|\boldsymbol{x}| + o(|\boldsymbol{x}|) \tag{18}$$

and

$$v_1 \le \left(s^{(1)} + 2\varepsilon, s^{(2)} + 2\varepsilon, \ldots, s^{(K-1)} + 2\varepsilon\right) \enspace . \tag{19}$$

Put

$$\mathcal{U}^{(2\varepsilon)} = \mathcal{U} \cap$$
$$\left(\left(-\infty, s^{(1)} + 2\varepsilon\right] \times \left(-\infty, s^{(2)} + 2\varepsilon\right] \times \ldots \times \left(-\infty, s^{(K-1)} + 2\varepsilon\right] \times \mathbb{R}\right) \enspace ;$$

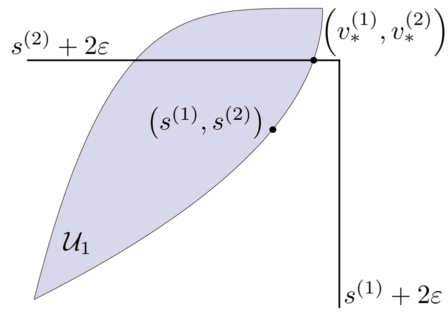

Figure 10: This illustrates the inductive step in the proof of Section 5.3.1 in $K - 1 = 2$ dimensions. For the picture the point $s_1 = \left(s^{(1)}, s^{(2)}\right)$ was chosen on the boundary of $\mathcal{U}_1$ and the overall configuration was chosen so that $v^* = \left(v_*^{(1)}, v_*^{(2)}\right)$ does not coincide with $\left(s^{(1)} + 2\varepsilon, s^{(2)} + 2\varepsilon\right)$; these arrangements demonstrate important special cases.

clearly, it is a compact convex sublattice. The set

$$\mathcal{U}_1^{(2\varepsilon)} = \mathcal{U}_1 \cap \left(\left(-\infty, s^{(1)} + 2\varepsilon\right] \times \left(-\infty, s^{(2)} + 2\varepsilon\right] \times \ldots \times \left(-\infty, s^{(K-1)} + 2\varepsilon\right]\right)$$

is its projection onto the first $K-1$ coordinates hyperplane. The bounds (18) and (19) imply that

$$\mathrm{AC}_K(L) \leq \max_{v_1 \in \mathcal{U}_1^{(2\varepsilon)}} \varphi(v_1) + \varepsilon \ .$$

By Proposition 27, φ achieves the maximum on $\mathcal{U}_1^{(2\varepsilon)}$ at

$$v^* = \left(v_*^{(1)}, v_*^{(2)}, \ldots, v_*^{(K-1)}\right) = \max \mathcal{U}_1^{(2\varepsilon)} \ .$$

The definition of $\mathcal{U}_1^{(2\varepsilon)}$ implies that $s^{(k)} \leq v_*^{(k)} \leq s^{(k)} + 2\varepsilon$ for every $k = 1, 2, \ldots, K-1$ and therefore v^* converges to s as $\varepsilon \to 0$. Proposition 27 implies that

$$s^{(K)} = \mathrm{AC}(L) \leq \varphi(s_1) \ . \tag{20}$$

We can apply this argument to every dimension and thus there are $a_k \geq 0$, $k = 1, 2, \ldots, K$, such that $u + a_k e_k \in \mathcal{U}$. We now need a lemma providing a kind of lower bound for u.

Lemma 34. *Let $\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_K$ be games with the same outcome space Ω. For every upper*[7] *asymptotic complexity* AC *and every suitable language L there is w from the $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$-entropy hull such that $w \leq (\mathrm{AC}_1(L), \mathrm{AC}_2(L), \ldots, \mathrm{AC}_K(L))$, where AC_k is upper complexity w.r.t. $\mathfrak{G}_k$, $k = 1, 2, \ldots, K$.*

Proof. For every $k = 1, 2, \ldots, K$ let $\mathfrak{G}_k = \langle \Omega, \Gamma_k, \lambda_k \rangle$ and let H_k be the generalised entropy w.r.t. $\mathfrak{G}_k$; let $|\Omega| = M$ and $\Omega = \left\{\omega^{(0)}, \omega^{(1)}, \ldots, \omega^{(M-1)}\right\}$.

For the entropy hull $\mathcal{H}$ define the set

$$\mathcal{S} = \{x \in \mathbb{R}^K \mid x \geq v \text{ for some } v \in \mathcal{H}\} \ .$$

Proving the lemma amounts to showing that

$$(\mathrm{AC}_1(L), \mathrm{AC}_2(L), \ldots, \mathrm{AC}_K(L)) \in \mathcal{S} \ .$$

Note that the set $\mathcal{S}$ is closed. Indeed, for every sequence $x_1, x_2, \ldots \in \mathcal{S}$ there is a sequence $v_1, v_2, \ldots \in \mathcal{H}$ such that $v_i \leq x_i$, $i = 1, 2, \ldots$. Since $\mathcal{H}$ is compact, the sequence of v_is has a converging subsequence and its limit ensures that the limit of x_i (if it exists) belongs to $\mathcal{S}$.

Let $\mathfrak{A}_k$ be a prediction strategy w.r.t. $\mathfrak{G}_k$, $k = 1, 2, \ldots, K$. By taking $p = e_i$, i.e., the vector of all zeroes except for 1 at the i-th position, we get

$$H_k(e_i) = \min_{\gamma \in \Gamma_k} \lambda_k(\omega^{(i)}, \gamma)$$

and therefore for every finite sequence x we get

$$\mathrm{Loss}_{\mathfrak{A}_k}^{\mathfrak{G}_k}(\boldsymbol{x}) \geq \sum_{i=0}^{M-1} \sharp_i \boldsymbol{x} \min_{\gamma \in \Gamma_k} \lambda_k(\omega^{(i)}, \gamma) = \sum_{i=0}^{M-1} \sharp_i \boldsymbol{x} H_k(e_i) \ .$$

Thus

$$\left(\frac{\mathrm{Loss}_{\mathfrak{A}_1}^{\mathfrak{G}_1}(\boldsymbol{x})}{|\boldsymbol{x}|}, \ldots, \frac{\mathrm{Loss}_{\mathfrak{A}_K}^{\mathfrak{G}_K}(\boldsymbol{x})}{|\boldsymbol{x}|} \right) \geq \sum_{i=0}^{M-1} \frac{\sharp_i \boldsymbol{x}}{|\boldsymbol{x}|} (H_1(e_i), \ldots, H_K(e_i)) \ ,$$

i.e., the point on the left-hand side belongs to $\mathcal{S}$.

Lemmas 29 and 30 imply that componentwise supremums of bounded sets of points in $\mathcal{S}$ and componentwise upper limits of bounded sequences of points from $\mathcal{S}$ belong to $\mathcal{S}$.

The tuples of upper complexities of a language may be thus approximated by points from $\mathcal{S}$ to any degree of precision and therefore belong to $\mathcal{S}$. □

[7] The previous version of this paper mistakenly claimed that this holds for all asymptotic complexities.

We will finally use the convexity of $\mathcal{U}$ to show that $u \in \mathcal{U}$.

Lemma 35. *Let $\mathcal{M} \subseteq \mathbb{R}^K$ be convex and $u \in \mathbb{R}^K$ be such that $u + a_k e_k \in \mathcal{M}$ for some $a_k \geq 0$ for all $k = 1, 2, \ldots, K$. If there is $w \in \mathcal{M}$ such that $w \leq u$, then $u \in \mathcal{M}$.*

Proof. By applying a shift we can ensure that $u = 0 = (0, 0, \ldots, 0)$. Then $w = \left(-w^{(1)}, -w^{(1)}, \ldots, -w^{(K)}\right)$, where $w^{(k)} \geq 0$, $k = 1, 2, \ldots, K$, and $u + a_k e_k = a_k e_k$.

If $a^{(k)} = 0$ for some k, there is nothing to prove, so suppose this is not the case. Let $q_k = w^{(k)}/a_k$, $k = 1, 2, \ldots, K$. We get $-w + \sum_{k=1}^{K} q_k a_k e_k = 0$. We can normalise the vector $(1, q_1, q_2, \ldots, q_K)$ so that its components sum up to 1. Thus u is a convex combination of points from $\mathcal{M}$. $\square$

Remark 36. The proofs in this section and Section 5.3.1 are quite similar. Note an important difference though. In Section 5.3.1 we link together pairs of complexities. Inequality (14) cannot be extended to more than one game, because the sets of strings $\boldsymbol{x}$ for different games do not necessarily have an intersection of the required type (of, for that matter, any intersection at all). This is due to the nature of lower limits. It is only Proposition 25 that allows us to make a step from cylinders over two-dimensional sets to the closure. By contrast, (17) holds for $K-1$ games because we deal with upper limits there.

5.4 Filling in the Closures of The Entropy Hull

In this subsection we finish the proof of the main theorem by showing that every point in a suitable closure of the entropy hull corresponds to a tuple of complexities.

5.4.1 Building Blocks

Let $|\Omega| = M$ and $\Omega = \left\{\omega^{(0)}, \omega^{(1)}, \ldots, \omega^{(M-1)}\right\}$ as usual. Take a distribution $p = \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)}\right)$ on Ω. We will now define a "basic building block" we will use to construct languages. Let $I^{(p)} \subseteq \{0, 1, \ldots, M-1\}$ be the set of indices of non-zero elements of p and $\Omega^{(p)} = \left\{\omega^{(i)} \in \Omega \mid i \in I^{(p)}\right\}$ be the support of the distribution p. Consider the set $\Xi_n^{(p)} \subseteq \left(\Omega^{(p)}\right)^n$ of sequences $\boldsymbol{x}$ of length n with the following property. For each $i \in I^{(p)}$, the number of $\omega^{(i)}$s among the elements of $\boldsymbol{x}$ is between the numbers $np_i - n^{3/4}$ and $np_i + n^{3/4}$, i.e., $np_i - n^{3/4} \leq \sharp_i \boldsymbol{x} \leq np_i + n^{3/4}$.

The following lemma summarises the properties of a building block.

Lemma 37. *Let $p = \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)}\right) \in \mathbb{P}_M$. For every game $\mathfrak{G}$ with the outcome space Ω of size M and the generalised entropy H there are constants $C = C(p, \mathfrak{G})$ and $D = D(p, \mathfrak{G})$ such that:*

1. *there is a prediction strategy $\mathfrak{A}$ such that*
$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}) \leq nH(p) + Cn^{3/4}$$
for all $\boldsymbol{x} \in \Xi_n^{(p)}$ and positive integer n; if the game $\mathfrak{G}$ is (polynomial-time) computable, then for every $\varepsilon > 0$ we can choose a (polynomial-time) computable $\mathfrak{A}$ achieving
$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}) \leq nH(p) + \varepsilon n + Cn^{3/4}$$
for all $\boldsymbol{x} \in \Xi_n^{(p)}$ and n;

2. *if*
$$n \geq N_0(p) = \max_{i \in I^{(p)} : p^{(i)} < 1} \left(\frac{1}{\left(p^{(i)}\right)^4}, \frac{1}{\left(1 - p^{(i)}\right)^4} \right) + 1 \tag{21}$$
then for every prediction strategy $\mathfrak{A}$ there is $\boldsymbol{x} \in \Xi_n^{(p)}$ such that
$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}) \geq nH(p) - Dne^{-2\sqrt{n}} \ .$$

In the definition of N_0 we assume $\max \varnothing = 0$ (this covers the case when the distribution is concentrated on one element of Ω).

Proof. Let $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$.

We start with a degenerate case when $p_j = 1$ for some j, i.e., $p = e_j$. Here $I^{(p)} = \{j\}$ and $\Xi_n^{(p)} = \left\{\omega^{(j)}\omega^{(j)} \ldots \omega^{(j)}\right\}$ consists of one finite string. On this string every strategy suffers loss greater than or equal to $nH(p)$ and a strategy that predicts $\gamma^* \in \arg\min_{\gamma \in \Gamma} \lambda\left(\omega^{(j)}, \gamma\right)$ suffers loss $nH(p)$. We can thus take $C = D = 0$.

For the rest of the proof assume that $p^{(i)} < 1$ for all $i = 0, 1, \ldots, M-1$. Take $\gamma^* \in \arg\min_{\gamma \in \Gamma} \sum_{i \in I^{(p)}} p^{(i)} \lambda\left(\omega^{(j)}, \gamma\right)$ and consider the strategy $\mathfrak{A}$ that always predicts γ^*. The number $C' = \max_{i \in I^{(p)}} \lambda\left(\omega^{(j)}, \gamma^*\right)$ is finite because $H(p)$ is finite. If $\boldsymbol{x} \in \Xi_n^{(p)}$ then by adding or removing no more than $n^{3/4}$ elements equal to $\omega^{(i)}$, $i \in I^{(p)}$, we can ensure that there are exactly $np^{(i)}$ elements $\omega^{(i)}$ in the string (imagine we can add or remove a fraction of the element and say that in some position the string has a fraction of $\omega^{(i)}$). After this operation the loss of $\mathfrak{A}$ will be $H(p)n$. The loss of $\mathfrak{A}$ on the original $\boldsymbol{x}$ differs from this value by no more than $C'Mn^{3/4}$. Put $C = C'M$.

If the game is (polynomial-time) computable, one can replace γ^* with a (polynomial-time) computable approximation $\bar{\gamma}^*$ suffering on every step extra loss not exceeding ε. This proves the first part of the lemma.

The proof of the second part of the lemma uses a probabilistic argument. Let $\xi_1^{(p)}, \xi_2^{(p)}, \ldots, \xi_n^{(p)}$ be independent random variables that accept the values $\omega^{(i)}$ with probabilities $p^{(i)}$, $i \in I^{(p)}$.

We need the Chernoff bound in Hoeffding's form (see Theorem 1 in [Hoe63]).

Proposition 38 (Chernoff bound)**.** *If $\xi_1, \xi_2, \ldots, \xi_n$ are independent random variables with finite first and second moments and such that $0 \le \xi_i \le 1$ for all $i = 1, 2, \ldots, n$ then*

$$\Pr\{\overline{\xi} - \mu \ge t\} \le e^{-2nt^2} \quad ,$$

for all $t \in (0, 1-\mu)$, where $\overline{\xi} = (\xi_1 + \xi_2 + \ldots + \xi_n)/n$ and $\mu = \mathbf{E}\overline{\xi}$.

It is easy to see that if $t \in (0, \min(\mu, 1-\mu))$ then the bound implies

$$\Pr\{|\overline{\xi} - \mu| \ge t\} \le 2e^{-2nt^2} \quad .$$

For every $\xi_j^{(p)}$ and $i \in I^{(p)}$ we can consider the random variable $\sharp_i \xi_j^{(p)}$ equal to 1 if $\xi_j^{(p)} = \omega^{(i)}$ and 0 otherwise. The expectation of this variable is $p^{(i)}$ and $\sharp_i \xi_1^{(p)}, \sharp_i \xi_2^{(p)}, \ldots, \sharp_i \xi_n^{(p)}$ satisfy the conditions of the Chernoff bound.

Take $t = n^{-1/4}$. If $n \ge N_0$ then $0 < t < \min_{i \in I^{(p)}} \left(p^{(i)}, 1 - p^{(i)}\right)$. For every $i \in I^{(p)}$ the Chernoff bound implies

$$\Pr\left\{\left|\sharp_i \left(\xi_1^{(p)} \xi_2^{(p)} \ldots, \xi_n^{(p)}\right) - p_i n\right| \ge n^{3/4}\right\} \le 2e^{-2\sqrt{n}} \quad . \tag{22}$$

For a set S we denote by $\Pr_p(S)$ the probability that the string $\xi_1^{(p)} \xi_2^{(p)} \ldots \xi_n^{(p)}$ belongs to S. We get $\Pr_p\left(\Omega^n \setminus \Xi_n^{(p)}\right) \le 2Me^{-2\sqrt{n}}$.

Consider a strategy $\mathfrak{A}$. First let us assume that λ is bounded and $D' = \max_{\omega \in \Omega, \gamma \in \Gamma} \lambda(\omega, \gamma)$. We get

$$\begin{aligned}
H(p)n &\le \mathbf{E}\,\mathrm{Loss}_{\mathfrak{A}}\left(\xi_1^{(p)} \xi_2^{(p)} \ldots, \xi_n^{(p)}\right) \\
&\le \Pr_p\left(\Xi_n^{(p)}\right) \max_{\boldsymbol{x} \in \Xi_n^{(p)}} \mathrm{Loss}_{\mathfrak{A}}(\boldsymbol{x}) + \Pr_p\left(\Omega^n \setminus \Xi_n^{(p)}\right) D'n \\
&\le \max_{\boldsymbol{x} \in \Xi_n^{(p)}} \mathrm{Loss}_{\mathfrak{A}}(\boldsymbol{x}) + \Pr_p\left(\Omega^n \setminus \Xi_n^{(p)}\right) D'n
\end{aligned}$$

(the first inequality follows from the definition of $H(p)$ and to get the second we can use upper bounds on the loss of $\mathfrak{A}$ on $\Xi_n^{(p)}$ and $\Omega^n \setminus \Xi_n^{(p)}$). Therefore there is a sequence $\boldsymbol{x} \in \Xi_n^{(p)}$ such that

$$\mathrm{Loss}_{\mathfrak{A}}(\boldsymbol{x}) \geq H(p)n - \Pr_p\left(\Omega^n \setminus \Xi_n^{(p)}\right) D'n \geq H(p)n - 2D'Mne^{-2\sqrt{n}}$$

provided $n \geq N_0$. We can take $D = 2D'M$.

Now let λ be unbounded. Take $\lambda^{(D)} = \min(\lambda, D)$, where $D > 0$ is a constant, and let $\mathfrak{G}^{(D)} = \langle \Omega, \Gamma, \lambda^{(D)} \rangle$. For every prediction strategy $\mathfrak{A}$ and every finite string $\boldsymbol{x} \in \Omega^*$ we have

$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}^{(D)}}(\boldsymbol{x}) \leq \mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x})$$

because the $\lambda^{(D)}$-loss is less than the λ-loss of the same prediction for all outcomes.

If $D' = 2H(p)/\min_{i \in I^{(p)}} p^{(i)}$ then

$$H(p) = \min_{\gamma \in \Gamma} \sum_{i=0}^{M-1} p^{(i)} \lambda\left(\omega^{(i)}, \gamma\right) = \min_{\gamma \in \Gamma} \sum_{i=0}^{M-1} p^{(i)} \lambda^{(D')}\left(\omega^{(i)}, \gamma\right) \quad .$$

Indeed, let the latter minimum be less than $H(p)$ and let it be achieved on $\gamma \in \Gamma$. For all $i \in I^{(p)}$ we get

$$p^{(i)} \lambda^{(D')}\left(\omega^{(i)}, \gamma\right) < H(p)$$

and thus

$$\lambda^{(D')}\left(\omega^{(i)}, \gamma\right) < H(p)/p^{(i)} \leq D' \quad .$$

Therefore $\lambda^{(D')}\left(\omega^{(i)}, \gamma\right) = \lambda\left(\omega^{(i)}, \gamma\right)$ for all $i \in I^{(p)}$ and the same γ ensures that the former minimum is also less than $H(p)$.

Repeating the above argument for the bounded game $\mathfrak{G}^{(D')}$ we conclude that for every strategy $\mathfrak{A}$ there is $\boldsymbol{x} \in \Xi_n^{(p)}$ such that

$$\begin{aligned} \mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}) &\geq \mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}^{(D')}}(\boldsymbol{x}) \\ &\geq H(p)n - 2D'Mne^{-2\sqrt{n}} \quad . \end{aligned}$$

Again we can take $D = 2D'M$. □

Remark 39. We do not explicitly mention the dependency on M in C and D; it comes in through p and $\mathfrak{G}$. We would like to warn the reader against assuming that C and D do not depend on M in a family of similar games parametrised by M. The values of C and D are however independent of n, the length of the block.

We will now combine basic building blocks into a more advanced block. Let $p_1, p_2, \ldots, p_N$ be distributions on Ω and let $q = (q_1, q_2, \ldots, q_N) \in \mathbb{P}_N$ (some q_i may be zero; we will keep those for padding). Take a positive integer n and put

$$\begin{aligned} n_0 &= 0 \ , \\ n_1 &= \lfloor q_1 n \rfloor \ , \\ n_2 &= \lfloor (q_1 + q_2) n \rfloor \ , \\ &\vdots \\ n_{N-1} &= \lfloor (q_1 + q_2 + \ldots + q_{N-1}) n \rfloor \ , \\ n_N &= n \ . \end{aligned}$$

If some $q_i = 0$, then $n_i = n_{i-1}$. Define

$$\Xi_n(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N) = \Xi^{(p_1)}_{n_1 - n_0} \times \Xi^{(p_2)}_{n_2 - n_1} \times \ldots \times \Xi^{(p_N)}_{n_N - n_{N-1}} \ .$$

If $q_i = 0$, then in this expression the ith term disappears.

The following counterpart of Lemma 37 holds.

Lemma 40. *Let $p_1, p_2, \ldots, p_N \in \mathbb{P}_M$ and $q = (q_1, q_2, \ldots, q_N) \in \mathbb{P}_N$. For every game $\mathfrak{G}$ with the outcome space Ω of size M and the generalised entropy H there are constants*

$$\begin{aligned} C &= C(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N; \mathfrak{G}) \ , \\ D &= D(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N; \mathfrak{G}) \end{aligned}$$

such that:

1. *there is a family of prediction strategies $\mathfrak{A}_n$, $n = 1, 2, \ldots$, such that*

$$\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}_n}(\boldsymbol{x}) \le n \sum_{i=1}^{N} q_i H(p_i) + C n^{3/4}$$

for all $\boldsymbol{x} \in \Xi_n(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N)$; if, moreover, the game $\mathfrak{G}$ is (polynomial-time) computable, then for every $\varepsilon > 0$ we can choose *a uniformly computable family $\mathfrak{A}_n$ (i.e., $\mathfrak{A}_n(\boldsymbol{y})$ can be computed from n and $\boldsymbol{y}$ (in time polynomial in n and $|\boldsymbol{y}|$)) such that*

$$\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}_n}(\boldsymbol{x}) \leq n \sum_{i=1}^{N} q_i H(p_i) + \varepsilon n + C n^{3/4}$$

for all $\boldsymbol{x} \in \Xi_n(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N)$;

2. *if*

$$n \geq N_0(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N) = \max_{i=1,2,\ldots,N, q_i>0} \frac{N_0(p_i)+1}{q_i} \ , \tag{23}$$

where $N_0(p)$ is as in (21), then for every prediction strategy $\mathfrak{A}$ there is $\boldsymbol{x} \in \Xi_n^{(p)}$ such that

$$\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}}(\boldsymbol{x}) \geq n \sum_{i=1}^{N} q_i H(p_i) - D \ .$$

Proof. Combining the strategies from the first part of Lemma 37 we can construct a strategy $\mathfrak{A}$ that on elements from $n_{i-1}+1$ to n_i (provided $q_i > 0$) suffers loss less than or equal to

$$(n_i - n_{i-1}) H(p_i) + C(p_i, \mathfrak{G})(n_i - n_{i-1})^{3/4} \ ,$$

where $n_0 = 0$ and $C(p_i, \mathfrak{G})$ is the constant guaranteed by Lemma 37. Let us use bounds $n_i - n_{i-1} \leq q_i n + 1$ (rounding-up can give the interval no more than one "extra" point on the left and can only "remove" points on the right) and $(q_i n + 1)^{3/4} \leq (n+1)^{3/4} \leq 2^{3/4} n^{3/4} \leq 2n^{3/4}$. Taking $C' = \max_{i=1,2,\ldots,N,q_i>0} C(p_i, \mathfrak{G})$ leads to the bound

$$\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}}(\boldsymbol{x}) \leq n \sum_{i=1}^{N} q_i H(p_i) + \sum_{i=1}^{N} H(p_i) + 2C' N n^{3/4} \ .$$

for all $\boldsymbol{x} \in \Xi_n(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N)$. We can take $C = \sum_{i=1}^{N} H(p_i) + 2C'N$.

Suppose that the game $\mathfrak{G}$ is (polynomial-time) computable. Take $\varepsilon > 0$. Let $r_1, r_2, \ldots, r_{N-1}$ be dyadic approximations of $q_1, q_1 + q_2, \ldots, q_1 + q_2 + \ldots + q_{N-1}$ such that $\left| r_i - \sum_{j=1}^{i} q_i \right| < \varepsilon$, $i = 1, 2, \ldots, N-1$. Let $\bar{\gamma}_i^*$ be a (polynomial-time) computable approximation to γ_i^* defined by p_i as in the proof of Lemma 37 so that $\bar{\gamma}_i^*$ suffers on every step extra loss not exceeding ε, $i = 1, 2, \ldots, N-1$.

We will now describe a uniform family of strategies $\mathfrak{A}_n$. Given n and $\boldsymbol{y}$ we calculate $\lfloor r_i n \rfloor$, $i = 1, 2, \ldots, N-1$ with precision 0.5 (in time polynomial in n). Taking the floor will give us an approximation to n_i accurate to within $\varepsilon n + 2$. Then we compare the length $|\boldsymbol{y}|$ against the approximations of n_i and depending on the result output the suitable γ_i^*. If

$$E = \max_{i=1,2,\ldots,N;\ \ j \in I^{(p_i)}} \lambda(\omega^{(j)}, \gamma_i^*) \enspace ,$$

then

$$\mathrm{Loss}_{\mathfrak{A}_n}^{\mathfrak{G}}(\boldsymbol{x}) \le n \sum_{i=1}^{N} q_i H(p_i) + \sum_{i=1}^{N} H(p_i) + 2C' N n^{3/4} + \varepsilon n + (E+\varepsilon)(\varepsilon n + 2)(N-1)$$

for all $\boldsymbol{x} \in \Xi_n(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N)$. We can adjust ε and C to get the desired inequality.

In order to prove the second part of the lemma first note that $n_i - n_{i-1} \ge q_i n - 1$ and $n \ge N_0$ guarantees that $n_i - n_{i-1} \ge N_0(p_i)$ provided $q_i > 0$. Since $N_0(p_i) \ge 1$, we also have $q_i n \ge 2$. For an arbitrary strategy $\mathfrak{A}$ we can use the second part of Lemma 37 to construct a "hard to predict" string for the interval from $n_{i-1}+1$ to n_i where $\mathfrak{A}$ suffers loss greater than or equal to

$$(n_i - n_{i-1}) H(p_i) - D(p_i, \mathfrak{G})(n_i - n_{i-1}) e^{-2\sqrt{n_i - n_{i-1}}} \enspace .$$

Take $D' = \max_{i=1,2,\ldots,N, q_i > 0} D(p_i, \mathfrak{G})$ and $q = \min_{i=1,2,\ldots,N, q_i > 0} q_i$. For the exponent we can use the bound

$$e^{-2\sqrt{n_i - n_{i-1}}} \le e^{-2\sqrt{qn - 1}} \le e^{-2(\sqrt{qn} - 1)} = e^2 e^{-2\sqrt{qn}} \enspace .$$

We can now put together those hard strings and obtain a string $\boldsymbol{x} \in \Xi_n(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N)$ such that

$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(x) \ge n \sum_{i=1}^{N} q_i H(p_i) - \sum_{i=1}^{N} H(p_i) - D' e^2 n e^{-2\sqrt{qn}} \enspace .$$

The last term tends to zero as $n \to \infty$ and we can take

$$D = \max\left(\sum_{i=1}^{N} H(p_i), \max_{n = N_0, N_0+1, \ldots} D' e^2 n e^{-2\sqrt{qn}} \right) \enspace .$$

$\square$

5.4.2 Regular Languages

Take a game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$. We call a language $L \subseteq \Omega^\infty$ *regular* if the quadrangle (8) collapses and all four complexities coincide: $\overline{\mathrm{AC}}(L) = \underline{\mathrm{AC}}(L) = \overline{\overline{\mathrm{AC}}}(L) = \underline{\underline{\mathrm{AC}}}(L)$.

Similarly, we will call a language $L \subseteq \Omega^\infty$ *effective regular* if all its effective complexities coincide and *polynomial-time regular* if all its polynomial-time complexities coincide.

Lemma 41. *A nonempty language $L \subseteq \Omega^\infty$ is regular w.r.t. a game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ and*

$$\overline{\mathrm{AC}}(L) = \underline{\mathrm{AC}}(L) = \overline{\overline{\mathrm{AC}}}(L) = \underline{\underline{\mathrm{AC}}}(L) = c$$

if and only if

1. *for every $\varepsilon > 0$ there is a prediction strategy $\mathfrak{A}$ w.r.t. $\mathfrak{G}$ such that*

$$\max_{\boldsymbol{x} \in L} \mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}|_n) \leq (c + \varepsilon)n + o(n)$$

as $n \to \infty$;

2. *for every prediction strategy $\mathfrak{A}$ w.r.t. $\mathfrak{G}$ there is $\boldsymbol{x} \in L$ such that*

$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}|_n) \geq cn - o(n)$$

as $n \to \infty$.

The same criteria hold for effective and polynomial-time regularity provided we restrict ourselves to computable or polynomial-time computable strategies.

The lemma immediately follows from the definitions of complexities.

We will now use Lemma 40 to construct a regular language of complexity $\sum_{i=1}^N q_i H(p_i)$. Take a positive integer

$$T_0 \geq N_0(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N)$$

(see (23) for the definition) and put

$$\begin{aligned} &L(T_0; p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N) = \\ &\qquad \prod_{k=1}^{\infty} \Xi_{kT_0}(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N) = \\ &\Xi_{T_0}(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N) \times \Xi_{2T_0}(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N) \times \\ &\qquad \Xi_{3T_0}(p_1, p_2, \ldots, p_N; q_1, q_2, \ldots, q_N) \times \ldots \in \Omega^\infty \ . \end{aligned}$$

Proposition 42. *Let* $p_1, p_2, \dots, p_N \in \mathbb{P}_M$ *and* $q = (q_1, q_2, \dots, q_N) \in \mathbb{P}_N$. *For every* $T_0 \geq N_0(p_1, p_2, \dots, p_N; q_1, q_2, \dots, q_N)$ *and for every game* $\mathfrak{G}$ *with the outcome space* Ω *of size* M *and the generalised entropy* H *the language* $L(T_0; p_1, p_2, \dots, p_N; q_1, q_2, \dots, q_N)$ *is regular of complexity* $\sum_{i=1}^N q_i H(p_i)$.

If the game $\mathfrak{G}$ *is computable, then the language is effective regular of complexity* $\sum_{i=1}^N q_i H(p_i)$. *If the game* $\mathfrak{G}$ *is polynomial-time computable, then the language is polynomial-time regular of complexity* $\sum_{i=1}^N q_i H(p_i)$.

Proof. The proof is by combining Lemma 41 with Lemma 40.

Let $L = L(T_0; p_1, p_2, \dots, p_N; q_1, q_2, \dots, q_N)$. Fix a game $\mathfrak{G}$ with the outcome space Ω. Let C and D be the constants provided by Lemma 40.

For every positive integer n let $k(n)$ be the number of the block Ξ where n belongs, i.e.,

$$k(n) = \max_{\sum_{i=1}^{k-1} iT_0 + 1 \leq n} k \ .$$

Respectively, let

$$l(n) = \sum_{k=1}^{k(n)-1} kT_0 + 1 = T_0 \frac{k(n)(k(n)-1)}{2} + 1$$

$$u(n) = \sum_{k=1}^{k(n)} kT_0 = T_0 \frac{k(n)(k(n)+1)}{2}$$

be the numbers of the first and the last elements of the block. It is easy to see that $k(n) \to \infty$ and

$$l(n) \sim n \sim u(n) \sim T_0 \frac{(k(n))^2}{2} \tag{24}$$

as $n \to \infty$.

In order to prove the first condition from Lemma 41 we will use the first part of Lemma 40. Let the strategy $\mathfrak{A}$ use the strategies provided by Lemma 40 within the respective blocks. For every $\boldsymbol{x} \in L$ we have

$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}|_n) \leq u(n) \sum_{i=1}^N q_i H(p_i) + C \sum_{k=1}^{k(n)} (kT_0)^{3/4} \ . \tag{25}$$

For the second term on the right-hand side we have

$$C \sum_{k=1}^{k(n)} (kT_0)^{3/4} \sim \frac{4}{7} C T_0^{3/4} (k(n))^{7/4} = o(n)$$

as $n \to \infty$. The first condition of Lemma 41 follows.

If the game is (polynomial-time) computable, then for every $\varepsilon > 0$ we can achieve (25) with an extra term εn on the right-hand side by a (polynomial-time) computable strategy $\mathfrak{A}$.

In order to prove the second condition from Lemma 41 we will use the second part of Lemma 40. For every strategy $\mathfrak{A}$ there is a "hard" element of Ξ_{kT_0}, $k = 1, 2, \ldots$. By combining them we get $\boldsymbol{x} \in L$ such that

$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}|_n) \geq l(n) \sum_{i=1}^{N} q_i H(p_i) - D(k(n) - 1) \ .$$

Clearly, for the last term on the right-hand side we have

$$D(k(n) - 1) = o(n)$$

as $n \to \infty$. The second condition of Lemma 41 follows. □

Proposition 43. *Let $L_1, L_2, \ldots, L_k \subseteq \Omega^\infty$ be regular languages of complexities $c_1, c_2, \ldots, c_K$ w.r.t. a game $\mathfrak{G}$ with the outcome space Ω. If $\mathfrak{G}$ is weakly mixable, then the union $L = L_1 \cup L_2 \cup \ldots L_K$ is a regular language of complexity $c = \max(c_1, c_2, \ldots, c_K)$. If $\mathfrak{G}$ is computably very weakly mixable, the same holds for effective regular languages; if $\mathfrak{G}$ is polynomial-time computably very weakly mixable, the same holds for polynomial-time regular languages.*

Proof. The proof is by Lemma 41. For every $\varepsilon > 0$ and $i = 1, 2, \ldots, K$ there is a ((polynomial-time) computable) strategy $\mathfrak{A}_i$ such that

$$\max_{\boldsymbol{x} \in L_i} \mathrm{Loss}_{\mathfrak{A}_i}^{\mathfrak{G}}(\boldsymbol{x}|_n) \leq n(c_i + \varepsilon) + o(n)$$

as $n \to \infty$. Weak mixability of $\mathfrak{G}$ implies that there is a strategy $\mathfrak{A}$ merging $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_K$ and achieving

$$\max_{\boldsymbol{x} \in L} \mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}|_n) \leq n(c + \varepsilon) + o(n) \ .$$

If $\mathfrak{G}$ is (polynomial-time) computably very weakly mixable, the strategy $\mathfrak{A}$ can be chosen to be (polynomial-time) computable at the cost of an extra εn on the right-hand side.

This gives us the first condition of Lemma 41.

Let the maximum of c_i be achieved on some j, i.e., $c = c_j$. For every ((polynomial-time) computable) strategy $\mathfrak{A}$ w.r.t. $\mathfrak{G}$ there is $\boldsymbol{x} \in L_j \subseteq L$ such that

$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}|_n) \geq nc - o(n)$$

as $n \to \infty$. This gives us the second condition of Lemma 41 and completes the proof. □

Corollary 44. *Let $\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_K$ be weakly mixable games with the outcome space Ω. Then for every point $u = \left(u^{(1)}, u^{(2)}, \ldots, u^{(K)}\right)$ from the upper $\leq$-closure of the $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$-entropy hull there is a language $L \subseteq \Omega^\infty$ such that L is regular of complexity $u^{(k)}$ w.r.t. $\mathfrak{G}_k$, $k = 1, 2, \ldots, K$. If the games are computable and computably very weakly mixable, the language can be taken to be effective regular; if the games are polynomial-time computable and polynomial-time computably very weakly mixable, the language can be taken to be polynomial-time regular.*

We have constructed the languages filling in the upper $\leq$-closure of the entropy hull.

5.4.3 Semiregular Languages

It remains to fill in the lower $\leq$-closure. We cannot use regular languages for obvious reasons and we need to relax the definition.

For a game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ we call a language $L \subseteq \Omega^\infty$ *lower semiregular* if its lower complexities coincide, $\underline{\mathrm{AC}}(L) = \underline{\underline{\underline{\mathrm{AC}}}}(L)$. Similarly, we call L *effective lower semiregular* if its lower effective complexities coincide and *lower polynomial-time semiregular* if its lower polynomial-time complexities coincide.

Lemma 45. *A nonempty language $L \subseteq \Omega^\infty$ is lower semiregular w.r.t. a game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ and*

$$\underline{\mathrm{AC}}(L) = \underline{\underline{\underline{\mathrm{AC}}}}(L) = c$$

if and only if

1. *for every $\varepsilon > 0$ there is a prediction strategy $\mathfrak{A}$ w.r.t. $\mathfrak{G}$ and a sequence $n_1 < n_2 < \ldots$ such that*

$$\max_{\boldsymbol{x} \in L} \mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}}(\boldsymbol{x}|_{n_k}) \leq (c + \varepsilon) n_k + o(n_k)$$

   as $n \to \infty$;

2. *for every prediction strategy $\mathfrak{A}$ w.r.t. $\mathfrak{G}$ there is $\boldsymbol{x} \in L$ such that*

$$\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}}(\boldsymbol{x}|_n) \geq cn - o(n)$$

   as $n \to \infty$.

The same criteria hold for effective and polynomial-time regularity provided we restrict ourselves to computable or polynomial-time computable strategies.

In order to fill in the $\leq$-closure, we will use the representation from Corollary 18.

Let $j = 1, 2, \ldots, J$, $r = 1, 2, \ldots, R$, $s = 1, 2, \ldots, S$. Suppose that we have JRS distributions $p_{j,r,s} \in \mathbb{P}_M$, and JR distributions

$$q^{(j,r)} = \left(q_1^{(j,r)}, q_2^{(j,r)}, \ldots, q_S^{(j,r)}\right) \in \mathbb{P}_S \ .$$

We will construct a lower semiregular language of complexity

$$c = \min_j \max_r \sum_s q_s^{(j,r)} H(p_{j,r,s}) \ .$$

Let

$$T_0 = \max_{j,r} N_0 \left(p_{j,r,1}, p_{j,r,2}, \ldots, p_{j,r,S}; q_1^{(j,r)}, q_2^{(j,r)}, \ldots, q_S^{(j,r)}\right) \ ;$$

put

$$L_{j,r} = L\left(T_0; p_{j,r,1}, p_{j,r,2}, \ldots, p_{j,r,S}; q_1^{(j,r)}, q_2^{(j,r)}, \ldots, q_S^{(j,r)}\right) \ .$$

By Proposition 42 for every $\mathfrak{G}$ with asymptotic complexity H this is a regular language of complexity $\sum_s q_s^{(j,r)} H(p_{j,r,s})$. By Proposition 43, if the game $\mathfrak{G}$ is weakly mixable, then $L_j = \bigcup_r L_{j,r}$ is a regular language of complexity $c_j = \max_r \sum_s q_s^{(j,r)} H(p_{j,r,s})$.

We will now define a language $L \subseteq \Omega^\infty$ by combining the languages L_j. We will "paint" the sequence $1, 2, 3, \ldots$ using colours $1, 2, \ldots, J$. The set L consists of all sequences $\boldsymbol{x}$ with the following property. For every $j = 1, 2, \ldots, J$, if we remove from $\boldsymbol{x}$ all elements standing in positions not painted in j and close the gaps by shifting what remains toward the beginning, we get a sequence from L_j.

The languages L_j consist of blocks of size kT_0, $k = 1, 2, \ldots$. The colouring will be done in such a way that each block will remain contiguous. The language L will not necessarily be a Cartesian product of blocks because each L_j is not necessarily a Cartesian product. However each L_j is a union of Cartesian products and L will also be a union of Cartesian products.

Now let us describe the colouring. Take a sequence k_n, $n = 1, 2, \ldots$, such that $k_{n-1} = o(k_n)$, e.g., $k_n = n!$. We paint as much of the beginning of the sequence $1, 2, \ldots$ using colour 1 so as to fit k_1 initial blocks from L_1. Then we paint as much of what remains using colour 2 so as to fit $k_2 - k_1$ initial blocks from L_2; then we paint as much of what remains using colour 3 so as to fit $k_3 - k_2$ initial blocks from L_3 etc. After painting in J enough to fit $k_J - k_{J-1}$ initial blocks from L_J, we again use colour 1 and paint enough of what remains to fit $k_{J+1} - k_J$ blocks from L_1 following those we already fit. Then the construction process repeats itself.

Proposition 46. *For all JRS distributions $p_{j,r,s} \in \mathbb{P}_M$ and JR distributions*

$$q^{(j,r)} = \left(q_1^{(j,r)}, q_2^{(j,r)}, \ldots, q_S^{(j,r)}\right) \in \mathbb{P}_S$$

if $\mathfrak{G}$ is a weakly mixable game with the outcome space Ω of size M, then the language L constructed above is lower semiregular of complexity

$$c = \min_j \max_r \sum_s q_s^{(j,r)} H(p_{j,r,s}) \enspace .$$

If $\mathfrak{G}$ is computable and computably very weakly mixable, then L is effective lower semiregular of complexity c; if $\mathfrak{G}$ is polynomial-time computable and polynomial-time computably very weakly mixable, then L is polynomial-time lower semiregular of complexity c.

Proof. The proof is by Lemma 45.

Let $c_j = \max_r \sum_s q_s^{(j,r)} H(p_{j,r,s})$. Fix j to a value such that $c_j = c = \min_{i=1,2,\ldots,J} c_i$. For the language L_j we have a strategy $\mathfrak{A}$ such that

$$\max_{\boldsymbol{x} \in L_j} \mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}|_n) \le (c + \varepsilon)n + o(n) \tag{26}$$

as $n \to \infty$. We can turn the strategy $\mathfrak{A}$ into a strategy for predicting sequences from L. Let $\mathfrak{S}$ follow $\mathfrak{A}$ on positions from blocks from L_j and output some minimax prediction γ^* such that $\lambda(\omega, \gamma^*) \le A < \infty$ for all $\omega \in \Omega$ otherwise.

Let n be such that at the nth step in the construction of L we took $k_n - k_{n-1}$ blocks from L_j. Let m be the number of the position in L where the last of these blocks finishes. We will obtain an upper bound on $\max_{\boldsymbol{x} \in L} \mathrm{Loss}_{\mathfrak{S}}^{\mathfrak{G}}(\boldsymbol{x}|_m)$.

Let $\boldsymbol{x} \in L$. The first m positions in $\boldsymbol{x}$ include at least $k_n - k_{n-1}$ initial blocks from L_j. Their total length does not exceed m, i.e.,

$$m \ge \sum_{k=1}^{k_n - k_{n-1}} kT_0 = T_0 \frac{(k_n - k_{n-1})(k_n - k_{n-1} + 1)}{2} \enspace . \tag{27}$$

We can use (26) to upper bound the loss of $\mathfrak{S}$ on elements from these blocks.

The number m has been chosen to ensure that elements of $\boldsymbol{x}$ painted differently make little contribution to the loss. Of every other language L_i, $i \neq j$, the interval from 1 to m includes no more than k_{n-1} blocks. In total their length does not exceed $(J-1)T_0 k_{n-1}(k_{n-1}+1)/2$ and the loss of $\mathfrak{S}$ on each element of each of those blocks does not exceed A. Combining $k_{n-1} = o(k_n)$ with (27), we get

$$\frac{1}{m}(J-1)T_0\frac{k_{n-1}(k_{n-1}+1)}{2} \leq (J-1)\frac{k_{n-1}(k_{n-1}+1)}{(k_n-k_{n-1})(k_n-k_{n-1}+1)} = o(1)$$

and

$$\begin{aligned}\max_{\boldsymbol{x}\in L}\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{S}}(\boldsymbol{x}|_m) &\leq (c+\varepsilon)m + o(m) + A(J-1)T_0\frac{k_{n-1}(k_{n-1}+1)}{2}\\ &\leq (c+\varepsilon)m + o(m)\end{aligned}$$

as $n \to \infty$.

This holds for infinitely many m from the sequence $k_1, k_2, \ldots$ because we took blocks from L_j infinitely many times in the construction process.

If $\mathfrak{G}$ has efficiency properties, the strategy $\mathfrak{A}$ can be chosen to be efficient and the resulting $\mathfrak{S}$ is efficient. In order to compute $\mathfrak{S}$ we need to be able to figure out whether the current position n is in the area painted in colour j. This can be done in time polynomial in j. Indeed, $n!$ can calculated in time polynomial in n by Part 2 of Corollary 51.

Let us turn to the second condition of Lemma 45. Take a positive integer n. Position n belongs to a block from some language L_j. Let $k(n)$ be the number of this block in L_j and $l(n)$ and $u(n)$ be the numbers of the first and the last elements in the block (in L). We get

$$\frac{u(n)}{l(n)} = \frac{l(n)+T_0k(n)}{l(n)} = 1 + \frac{T_0k(n)}{l(n)} \to 1$$

as $n \to \infty$ because $l(n) \geq T_0k(n)(k(n)-1)/2+1$. Therefore we get an equivalence similar to (24):

$$l(n) \sim n \sim u(n)$$

as $n \to \infty$.

Take a prediction strategy $\mathfrak{A}$. We need to find a “hard” sequence in L. By construction for every L_j there is a regular language $L_{j,r_j} \subseteq L_j$ that has the same complexity as L_j and that is an infinite Cartesian product of Ξs. We can construct the hard sequence by induction concatenating hard strings existing by Lemma 40 in blocks of L_{j,r_j} (note that we cannot choose hard string independently: we do not know how $\mathfrak{A}$ will behave on a block until we have constructed all preceding elements).

Let D be the maximum of D's defined by Lemma 40 (we get one for each distribution q^{j,r_j}). We get

$$\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}}(\boldsymbol{x}|_n) \geq cl(n) - Db(n) \enspace ,$$

where $b(n)$ is the number of blocks from all languages in the interval from 1 to n.

If $k_0 T_0$ is the length of a longest block and it belongs to a language L_i, then this block is preceded by blocks of length $T_0, 2T_0, \ldots, (k_0-1)T_0$ from L_i and their total length is $T_0 k_0(k_0+1)/2$. It does not exceed $l(n)$ and therefore

$$\frac{T_0 k_0^2}{2} \leq T_0 \frac{k_0(k_0+1)}{2} \leq l(n) \enspace .$$

From each language L_i there are no more than k_0 blocks in the interval, i.e.,

$$b(n) \leq k_0 J \leq J \sqrt{\frac{2l(n)}{T_0}} = o(n) \tag{28}$$

as $n \to \infty$.

The lemma follows. $\square$

Corollary 18 implies the following.

Corollary 47. *Let $\mathfrak{G}_1, \mathfrak{G}_2, \ldots, \mathfrak{G}_K$ be weakly mixable games with the same outcome space Ω. Then for every point $s = \left(s^{(1)}, s^{(2)}, \ldots, s^{(K)}\right)$ from the $\leq$-closure of the $\mathfrak{G}_1/\mathfrak{G}_2/\ldots/\mathfrak{G}_K$-entropy hull there is a language $L \subseteq \Omega^\infty$ such that L is lower semiregular of lower complexity $s^{(k)}$ w.r.t. $\mathfrak{G}_k$, $k = 1, 2, \ldots, K$. If the games are computable and computably very weakly mixable, the language can be taken to be effective lower semiregular; if the games are polynomial-time computable and polynomial-time computably very weakly mixable, the language can be taken to be polynomial-time lower semiregular.*

6 Predictability and Dimension

In this section we discuss an application of the main theorem.

We reproduce the definitions of predictability and dimension from [FL05] and show how they can be reinterpreted in terms of asymptotic complexities. Then we apply the main theorem to describe the set of pairs of predictabilities and dimensions for all non-empty languages $L \subseteq \Omega^\infty$.

We will be using our notation rather than that from [FL05].

6.1 Non-effective Case

Consider the outcome space $\Omega = \{\omega^{(0)}, \omega^{(1)}, \ldots, \omega^{(M-1)}\}$ of size M and the prediction space

$$\mathbb{P}_M = \left\{ \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)} \right) \in [0,1]^M \mid \sum_{i=0}^{M-1} p^{(i)} = 1 \right\}$$

of all distributions on Ω.

The *success rate* of a strategy $\mathfrak{A} : \Omega^* \to \mathbb{P}_M$ is defined as follows. Let $\mathfrak{A}_i(\boldsymbol{x})$ be the i-th component ($i = 0, 1, \ldots, M-1$) of the prediction output by $\mathfrak{A}$ on a finite sequence $\boldsymbol{x} \in \Omega^*$ of previous outcomes. Then for every finite sequence $\boldsymbol{y} = \omega^{(i_1)}, \omega^{(i_2)}, \ldots, \omega^{(i_n)}$ the success rate is given by

$$\mathfrak{A}^+(\boldsymbol{y}) = \frac{1}{n} \sum_{j=1}^{n} \mathfrak{A}_{i_j} \left(\omega^{(i_1)}, \omega^{(i_2)}, \ldots, \omega^{(i_{j-1})} \right) \ .$$

In other terms, the success of a prediction $\left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)} \right)$ given an outcome $\omega^{(i)}$ is $p^{(i)}$ and the success rate of a prediction strategy on a finite sequence is the cumulative success per element.

If $\boldsymbol{x} \in \Omega^\infty$ is an infinite sequence, then the success rate is defined as $\mathfrak{A}^+(\boldsymbol{x}) = \limsup_{n\to\infty} \mathfrak{A}^+(\boldsymbol{x}|_n)$. The worst-case success rate on a language $L \subseteq \Omega^\infty$ is $\mathfrak{A}^+(L) = \inf_{\boldsymbol{x}\in L} \mathfrak{A}^+(\boldsymbol{x})$. The *predictability* of a language is the supremum of success rates over strategies, i.e.,

$$\operatorname{pred}(L) = \sup_{\mathfrak{A}} \mathfrak{A}^+(L) = \sup_{\mathfrak{A}} \inf_{\boldsymbol{x}\in L} \limsup_{n\to\infty} \mathfrak{A}^+(\boldsymbol{x}|_n) \ . \tag{29}$$

We will now reinterpret the notion of predictability in terms of games and losses. Let the *multidimensional absolute-loss game* be $\mathfrak{G}_{\text{abs}} = \langle \Omega, \mathbb{P}_M, \lambda_{\text{abs}} \rangle$, where

$$\lambda_{\text{abs}} \left(\omega^{(i)}, \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)} \right) \right) = 1 - p^{(i)} \ ,$$

$i = 0, 1, \ldots, M-1$. For every finite sequence $\boldsymbol{x} \in \Omega^*$ the success rate and the loss of a strategy are related through the equality $\mathfrak{A}^+(\boldsymbol{x}) = 1 - \frac{1}{n} \operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{x})$ and therefore for every language $L \subseteq \Omega^\infty$ we get $\operatorname{pred}(L) = 1 - \underline{\underline{\mathrm{AC}}}_{\text{abs}}(L)$, where $\underline{\underline{\mathrm{AC}}}_{\text{abs}}(L)$ is lower non-uniform complexity w.r.t. the game.

Let us move on to the concept of dimension. An *s-gale* is a function $d : \Omega^* \to [0, +\infty)$ such that for every finite sequence $\boldsymbol{y} \in \Omega^*$ the equality $d(\boldsymbol{y}) = M^{-s} \sum_{\omega\in\Omega} d(\boldsymbol{y}\omega)$ holds. An s-gale *succeeds* on $\boldsymbol{x} \in \Omega^\infty$ if $\limsup_{n\to\infty} d(\boldsymbol{x}|_n) = +\infty$. The *dimension* of a non-empty language $L \subseteq \Omega^\infty$ is given by $\dim(L) = \inf \mathcal{G}(L)$, where

$$\mathcal{G}(L) = \{ s \mid \text{there is an } s\text{-gale that succeeds on all } \boldsymbol{x} \in L \} \ .$$

The definition of $\mathcal{G}$ can be restated as follows. A 0-gale is a function $d_0 : \Omega^* \to [0, +\infty]$ such that for every $\boldsymbol{y} \in \Omega^*$ we have $d_0(\boldsymbol{y}) = \sum_{\omega \in \Omega} d_0(\boldsymbol{y}\omega)$. Some d is an s-gale if and only if $d(\boldsymbol{y}) = M^{s|\boldsymbol{y}|} d_0(\boldsymbol{y})$ for all $\boldsymbol{y} \in \Omega^*$, where d_0 is a 0-gale. We will say that a 0-gale d_0 s-succeeds on $\boldsymbol{x} \in \Omega^\infty$ if $\limsup_{n\to\infty} d_0(\boldsymbol{x}|_n) M^{sn} = +\infty$. Clearly, $d(\boldsymbol{x}|_n) = M^{sn} d_0(\boldsymbol{x}|_n)$ succeeds on $\boldsymbol{x} \in \Omega^\infty$ if and only if d_0 s-succeeds on x. Thus

$$\mathcal{G}(L) = \{s \mid \text{there is an 0-gale that } s\text{-succeeds on all } \boldsymbol{x} \in L\} \ .$$

Note that a 0-gale $d_0(\boldsymbol{x}) = 1/M^{|\boldsymbol{x}|}$ $(1+\varepsilon)$-succeeds on every $\boldsymbol{x} \in \Omega^\infty$ for every $\varepsilon > 0$. On the other hand, for every 0-gale d_0 and $\boldsymbol{x} \in \Omega^\infty$ the sequence $d_0(x|_n)$, $n = 1, 2, \ldots$ is nondecreasing and thus can be bounded from above by a constant. Therefore d_0 cannot s-succeed on $\boldsymbol{x}$ for $s \leq 0$. Hence for all $L \subseteq \Omega^\infty$ we get $0 < \dim(L) \leq 1$.

Let us give a game interpretation[8]. The *multidimensional logarithmic game* is $\mathfrak{G}_{\log} = \langle \Omega, \mathbb{P}_M, \lambda_{\log} \rangle$, where $\lambda_{\log}\left(\omega^{(i)}, \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)}\right)\right) = -\log_M p^{(i)}$. We will now relate 0-gales to losses of strategies and dimension to asymptotic complexity. For every strategy $\mathfrak{A}$ the function $d_0(\boldsymbol{y}) = M^{-\operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{y})}$ is a 0-gale. Indeed, let

$$\mathfrak{A}(\boldsymbol{y}) = \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)}\right) \ .$$

for some finite string $\boldsymbol{y} \in \Omega^*$. Then

$$\sum_{\omega \in \Omega} d_0(\boldsymbol{y}\omega) = M^{-\operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{y})} \sum_{i=0}^{M-1} p^{(i)} = M^{-\operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{y})} = d_0(\boldsymbol{y}) \ .$$

Let $\liminf_{n\to\infty} \operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{x}|_n)/n = s$ for some infinite string $\boldsymbol{x} \in \Omega^\infty$. Then for every $\varepsilon > 0$ there are infinitely many $n \in \mathbb{N}$ such that $\operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{x}|_n)/n \leq s + \varepsilon$, i.e., $M^{-\operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{x}|_n)} \geq M^{-(s+\varepsilon)n}$ and $M^{-\operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{x}|_n)} M^{(s+2\varepsilon)n} \geq M^{\varepsilon n} \to +\infty$ as $n \to \infty$. Thus d_0 $(s+2\varepsilon)$-succeeds on $\boldsymbol{x}$ and for every language $L \subseteq \Omega^\infty$ we have $\dim(L) \leq \underline{\underline{\mathrm{AC}}}_{\log}(L)$, where $\underline{\underline{\mathrm{AC}}}_{\log}$ is lower non-uniform complexity w.r.t. the multidimensional logarithmic game.

Now let d_0 be a 0-gale. The function $\tilde{d}_0 : \Omega^* \to [0, +\infty)$ defined by

$$\tilde{d}_0(\boldsymbol{y}) = d_0(\boldsymbol{y}) + \frac{1}{M^{|\boldsymbol{y}|}} \tag{30}$$

is a 0-gale. For every $s \leq 1$ and infinite sequence $\boldsymbol{x} \in \Omega^\infty$ it s-succeeds on $\boldsymbol{x}$ if and only if d_0 succeeds on $\boldsymbol{x}$. (Recall that for every language $L \subseteq \Omega^\infty$ we have $\dim(L) \leq 1$.)

[8]As pointed out by a reviewer, this argument essentially reproduces Theorem 3.1 from [Hit03].

For every finite string $\boldsymbol{y} \in \Omega^*$ the inequality $\tilde{d}_0(\boldsymbol{x}) > 0$ holds and we can define numbers $p^{(i)} = \tilde{d}_0\left(\boldsymbol{y}\omega^{(i)}\right)/\tilde{d}_0(\boldsymbol{y})$, $i = 0, 1, \ldots, M-1$, such that $p^{(i)} \in [0,1]$ and $\sum_{i=0}^{M-1} p^{(i)} = 1$. Let $\mathfrak{A}$ be the strategy outputting $p = \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)}\right)$ on $\boldsymbol{y} \in \Omega^\infty$. We get $\tilde{d}_0(\boldsymbol{y}) = M^{-\operatorname{Loss}_{\mathfrak{A}}(|\boldsymbol{y}|)}$ and $\operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{y}) = -\log_M \tilde{d}_0(\boldsymbol{y})$.

If $\tilde{d}_0$ s-succeeds on an infinite string $\boldsymbol{x} \in \Omega^\infty$, then there is a sequence of positive integers $n_1 < n_2 < \ldots$ such that $\tilde{d}_0(\boldsymbol{x}|_{n_k})M^{sn_k} \to +\infty$ as $k \to \infty$. Taking the logarithm to the base M yields $\operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{x}|_{n_k}) - sn_k \to -\infty$ and from some $k = k_0$ on $\operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{x}|_{n_k}) - sn_k \leq 0$, i.e., $\operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{x}|_{n_k})/n_k \leq s$. Thus $\liminf_{n\to\infty} \operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{x}|_n) \leq s$ and for every language $L \subseteq \Omega^\infty$ we have $\dim(L) \geq \underline{\underline{\mathrm{AC}}}_{\log}(L)$. Hence $\dim(L) = \underline{\underline{\mathrm{AC}}}_{\log}(L)$.

6.2 Effective Versions

In this subsection we consider the effective and polynomial-time counterparts of predictability and dimension.

The effective and polynomial-time predictability predE and predP are defined by (29), where the supremums are restricted to computable and polynomial-time computable strategies, respectively. It is easy to see that $\mathrm{predE} = 1 - \underline{\underline{\mathrm{ACE}}}_{\mathrm{abs}}$ and $\mathrm{predP} = 1 - \underline{\underline{\mathrm{ACP}}}_{\mathrm{abs}}$, where $\underline{\underline{\mathrm{ACE}}}_{\mathrm{abs}}$ and $\underline{\underline{\mathrm{ACP}}}_{\mathrm{abs}}$ are effective and polynomial-time non-uniform lower complexities w.r.t. the multidimensional absolute-loss game.

The effective and polynomial-time dimensions dimE and dimP are defined in the same way as dim, except that computable and polynomial-time computable gales are considered.

Let us establish the equality of these functions and corresponding complexities. The 0-gale $d_0(\boldsymbol{y}) = M^{-\operatorname{Loss}_{\mathfrak{A}}(\boldsymbol{y})}$ equals the product of relevant components of predictions output by the strategy $\mathfrak{A}$ on substrings of $\boldsymbol{y}$. Clearly, if $\mathfrak{A}$ is computable, then d_0 is computable. By Part 2 of Corollary 51 if $\mathfrak{A}$ is polynomial-time computable, then d_0 is polynomial-time computable.

If a 0-gale d_0 is (polynomial-time) computable, then the 0-gale $\tilde{d}_0$ defined by (30) is also (polynomial-time) computable. Indeed, the function $\boldsymbol{y} \to 1/M^{|\boldsymbol{y}|}$ is polynomial-time computable by Part 2 of Corollary 51 and the sum is trivially computable and polynomial-time computable by Part 1 of Corollary 51. The ratio $\boldsymbol{y} \to \tilde{d}_0(\boldsymbol{y}\omega)/\tilde{d}_0(\boldsymbol{y})$ is computable for every $\omega \in \Omega$ because $\tilde{d}_0(\boldsymbol{y}) > 0$ and polynomial-time computable by Part 4 of Corollary 51 because $\tilde{d}_0(\boldsymbol{y}) \geq 1/M^{|\boldsymbol{y}|}$. Thus if d_0 is (polynomial-time) computable then the strategy $\mathfrak{A}$ predicting $p = \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)}\right)$, where $p^{(i)} = \tilde{d}_0\left(\boldsymbol{y}\omega^{(i)}\right)/\tilde{d}_0(\boldsymbol{y})$, $i = 0, 1, \ldots, M-1$, on input $\boldsymbol{y}$ is (polynomial-time) computable.

Repeating the argument from the previous subsection, we conclude that $\mathrm{dimE} = \underline{\underline{\mathrm{ACE}}}_{\log}$ and $\mathrm{dimP} = \underline{\underline{\mathrm{ACP}}}_{\log}$, where $\underline{\underline{\mathrm{ACE}}}_{\log}$ and $\underline{\underline{\mathrm{ACP}}}_{\log}$ are effective and polynomial-time non-uniform lower complexities w.r.t. the multidimensional logarithmic game.

Remark 48. If follows from Proposition 63 and Lemmas 55 and 60 that the non-effective, effective, and polynomial-time versions of complexities $\mathrm{AC}_{\mathrm{abs}}$ and $\mathrm{AC}_{\log}$ do not coincide. Therefore non-effective, effective, and polynomial-time versions of predictability and dimension differ. The values $\dim(L)$, $\mathrm{dimE}(L)$, and $\mathrm{dimP}(L)$ do not necessarily coincide and the values $\mathrm{pred}(L)$, $\mathrm{predE}(L)$, and $\mathrm{predP}(L)$ do not necessarily coincide.

6.3 Computability and Mixability of the Games

In order to apply Theorem 6 we need to check that its requirements hold. The multidimensional absolute-loss game is polynomial-time computable. Indeed, polynomial-time computable numbers are dense in $\mathbb{P}_M$ and the function $e^{-\lambda_{\mathrm{abs}}\left(p,\omega^{(i)}\right)} = e^{-\left(1-p^{(i)}\right)}$, where $p = \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)}\right)$, is polynomial-time computable.

In order to establish the polynomial-time computability of the multidimensional logarithmic game, we need to show that the function

$$e^{-\lambda_{\log}\left(p,\omega^{(i)}\right)} = e^{\log_M p^{(i)}} = \left(p^{(i)}\right)^{1/\ln M}$$

is polynomial-time computable.

Lemma 49. *For every polynomial-time computable $\alpha > 0$ the function $x \to x^{\alpha}$ is polynomial-time computable for $x \in [0,1]$.*

The proof of the lemma is given in Appendix A.2.

It follows from the lemma that the multidimensional logarithmic game is polynomial-time computable.

Note that the simplex $\mathbb{P}_M$ is convex and both λ_{abs} and $\lambda_{\log}$ are convex in the second argument. Therefore the multidimensional absolute-loss and logarithmic games are mixable. Proposition 3 implies that both the games are computably very weakly mixable and polynomial-time computably very weakly mixable.

6.4 Applying the Main Theorem

Theorem 6 can be used to establish relations between dim and pred and between their effective counterparts.

We need to calculate the generalised entropies for the games. Let $p = \left(p^{(0)}, p^{(1)}, \ldots, p^{(M-1)}\right) \in \mathbb{P}_M$. For the multidimensional absolute-loss game we have

$$\begin{aligned} H_{\text{abs}}(p) &= \min_{\gamma=\left(\gamma^{(0)},\gamma^{(1)},\ldots,\gamma^{(M-1)}\right)\in\mathbb{P}_M} \sum_{i=0}^{M-1} p^{(i)}\left(1-\gamma^{(i)}\right) \\ &= 1 - \max_{\gamma=\left(\gamma^{(0)},\gamma^{(1)},\ldots,\gamma^{(M-1)}\right)\in\mathbb{P}_M} \sum_{i=0}^{M-1} p^{(i)}\gamma^{(i)} \quad . \end{aligned}$$

The maximum of a linear function must be achieved at one of the vertices of the simplex, i.e., points of the form $(0,\ldots,0,1,0,\ldots,0)$ with exactly one 1. Hence

$$H_{\text{abs}}(p) = 1 - \max_{i=0,1,\ldots,M-1} p^{(i)} \quad .$$

For the multidimensional logarithmic game we have

$$H_{\log}(p) = \min_{\gamma=\left(\gamma^{(0)},\gamma^{(1)},\ldots,\gamma^{(M-1)}\right)\in\mathbb{P}_M} \left(-\sum_{i=0}^{M-1} p^{(i)} \log_M \gamma^{(i)}\right) \quad .$$

The expression that should be minimised is convex in γ. At the point $\gamma = p$ its gradient is orthogonal to the hyperplane of the simplex and therefore $\gamma = p$ is the minimum. We get

$$H_{\log}(p) = -\sum_{i=0}^{M-1} p^{(i)} \log_M p^{(i)} \quad .$$

By considering the Hessian of $H_{\log}(p)$ one can check that $H_{\log}(p)$ is concave on $\mathbb{P}_M$.

Let us construct the $\mathfrak{G}_{\text{abs}}/\mathfrak{G}_{\log}$-entropy set and hull. The entropy $H_{\text{abs}}(p)$ ranges from 0 to $1-1/M$ since the minimal value of $\max_{i=0,1,\ldots,M-1} p^{(i)}$ is $1/M$. For $a \in [0, 1-1/M]$ take

$$\begin{aligned} l(a) &= \min_{p\in\mathbb{P}_M : H_{\text{abs}}(p)=a} H_{\log}(p) \\ u(a) &= \max_{p\in\mathbb{P}_M : H_{\text{abs}}(p)=a} H_{\log}(p) \quad . \end{aligned}$$

The $\mathfrak{G}_{\text{abs}}/\mathfrak{G}_{\log}$-entropy set is bounded by the graphs of $l(a)$ and $u(a)$.

Let us take $a \in [0, 1-1/M]$ and calculate $u(a)$ and $l(a)$. We can write

$$\begin{aligned} B_a &= \{p \in \mathbb{P}_M | H_{\text{abs}}(p) = a\} \\ &= \left\{ \left(p^{(0)}, p^{(1)}, \dots, p^{(M-1)}\right) \in \mathbb{P}_M | \max_{i=0,1,\dots,M-1} p^{(i)} = 1-a \right\} \\ &= \bigcup_{i=0}^{M-1} B_{i,a} \ , \end{aligned}$$

where

$$B_{i,a} = \left\{ \left(p^{(0)}, p^{(1)}, \dots, p^{(M-1)}\right) \in \mathbb{P}_M | \right. \\ \left. p^{(i)} = 1-a \text{ and } p^{(j)} \leq 1-a, j = 1, 2, \dots, M-1 \right\} \ .$$

By symmetry the maximum of $H_{\log}(p)$ on B_a equals the maximum on one of these sets. The set

$$B_{M-1,a} = \left\{ \left(p^{(0)}, p^{(1)}, \dots, p^{(M-2)}, 1-a\right) | \right. \\ \left. p^{(i)} \in [0, 1-a], i = 0, 1, \dots, M-2, \sum_{i=0}^{M-2} p^{(i)} = a \right\} \ .$$

can be thought as a part of the "scaled down" simplex in $\mathbb{R}^{M-1}$. The function $H_{\log}$ is concave on $B_{M-1,a}$ and by the gradient argument its maximum is achieved at $(a/(M-1), \dots, a/(M-1), 1-a)$. Thus

$$u(a) = -(1-a)\log_M(1-a) - a\log_M \frac{a}{M-1} \ . \tag{31}$$

In order to evaluate $l(a)$ consider the set $P_a \supseteq B_a$ defined as

$$\begin{aligned} P_a &= \left\{ \left(p^{(0)}, p^{(1)}, \dots, p^{(M-1)}\right) \in \mathbb{P}_M | p^{(i)} \leq 1-a, i = 0, 1, \dots, M-1 \right\} \\ &= \left\{ \left(p^{(0)}, p^{(1)}, \dots, p^{(M-1)}\right) | \right. \\ &\qquad \left. p^{(i)} \in [0, 1-a], i = 0, 1, \dots, M-1, \sum_{i=0}^{M-1} p^{(i)} = 1 \right\} \ . \end{aligned}$$

This is a convex polygon; it is the intersection of the hyperplane $\sum_{i=0}^{M-1} p^{(i)} = 1$ with subspaces $p^{(i)} \geq 0$ and $p^{(i)} \leq 1-a$, $i = 0, 1, \dots, M-1$. Since $H_{\log}$ is concave, it achieves the minimum at a vertex of the polygon. Let us work out the coordinates of a vertex. It is the intersection of M hyperplanes specified by equations from the list $p^{(i)} = 0$, $p^{(i)} = 1 - a$ $(i = 0, 1, \ldots, M-1)$, and $\sum_{i=0}^{M-1} p^{(i)} = 1$. One hyperplane must be $\sum_{i=0}^{M-1} p^{(i)} = 1$ as P_a is its subset. The other $M-1$ hyperplanes are of the type $p^{(i)} = 0$ or $p^{(i)} = 1 - a$ and each of them ensures that one of the coordinates of the vertex is 0 or $1-a$. The coordinates or a vertex thus consist of $M-1$ zeroes or values $(1-a)$ and a "remainder" $r \in [0, 1-a]$. This is only possible if there are $\lfloor 1/(1-a) \rfloor$ values $(1-a)$ and $r = 1 - (1-a)\lfloor 1/(1-a) \rfloor$. All these vertices belong to the original B_a. Subject to rearranging the coordinates there is only one vertex and the value of $H_{\log}$ there provides the minimum:

$$l(a) = -\lfloor 1/(1-a) \rfloor (1-a) \log_M (1-a) - r \log_M r \ ,$$

where $r = 1 - (1-a)\lfloor 1/(1-a) \rfloor$. We get $r = 0$ if and only if $1/(1-a)$ is an integer, i.e., $a = 1 - 1/k$, $k = 1, 2, \ldots, M$. At those points we get

$$l(1 - 1/k) = \log_M k \ .$$

Let us construct the $\mathfrak{G}_{\mathrm{abs}}/\mathfrak{G}_{\log}$-entropy hull. The function $u(a)$ is concave on $[0, 1 - 1/M]$ and therefore it represents the upper bound of the hull. The function $l(a)$ is concave on each $[1 - 1/k, 1 - 1/(k+1)]$, $k = 1, 2, \ldots, M-1$. Therefore the lower bound of the hull is represented by the function $h(a)$ that coincides with $l(a)$ at the points $a = 1 - 1/k$ and equals a convex combination between two adjacent points:

$$h(a) = \begin{cases} \log_M k, & \text{if } a = 1 - 1/k \ , \\ k(k+1)\Big[\big(a - \big(1 - \frac{1}{k}\big)\big) \log_M (k+1) + & \\ \quad \Big(\Big(1 - \frac{1}{k+1}\Big) - a\Big) \log_M k\Big], & \text{if } a \in \Big(1 - \frac{1}{k}, 1 - \frac{1}{k+1}\Big) \end{cases} , \tag{32}$$

where $k = 1, 2, \ldots, M$.

We obtain the following corollary from the main theorem.

Corollary 50. *For every $\Omega = \{1, 2, \ldots, M-1\}$ the sets*

$$\{(\dim(L), \mathrm{pred}(L)) \mid L \subseteq \Omega^\infty\}$$
$$\{(\mathrm{dimE}(L), \mathrm{predE}(L)) \mid L \subseteq \Omega^\infty\}$$
$$\{(\mathrm{dimP}(L), \mathrm{predP}(L)) \mid L \subseteq \Omega^\infty\}$$

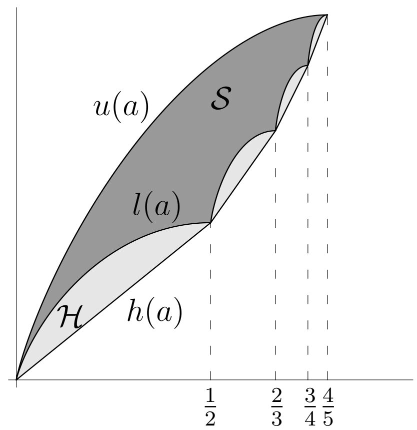

Figure 11: The graph of the $\mathfrak{G}_{\text{abs}}/\mathfrak{G}_{\text{log}}$-entropy hull $\mathcal{H}$; the $\mathfrak{G}_{\text{abs}}/\mathfrak{G}_{\text{log}}$-entropy set inside it is in darker grey.

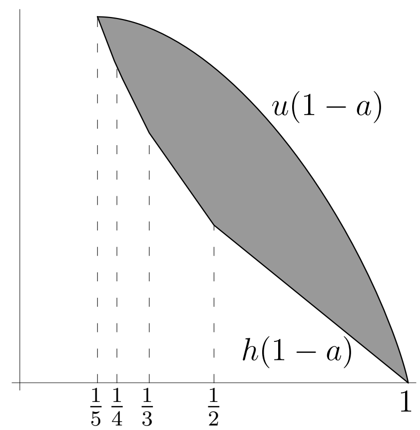

Figure 12: The set of pairs $(\dim(L), \operatorname{pred}(L))$.

coincide with the set

$$\left\{(x,y) \mid x \in \left[\frac{1}{M}, 1\right], h(1-x) \le y \le u(1-x)\right\} \quad ,$$

where u is given by (31) and h is given by (32).

It follows from Remark 48 that the three statements of the corollary about non-effective, effective, and polynomial-time predictability and dimension do not trivially follow from each other.

Figures 11 and 12 provide an illustration for $M = 5$.

Appendix A. Some Properties of Computable Functions

In this appendix we discuss some important properties of computable and polynomial-time computable functions.

A.1 Computability and Operations on Computable Functions

Let us state several properties of computable and polynomial-time computable functions. It is easy to see that addition and multiplication are computable functions on $\mathbb{R} \times \mathbb{R}$, taking the exponent e^x is a computable function on $\mathbb{R}$, the inversion $1/x$ is computable on $\mathbb{R} \setminus \{0\}$ and the logarithm $\ln x$ is computable on $(0, +\infty)$. The situation with polynomial-time computability is a bit more difficult. Corollaries 7.3.2 and 7.3.10 and Theorems 7.3.12 and 7.3.18 from [Wei00] (together with Theorem 9.4.3 establishing the equivalence of the approaches from [Wei00] and [Ko91]) state that the above functions are polynomial-time computable on compact subsets of their domains. Therefore we can apply the above operations to polynomial-time computable functions $f : \Omega^* \to \mathbb{R}$ provided their values belong to respective compact subsets of $\mathbb{R}$ and get polynomial-time computable functions.

Unfortunately for the purposes of this paper we cannot restrict ourselves to functions from Ω^* to compact subsets of $\mathbb{R}^k$, but we can prove the following corollary.

Corollary 51. *1. If functions $f_1, f_2, \ldots, f_k : \Omega^* \to \mathbb{R}$ are polynomial-time computable, then their sum $f_1 + f_2 + \ldots + f_k$ and product $f_1 \cdot f_2 \cdot \ldots \cdot f_k$ are polynomial-time computable.*

2. If a function $f : \Omega^* \to \mathbb{R}$ is polynomial-time computable, then the cumulative product

$$h(\omega_1\omega_2\ldots\omega_n) = f(\omega_1)f(\omega_1\omega_2)\cdots f(\omega_1\omega_2\ldots\omega_n)$$

is polynomial-time computable.

3. If a function $f : \Omega^* \to \mathbb{R}$ is polynomial-time computable and there are positive integers C and r such that for all $\boldsymbol{x} \in \Omega^*$ we have $|f(\boldsymbol{x})| \leq C(|\boldsymbol{x}|^r + 1)$, then e^f is polynomial-time computable.

4. If a function $f : \Omega^* \to \mathbb{R}$ is polynomial-time computable and there are positive integers C and r such that for all $\boldsymbol{x} \in \Omega^*$ we have $f(x) \geq 2^{-C(|\boldsymbol{x}|^k+1)}$, then the functions $1/f$ and $\ln f$ are polynomial-time computable.

We need a simple lemma.

Lemma 52. *If $f : \Omega^* \to \mathbb{R}$ is polynomial-time computable and $g : \Omega^* \to \mathbb{Z}$ is an integer-valued polynomial-time computable function not exceeding in the absolute value a polynomial in the length of its argument, then $f(\boldsymbol{x})2^{g(\boldsymbol{x})}$ is also polynomial-time computable.*

Proof of the lemma. Given a representation of a dyadic number $d = sm/2^n$ one can easily multiply or divide d by 2. This can be done in time linear in the length of the representation.

If d is an approximation of $y2^k$, where $y \in \mathbb{R}$ and k is an integer, accurate to within 2^{-n}, i.e., $|y2^k - d| < 2^{-n}$, then $|y - d2^{-k}| < 2^{-n-k}$ and $d2^{-k}$ is an approximation to y (though with a different precision). Calculating $d2^{-k}$ given a representation for d requires k multiplications or divisions by 2 and can be done in time polynomial in k and the length of the representation of d. □

We will refer to the multiplications or divisions by 2 necessary to calculate g from f as an adjustment of the precision.

Proof of the corollary. Part 1. If $f : \Omega^* \to \mathbb{R}$ is polynomial-time computable then in time polynomial in the length $|\boldsymbol{x}|$ we can produce an approximation to $f(\boldsymbol{x})$ with precision 1. The length of this approximation must be polynomial in $|\boldsymbol{x}|$ and therefore the absolute value $|f(\boldsymbol{x})|$ cannot exceed $2^{C(|\boldsymbol{x}|^r+1)}$ for some positive integers C and r.

Let $C_1, C_2, \ldots, C_k$ and $r_1, r_2, \ldots, r_k$ be such constants for $f_1, f_2, \ldots, f_k$. Take $C = \max_{i=1,2,\ldots,k} C_i$ and $r = \max_{i=1,2,\ldots,k} r_i$. For the functions $g_i(\boldsymbol{x}) = f_i(\boldsymbol{x})/2^{C(|\boldsymbol{x}|^r+1)}$ we have $-1 \le g_i(\boldsymbol{x}) \le 1$ and their sums and products can be calculated in time polynomial in precision and length of $\boldsymbol{x}$. We then adjust the precision.

Part 2. As before, there are positive integers C and r such that $|f(\boldsymbol{x})| \le 2^{C(|\boldsymbol{x}|^r+1)}$. Take $\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x})/2^{C(|\boldsymbol{x}|^r+1)}$ and let $\tilde{h}$ be the cumulative product of $\tilde{f}$s. The functions $\tilde{h}(\boldsymbol{x})$ and $h(\boldsymbol{x})$ differ by the factor $2^{q(\boldsymbol{x})}$, where $q(\boldsymbol{x})$ is polynomial-time computable and does not exceed $C(|\boldsymbol{x}|^r + 1)|\boldsymbol{x}|$. Thus in order to calculate h it is sufficient to calculate $\tilde{h}$ and adjust the precision.

Let us calculate the precision required to calculate the product. Suppose that $x_1, x_2, \ldots, x_n \in [-1, 1]$ but instead of x_i we know an approximation $x_i + \delta_i$ so that $|\delta_i| \le \delta$, $i = 1, 2, \ldots, n$. We get

$$|(x_1 + \delta_1)(x_2 + \delta_2) \cdots (x_n + \delta_n) - x_1 x_2 \ldots x_n| \le \\ \delta n + \delta^2 \binom{n}{2} + \ldots + \delta^n \binom{n}{n} = (1 + \delta)^n - 1 \le e^{\delta n} - 1 \le 2\delta n \quad (33)$$

provided $\delta n \le 1$ (we used the inequalities $1 + x \le e^x$ and $e^x \le 1 + 2x$ for $x \in [0, 1]$).

Suppose that we are required to calculate $\tilde{h}(\boldsymbol{x})$ with a precision m. It is sufficient to know the values of $\tilde{f}$ accurate to within $2^{-m}/(2|\boldsymbol{x}|)$. One can find the minimal integer k such that $2^k \ge |\boldsymbol{x}|$ (in time polynomial in $|\boldsymbol{x}|$) and calculate dyadic approximations to $|\boldsymbol{x}|$ values of $\tilde{f}$ accurate to within 2^{-m-1-k} (in time polynomial in $|\boldsymbol{x}|$ and m). Then the values can be multiplied (ignoring the insignificant digits) in time polynomial in $|\boldsymbol{x}|$ and m.

Part 3. Given $|\boldsymbol{x}|$ and integers $C, r > 0$, one can calculate a positive integer p such that $C(|\boldsymbol{x}|^r + 1) \leq 2^p < 2C(|\boldsymbol{x}|^r + 1)$.

Let $g(\boldsymbol{x}) = f(\boldsymbol{x})/2^p$ and consider the formula

$$e^{f(\boldsymbol{x})} = \left(\frac{e^{g(\boldsymbol{x})}}{4}\right)^{2^p} 2^{2 \cdot 2^p} \quad .$$

Since $-1 \leq g(\boldsymbol{x}) \leq 1$, the exponent of g can be calculated in polynomial time and it does not exceed $e < 4$. We can then calculate the product of $2^p < 2(C|\boldsymbol{x}|^r + 1)$ values not exceeding 1. Adjusting the precision completes this part of the proof.

Part 4. Take $p = C(|\boldsymbol{x}|^r + 1)$. Consider the function $g(\boldsymbol{x}) = 2^p f(\boldsymbol{x}) \geq 1$. It can be computed in polynomial time and therefore does not exceed $2^{C_2(|\boldsymbol{x}|^{r_2}+1)}$ for some C_2 and r_2 (chosen independently of $\boldsymbol{x}$). Calculating $g(\boldsymbol{x})$ to the precision 1 one can find an integer $p_2 \geq 0$ such that $2^{p_2} \leq g(x) \leq 2^{p_2+2}$. Therefore for $h(\boldsymbol{x}) = g(\boldsymbol{x})/2^{p_2+1}$ we have $0.5 \leq h(\boldsymbol{x}) \leq 2$. The set $[0.5, 2]$ is compact and separated from 0, so one can calculate $1/(h(\boldsymbol{x}))$ in polynomial time and then apply the adjustment of the precision using

$$\frac{1}{f(\boldsymbol{x})} = 2^{p-p_2-1} \frac{1}{h(\boldsymbol{x})} \quad .$$

One can also calculate $\ln h$ in polynomial time and use the formula

$$\ln f(\boldsymbol{x}) = \ln h(\boldsymbol{x}) + (p_2 - p + 1) \ln 2 \quad .$$

The value of $\ln 2$ can be calculated in polynomial time. Then it can be multiplied by ps and added to $\ln h(\boldsymbol{x})$ by the parts of the corollary proved earlier. □

Remark 53. Note that the definition of polynomial-time computability is somewhat asymmetric.

The function $f : \mathbb{N} \to \mathbb{R}$ given by $f(n) = 2^{2^n}$ cannot be computed in time polynomial in n. Indeed, if d approximates $f(n)$ accurate to within 1, then a binary expansion of d cannot be shorter than 2^n and therefore it cannot be printed in time polynomial in n.

On the other hand, the function $g(n) = 1/f(n) = 2^{-2^n}$ is polynomial-time computable. Suppose that we are required to calculate $g(n)$ accurate to within 2^{-m}. We start by checking whether the condition $2^n \geq m$ holds, which can be done in time polynomial in m and n. If this condition holds then $g(n) = 2^{-2^n} \leq 2^{-m}$ and we can output 0 as an approximation to $g(n)$.

If the condition does not hold and $2^n \leq m$, then a binary representation of $g(n)$ can be printed in time polynomial in m.

Thus the inverse of a polynomial-time computable non-vanishing function from Ω^* to $\mathbb{R}$ is not necessarily polynomial-time computable.

Similarly, the function $g(n) = 2^{-2^{2^n}}$ is computable in time polynomial in n while its logarithm $f(n) = \ln n = -2^{2^n} \ln 2$ is not polynomial-time computable

A.2 Computability of the Power Function

In this appendix we prove Lemma 49 by showing that for every polynomial-time computable $\alpha > 0$ the function x^α is polynomial-time computable for $x \in [0, 1]$.

The identity $x^\alpha = e^{\alpha \ln x}$ can be used to show that x^α is polynomial-time computable on any separated from 0 interval $[\delta, A]$, where $0 < \delta < A < +\infty$. For $[0, 1]$ the identity cannot be applied directly, because $\ln x$ loses polynomial-time computability and its value tends to infinity as x approaches 0. The algorithm we will describe deals with this anomaly.

Let $N \geq 1$ be the smallest integer such than $N \geq 1/\alpha$.

Suppose that we are required to calculate x^α accurate to within 2^{-m}. We will start by requesting a dyadic approximation to x accurate to within $2^{-mN}/4$. Let d be such an approximation.

If $d \leq 2^{-mN}3/4$, we can conclude that $x \leq 2^{-mN}$ and therefore $x^\alpha \leq 2^{-mN\alpha} \leq 2^{-m}$. We thus output an appropriate dyadic approximation to 0.

Otherwise $x > 2^{-mN}/2$ and we can find an integer k such that $x \in [k2^{-mN}/2, (k+2)2^{-mN}/2]$, where $1 \leq k \leq 2^{mN+1} - 2$. Further, we can find an integer $l \geq 0$ such that $k \geq 2^l$ but $k + 2 \leq 4 \cdot 2^l = 2^{l+2}$ and $2^{l+2}2^{-mN}/2 = 2^{l-mN-1} \leq 1$. We get

$$x \in \left[\frac{1}{2}2^{l-mN}, 2 \cdot 2^{l-mN}\right] \subseteq (0, 1] \ .$$

Take $\tilde{x} = x/(2 \cdot 2^{l-mN})$. We get $\tilde{x} \in [1/4, 1]$ and

$$x^\alpha = \tilde{x}^\alpha \left(\frac{1}{2^\alpha}\right)^{mN-l-1} \quad .$$

The value of x^α is thus the product of $\tilde{x}^\alpha$ and $mN - l - 1 \geq 0$ instances of $1/2^\alpha$. It follows from (33) that it is sufficient to calculate these values accurate to within $2^{-m}/(2(mN-l))$. This can be done in time polynomial in m and the multiplications can be carried out in time polynomial in m.

A.3 Enumerations of Strategies

In this section we discuss the existence of (effective) enumerations of strategies w.r.t. a game $\mathfrak{G}$. The discussion provides the background for Appendix C.

There always exists an enumeration (not necessarily effective) of all computable strategies simply because there are countable many of them. Let us show that a similar property is not uncommon for polynomial-time computable strategies. We start by enumerating all vector-valued functions.

Lemma 54. *For every finite Ω and positive integer K there is an effective enumeration of Turing machines such that*

1. *every machine in the enumeration calculates a function from Ω^* to $\mathbb{R}^K$ and the running time of the machine does not exceed a polynomial in the length of the inputs;*

2. *every polynomial-time computable function from Ω^* to $\mathbb{R}^K$ is calculated by some machine in the enumeration.*

Proof. We will show how to construct an enumeration of machines calculating polynomial-time computable functions from Ω^* to $\mathbb{R}$. An enumeration of machines computing functions from Ω^* to $\mathbb{R}^K$ can be easily constructed from it.

There is an effective enumeration of all Turing machines taking a pair of a string from Ω^* and a unary number as inputs. Take a machine of this kind and a polynomial $p(n,m)$ with integer coefficients. Let us alter the machine in the following way. Given an input $\boldsymbol{x} = \omega_1\omega_2\ldots\omega_n$ and precision m we run the original machine on each of the pairs $(\boldsymbol{x}, i)$, $i = 1, 2, \ldots, m$, limiting the calculation time to $p(n,i)$. Then we take the maximum $i \le m$ with the following properties:

- the output on $(\boldsymbol{x}, j)$, $j = 1, 2, \ldots, i$, is a representation r_j of a dyadic number $d_j = \mathrm{d}(r_j)$ and the precision of r_j is j;

- for all $j, l = 1, 2, \ldots, i$ the difference between d_j and d_l is consistent with them being approximations of the same number, i.e.,

$$|d_j - d_l| \le 2^{-j} + 2^{-l} \ .$$

If no such i can be found, we output the dyadic representation of 0 with precision m. Otherwise we output the representation of d_i adjusted to have precision m.

It is easy to see that on a fixed $\boldsymbol{x}$ the outputs d_m for different ms form a Cauchy sequence converging to $y \in \mathbb{R}$ and $|d_m - y'| \leq 2^{-m}$. This procedure can be implemented in time polynomial in $|\boldsymbol{x}|$ and m and if the original machine computed a function from Ω^* to $\mathbb{R}$ the new procedure will compute the same function.

By enumerating all machines and all polynomials we prove the lemma. $\square$

The question of enumerating functions from Ω^* to $\Gamma \subseteq \mathbb{R}^K$ is difficult and the general answer probably depends on computability properties of Γ. We will investigate one simple special case.

Lemma 55. *For every finite Ω and positive integer K there is an effective enumeration (in the sense of Lemma 54) of all polynomial-time computable functions from Ω^* to the simplex*

$$\mathbb{P}_K = \left\{ (p_1, p_2, \ldots, p_K) | \sum_{i=1}^{K} p_i = 1 \text{ and } p_i \in [0,1], i = 1, 2, \ldots, K \right\} \subseteq \mathbb{R}^K \quad .$$

Proof. Let us describe a modification of a polynomial-time Turing machine computing a function $f = (f_1, f_2, \ldots, f_K)$ from Ω^* to $\mathbb{R}^K$ such that the resulting machine still runs in polynomial time and computes a function with the range $\mathbb{P}_K$ and the function coincides with the one computed by the original machine provided its range was $\mathbb{P}_K$.

First let us make sure that the function maps Ω^* to $[0, +\infty)^K$. The machine can be modified as follows. After the machine outputs a dyadic representation of $d = (d_1, d_2, \ldots, d_K)$, we replace each d_k with $\max(d_k, 0)$, $k = 1, 2, \ldots, K$. Clearly, the new machine can be made to run in polynomial time and it calculates a function from Ω^* to $[0, +\infty)^K$; whenever $f(\boldsymbol{x})$ had its value in $[0, +\infty)^K$, it remains unchanged. Let us assume $f_k(\boldsymbol{x}) \geq 0$, $k = 1, 2, \ldots, K$, from now on.

Secondly let us force the value $f(\boldsymbol{x})$ into $\mathbb{P}_K$. Let l be the minimum integer such that $2^l \geq K$. Whenever we are asked to calculate the value of $f(\boldsymbol{x})$ accurate to within 2^{-m}, we start by calculating each $f_k(\boldsymbol{x})$ accurate to within $2^{-(l+2)}$. Let the result be $d = (d_1, d_2, \ldots, d_K)$.

If $f(\boldsymbol{x}) \in \mathbb{P}_K$, then the inequalities $0.75 \leq \sum_{k=1}^{K} d_k \leq 1.25$ must hold. Let us check if they indeed hold. If either of the two does not hold, the calculation of $f(\boldsymbol{x})$ cannot possibly result in a value from $\mathbb{P}_K$. Let us abandon it altogether and output an appropriate dyadic approximation to $(1/K, 1/K, \ldots, 1/K)$. Now suppose that both the inequalities hold. They guarantee that $0.5 \le \sum_{k=1}^{K} f_k(\boldsymbol{x}) \le 1.5$ and therefore for all $k = 1, 2, \ldots, K$ the values $f_k(\boldsymbol{x}) / \sum_{k=1}^{K} f_k(\boldsymbol{x})$ can be computed in polynomial time. We then calculate those values accurate to within 2^{-m} and output them.

The calculation always results in values from $\mathbb{P}_K$ and if the original function f always had its values in $\mathbb{P}_K$, it remains unchanged. □

Appendix B. Weak Mixability and Computability

In this appendix we consider variations of the weak mixability property. In particular, we prove Proposition 3.

B.1 Effective Mixability of Bounded Games

If an extra requirement that the loss function is bounded is imposed, then the conclusion of Proposition 3 can be strengthened.

Lemma 56. *If a (polynomial-time) computable game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ has a convex prediction space Γ and a convex in the second argument and bounded loss function λ, then for every two (polynomial-time) computable strategies $\mathfrak{A}_1$ and $\mathfrak{A}_2$ w.r.t. $\mathfrak{G}$ there is a (polynomial-time) computable strategy $\mathfrak{A}$ w.r.t. $\mathfrak{G}$ and a function $\alpha : \mathbb{N} \to \mathbb{R}$ such that $\alpha(n) = O(\sqrt{n})$ as $n \to +\infty$ and*

$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}) \le \min\left(\mathrm{Loss}_{\mathfrak{A}_1}(\boldsymbol{x}), \mathrm{Loss}_{\mathfrak{A}_2}(\boldsymbol{x})\right) + \alpha(|\boldsymbol{x}|) \tag{34}$$

for all finite sequences $\boldsymbol{x} \in \Omega^*$.

Proof. The proof uses Remark 10 from [KV08]. Let

$$\gamma_t^{(j)} = \gamma_t^{(j)}(\omega_1 \omega_2 \ldots \omega_{t-1})$$

be the prediction output by $\mathfrak{A}_j$ on step t, where $\omega_1, \omega_2, \ldots, \omega_{t-1}$ are the outcomes observed on previous steps. Take

$$w_t^{(j)} = e^{-\sum_{i=1}^{t-1} \lambda\left(\omega_i, \gamma_i^{(j)}\right)/\sqrt{t}}$$

and

$$p_t^{(j)} = \frac{w_t^{(j)}}{w_t^{(1)} + w_t^{(2)}} \ ,$$

$j = 1, 2$, and let $\mathfrak{A}$ output

$$\gamma_t = p_t^{(1)} \gamma_t^{(1)} + p_t^{(2)} \gamma_t^{(2)} \ .$$

The convexity of Γ implies that this is a valid prediction. It is proven in [KV08] that the resulting strategy satisfies (34).

Let us show that $\mathfrak{A}$ is (polynomial-time) computable given the conditions of the lemma. Computability can be obtained straightforwardly. In what follows we assume that $\mathfrak{G}$, $\mathfrak{A}_1$, and $\mathfrak{A}_2$ are polynomial-time computable and show that $\mathfrak{A}$ is polynomial-time computable.

The expressions $e^{-\lambda\left(\omega_t,\gamma_t^{(j)}\right)}$, $j = 1, 2$ are polynomial-time computable as function on Ω^* by the definitions of polynomial-time computable games and strategies. Their cumulative $(t-1)$-products

$$Q_t^{(j)} = e^{-\sum_{i=1}^{t-1}\lambda\left(\omega_i,\gamma_i^{(j)}\right)} = \prod_{i=1}^{t-1} e^{-\lambda\left(\omega_i,\gamma_i^{(j)}\right)}$$

are polynomial-time computable by Part 2 of Corollary 51. Since $\lambda(\omega,\gamma) \le L$ for all outcomes $\omega \in \Omega$ and predictions $\gamma \in \Gamma$, we get that $e^{-L(t-1)} \le Q_t^{(j)} \le 1$ and $\ln Q_t^{(j)}$ is polynomial-time computable by Part 4 of Corollary 51. The function $1/\sqrt{t} = (1/t)^{1/2}$, is polynomial-time computable by Part 4 of Corollary 51 and Lemma 49. Thus

$$w_t^{(j)} = e^{\frac{1}{\sqrt{t}}\ln Q_t^{(j)}}$$

is also polynomial-time computable. The lower bound $e^{-L(t-1)/\sqrt{t}} \le w_t^{(j)}$ ensures that $p_t^{(j)}$ is polynomial-time computable. Finally, γ_t is polynomial-time computable. □

B.2 Effective Mixability of Unbounded Games

We prove Proposition 3 in the unbounded case using the following trick. The same algorithm as in Lemma 34 will be used, but it will be preceded by a "truncation" procedure.

Lemma 57. *For every game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ and $\varepsilon > 0$ there is a finite set of predictions $\Gamma^{(\varepsilon)} \subseteq \Gamma$ such that for all $\tilde{\gamma} \in \Gamma$ and $\omega \in \Omega$ the value $\lambda(\omega, \tilde{\gamma})$ is finite and for all $\gamma \in \Gamma$ there is $\tilde{\gamma} \in \Gamma^{(\varepsilon)}$ such that for all $\omega \in \Omega$ the inequality*

$$\lambda(\omega, \tilde{\gamma}) \le \lambda(\omega, \gamma) + \varepsilon$$

holds.

If $\mathfrak{G}$ is (polynomial-time) computable, then $\Gamma^{(\varepsilon)}$ can be chosen to consist of (polynomial-time) computable points $\tilde{\gamma}_1, \tilde{\gamma}_2, \ldots, \tilde{\gamma}_k$ such that there is a computable mapping $\pi : \Gamma \to \{1, 2, \ldots, k\}$ satisfying

$$\lambda(\omega, \tilde{\gamma}_{\pi(\gamma)}) \le \lambda(\omega, \gamma) + \varepsilon$$

for all $\tilde{\gamma} \in \Gamma$ and $\omega \in \Omega$. If $e^{-\lambda(\omega,\gamma)}$ can be computed for all $\gamma \in \Gamma$ in time depending only on the required precision (e.g., if $\mathfrak{G}$ is polynomial-time computable) then $\Gamma^{(\varepsilon)}$ and π can be chosen so that $\pi(\gamma)$ is computable (by an oracle Turing machine) in finite time bounded from above by $T = T(\varepsilon)$ independent of γ.

Proof. Lemma 33 implies that there is a finite set $\widehat{\Gamma}^{(\varepsilon/3)}$ such that for any $\gamma \in \Gamma$ there is $\gamma^* \in \widehat{\Gamma}^{(\varepsilon/3)}$ such that $\lambda(\omega, \gamma^*) \leq \lambda(\omega, \gamma) + \varepsilon/3$ for every $\omega \in \Omega$.

It follows from continuity of λ (implied by the definition of a game) that we can approximate each $\gamma^* \in \Gamma$ by $\bar{\gamma} \in \Gamma$ such that all values $\lambda(\omega, \bar{\gamma})$ are finite and if $\lambda(\omega, \gamma^*)$ was finite then $|\lambda(\omega, \bar{\gamma}) - \lambda(\omega, \gamma^*)| \leq \varepsilon/3$. For a non-computable game $\mathfrak{G}$ let $\Gamma^{(\varepsilon)}$ consist of $\bar{\gamma}$s approximating γ^*s from $\widehat{\Gamma}^{(\varepsilon/3)}$.

The definition of a (polynomial-time) computable game implies that (polynomial-time) computable points are dense in Γ and therefore every $\bar{\gamma}$ can be approximated by a (polynomial-time) computable $\tilde{\gamma}$ at a cost of no more than further $\varepsilon/3$ in the values of λ. For a (polynomial-time) computable game $\mathfrak{G}$ let $\widetilde{\Gamma}^{(\varepsilon)}$ consist of $\tilde{\gamma}$s approximating $\bar{\gamma}$s approximating γ^*s from $\widehat{\Gamma}^{(\varepsilon/3)}$.

Let us assume that $\mathfrak{G}$ is computable and construct the mapping π. Take $\widetilde{\Gamma}^{(\varepsilon/2)} = \{\tilde{\gamma}_1, \tilde{\gamma}_2, \ldots, \tilde{\gamma}_k\}$ that exists by what we have already proven. For every $\gamma \in \Gamma$ there is $\tilde{\gamma}_i$ such that

$$\lambda(\omega, \tilde{\gamma}_i) \leq \lambda(\omega, \gamma) + \varepsilon/2$$

i.e.,

$$\frac{e^{-\lambda(\omega,\gamma)}}{e^{-\lambda(\omega,\tilde{\gamma}_i)}} \leq e^{\varepsilon/2} \tag{35}$$

for all $\omega \in \Omega$.

Since all $\lambda(\omega, \tilde{\gamma}_i)$ are finite and there are finitely many of them, there is an upper bound on them and therefore some $\theta > 0$ such that $e^{-\lambda(\omega,\tilde{\gamma}_i)} \geq \theta$ for all $i = 1, 2, \ldots, k$ and $\omega \in \Omega$. Suppose that instead of some numbers $x \in [0, 1]$ and $y \in [\theta, 1]$ we know their approximations $x + \delta_1$ and $y + \delta_2$, where $|\delta_1|, |\delta_2| \leq \delta$ and $\theta > \delta$. The accuracy of their ratio can be bounded from above as follows:

$$\left|\frac{x}{y} - \frac{x + \delta_1}{y + \delta_2}\right| \leq \frac{x|\delta_2| + y|\delta_1|}{|y(y + \delta_2)|} \leq \frac{2\delta}{(\theta - \delta)^2} \ .$$

Thus if we know $e^{-\lambda(\omega,\tilde{\gamma})}$ and $e^{-\lambda(\omega,\tilde{\gamma}_i)}$ accurate to within $\min(\sigma\theta^2/8, \theta/2)$, we can calculate their ratio accurate to within σ.

Take $\gamma \in \Gamma$. Let us calculate all ratios $e^{-\lambda(\omega,\gamma)}/e^{-\lambda(\omega,\tilde{\gamma}_i)}$, $i = 1, 2, \dots, k$, $\omega \in \Omega$, accurate to within $e^{\varepsilon} - e^{\varepsilon/2}$. Since (35) holds for some i, we will find i for which

$$\frac{e^{-\lambda(\omega,\gamma)}}{e^{-\lambda(\omega,\tilde{\gamma}_i)}} \leq e^{\varepsilon}$$

and therefore

$$\lambda(\omega, \tilde{\gamma}_i) \leq \lambda(\omega, \gamma) + \varepsilon$$

for all $\omega \in \Omega$. Let $\pi(\gamma)$ be the minimal i we have found. The set $\Gamma^{(\varepsilon)} = \widetilde{\Gamma}^{(\varepsilon/2)}$ and the mapping π satisfy the requirements of the theorem for ε. $\square$

Now we can describe how the strategy $\mathfrak{A}$ can be constructed given two (polynomial-time) computable strategies $\mathfrak{A}_1$ and $\mathfrak{A}_2$ w.r.t. a (polynomial-time) computable game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$. For $\varepsilon > 0$ there is a set $\Gamma^{(\varepsilon)} = \{\tilde{\gamma}_1, \tilde{\gamma}_2, \dots, \tilde{\gamma}_k\}$ and a mapping π defined in the lemma. The strategies $\mathfrak{A}_1^{(\varepsilon)}$ and $\mathfrak{A}_2^{(\varepsilon)}$ that output $\gamma_{\pi(\gamma)}$ whenever the respective strategies output γ are (polynomial-time) computable and

$$\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}_i^{(\varepsilon)}}(\boldsymbol{x}) \leq \mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}_i}(\boldsymbol{x}) + \varepsilon|\boldsymbol{x}|$$

for $i = 1, 2$ and all $\boldsymbol{x} \in \Omega^*$. We can use the construction from Lemma 34 to obtain strategy $\mathfrak{A}$ such that

$$\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}}(\boldsymbol{x}) \leq \min\Big(\mathrm{Loss}_{\mathfrak{A}_1^{(\varepsilon)}}(\boldsymbol{x}), \mathrm{Loss}_{\mathfrak{A}_2^{(\varepsilon)}}(\boldsymbol{x})\Big) + o(|\boldsymbol{x}|)$$

as $|\boldsymbol{x}| \to \infty$.

Remark 58. This construction is a simplified version of that from Section 6.3 and Appendix E from [KV08]. The full construction of [KV08] can also be implemented and would lead to a better bound but at the cost of complicated restrictions on the game, such as that the ε-nets and some relevant constants should be obtainable effectively given $\varepsilon > 0$. For this paper we choose a weaker but more general approach.

B.3 Weak Mixability of Countable Families

The discussion of this section provides the background for Appendix C.

The results from [KV08] can be extended to cover countable families of strategies.

Lemma 59. *Let a game $\mathfrak{G}$ be weakly mixable. Then there are functions $\alpha_1, \alpha_2, \ldots : \mathbb{N} \to \mathbb{R}$ such that for all $i = 1, 2, \ldots$ we have $\alpha_i(n) = o(n)$ as $n \to \infty$ and for every sequence of strategies $\mathfrak{A}_1, \mathfrak{A}_2, \ldots$ w.r.t. $\mathfrak{G}$ there is a strategy $\mathfrak{A}$ such that*

$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}) \leq \mathrm{Loss}_{\mathfrak{A}_i}^{\mathfrak{G}}(\boldsymbol{x}) + \alpha_i(|\boldsymbol{x}|)$$

for all $\boldsymbol{x} \in \Omega^*$ and $i = 1, 2, \ldots$.

Proof. The lemma follows from [KV08]. For bounded losses the result is formulated in Remark 12. For unbounded losses a construction similar to that from Section 6.3 and Appendix E can be used. Indeed, consider (7) in [KV08]. The term $cL^2\sqrt{n}$ can be approached in the same fashion as in Appendix E and the term $\frac{1}{c}\frac{1}{q_i}\sqrt{n}$ will lead to a small extra term by Lemma 21. $\square$

The situation with effective families is more complicated. We will say that a computable game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ is *computably countably very weakly mixable* if for every $\varepsilon > 0$ there are functions $\alpha_{i,\varepsilon} : \mathbb{N} \to \mathbb{R}$ such that $\alpha_{i,\varepsilon}(n) = o(n)$ as $n \to +\infty$ for $i = 1, 2, \ldots$ and for every computable family of computable prediction strategies $\mathfrak{A}_1, \mathfrak{A}_2, \ldots$ there is a computable strategy $\mathfrak{A}$ such that

$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}) \leq \mathrm{Loss}_{\mathfrak{A}_i}(\boldsymbol{x}) + \varepsilon|\boldsymbol{x}| + \alpha_{i,\varepsilon}(|\boldsymbol{x}|) \tag{36}$$

for all finite sequences $\boldsymbol{x} \in \Omega^*$ and $i = 1, 2, \ldots$.

Lemma 60. *If a computable game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ has a convex prediction space Γ and a convex in the second argument loss function λ, then it is computably countably very weakly mixable.*

Proof. Let $\mathfrak{A}_1, \mathfrak{A}_2, \ldots : \Omega^* \to \Gamma$ be a computable family of computable strategies.

First let us truncate predictions using Lemma 57. (For bounded games this step is unnecessary and we can drop $\varepsilon|\boldsymbol{x}|$ from (36) as before.) Let

$$\gamma_t^{(j)} = \gamma_t^{(j)}(\omega_1\omega_2\ldots\omega_{t-1})$$

be the truncated prediction output by $\mathfrak{A}_j$ on step t, where $\omega_1, \omega_2, \ldots, \omega_{t-1}$ are the outcomes observed on previous steps.

Fix a computable sequence of positive numbers summing to 1, e.g., $q_j = 2^{-j}$. Take

$$w_t^{(j)} = q_j e^{-\sum_{i=1}^{t-1} \lambda\left(\omega_i, \gamma_i^{(j)}\right)/\sqrt{t}}$$

and

$$p_t^{(j)} = \frac{w_t^{(j)}}{\sum_{i=1}^{\infty} w_t^{(i)}} \ ,$$

$j = 1, 2, \ldots$, $t = 1, 2, \ldots$. The strategy $\mathfrak{A}$ outputting the predictions

$$\gamma_t = \sum_{i=1}^{\infty} p_t^{(i)} \gamma_t^i$$

on step t satisfies (36).

In order to show that γ_t is computable, we need to upper bound the tail of the series. Let $L < +\infty$ be the maximum of $\lambda(\omega, \gamma)$ for all outcomes $\omega \in \Omega$ and "truncated" predictions γ. We get $q_j e^{-L\sqrt{t}} \leq w_t^{(j)} \leq q_j$ and $p_t^{(j)} \leq q_j e^{L\sqrt{t}}$. By the definition of a game the set Γ is compact and therefore bounded. Let $G > 0$ be such that each component of each $\gamma \in \Gamma$ does not exceed G. Then by taking m terms from the infinite sum for γ_t we get an approximation accurate to within $G 2^{-m} e^{L\sqrt{t}}$. If we know computable upper bounds to L and G, we can work out how many terms to take to approximate γ_t with a given precision.

In order to calculate $p_t^{(j)}$ we need to approximate the sum $\sum_{i=1}^{\infty} w_t^{(i)}$. The above inequalities provide lower and upper bounds to its initial segments. □

Appendix C. Why Complexities Differ

In this appendix we show that complexities introduced in Sections 3.1 and 3.2 differ. We first show that different types of complexities differ and then proceed to show that computability matters.

C.1 Lower vs Upper and Uniform vs Non-Uniform

Consider a game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ with $|\Omega| = M$. Let S be the set of superpredictions for $\mathfrak{G}$. Take $A = \min\{x \in \mathbb{R} \mid (x, x, \ldots, x) \in S\}$ (the point $(A, A, \ldots, A)$ is where the diagonal of the positive orthant meets S) and take $m_i = \min_{\gamma \in \Gamma} \lambda(\omega^{(i)}, \gamma)$ for $i = 0, 1, \ldots, M-1$. Clearly, $m_i \leq A$ for all $i = 1, 2, \ldots, M-1$. If $m_0 = m_1 = \ldots = m_{M-1} = A$, we will call the game *symmetric degenerate*. Clearly, for a symmetric degenerate game the set of superpredictions equals $[A, +\infty]^M$ and every asymptotic complexity of every language equals A; if the game is computable, the same applies to effective complexities and if it is polynomial-time computable, the same applies to polynomial-time complexities.

Proposition 61. *If a game $\mathfrak{G}$ is not symmetric degenerate, neither two of the complexities $\overline{\mathrm{AC}}$, $\underline{\mathrm{AC}}$, $\overline{\overline{\mathrm{AC}}}$, or $\underline{\underline{\mathrm{AC}}}$ coincide on all infinite sets of finite sequences or all non-empty sets of infinite sequences. If the game is computable and not symmetric degenerate, the same is true of effective complexities $\overline{\mathrm{ACE}}$, $\underline{\mathrm{ACE}}$, $\overline{\overline{\mathrm{ACE}}}$, and $\underline{\underline{\mathrm{ACE}}}$. If the game is polynomial-time computable and not symmetric degenerate, the same is true of polynomial-time complexities complexities $\overline{\mathrm{ACP}}$, $\underline{\mathrm{ACP}}$, $\overline{\overline{\mathrm{ACP}}}$, and $\underline{\underline{\mathrm{ACP}}}$.*

Proof. It is sufficient to show that the complexities differ on languages consisting of infinite sequences.

If a game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ is not symmetric degenerate there is i such that $m_i < A$. Let m_i be achieved by $\lambda(\omega^{(i)}, \cdot)$ on γ_0. The prediction γ_0 may lead to infinite losses on other outcomes but for every $\varepsilon > 0$ there is γ_ε such that $\lambda(\omega^{(i)}, \gamma_\varepsilon) \leq m_i + \varepsilon$ but $\lambda(\omega^{(j)}, \gamma_\varepsilon)$ is bounded by a finite C for all $j = 0, 1, \ldots, M-1$. If the game is (polynomial-time) computable, γ_ε can be chosen to be (polynomial-time) computable.

The strategy predicting γ_ε suffers loss not exceeding $(m_i + \varepsilon)n$ on a string $\left(\omega^{(i)}\right)^n$ and loss not exceeding $C|\boldsymbol{x}|$ on every finite string $\boldsymbol{x} \in \Omega^*$. If the game is (polynomial-time) computable, the strategy is (polynomial-time) computable.

On the other hand, for every strategy $\mathfrak{A}$ and every positive integer n there is a string $\boldsymbol{x}$ of length n such that $\mathrm{Loss}_{\mathfrak{A}}(\boldsymbol{x}) \geq A|\boldsymbol{x}|$. Indeed, $\boldsymbol{x}$ can be constructed by induction. On step T let $\mathfrak{A}$ output a prediction γ_t. There must be an outcome $\omega^{(j)}$ such that $\lambda(\omega^{(j)}, \gamma_t) \geq A$ (or A could have been decreased) and we can choose $\omega^{(j)}$ to be the T-th element of x.

In order to distinguish between upper and lower complexities consider the language $L = \prod_{i=0}^{\infty} \Xi_{2^{2^i}} \subseteq \Omega^\infty$, where $\Xi_n = \left\{ \left(\omega^{(i)}\right)^n \right\} \times \Omega^n$. Its lower complexity equals m_i while upper complexity equals A.

In order to distinguish between upper and lower complexities consider the language $L = \cup_{n=0}^{\infty} \left(\Omega^n \times \left\{ \left(\omega^{(i)}\right)^\infty \right\} \right)$. We have $\overline{\overline{\mathrm{AC}}}(L) = \underline{\underline{\mathrm{AC}}}(L) = m_i$ while $\overline{\mathrm{AC}}(L) = \underline{\mathrm{AC}}(L) = A$. $\square$

Remark 62. Note an interesting property of degenerate games. Suppose that the set of superpredictions for a game is $[A_0, +\infty] \times [A_1, +\infty] \times \ldots \times$

$[A_{M-1}, +\infty]$, where not all A_is are equal. It is natural to call such a game *asymmetric degenerate*. For such a game it only make sense to ever predict γ such that $\lambda(\omega^{(i)}, \gamma) = A_i$, $i = 0, 1, \ldots, M-1$. All other predictions (if any) always lead to loss greater than or equal to the loss caused by this γ. However the complexities still differ.

C.2 Effective vs Non-Effective

In this subsection we formulate sufficient conditions for effective and polynomial-time complexities to differ.

The argument in this section depends on the existence of enumerations of computable and polynomial-time computable strategies discussed in Appendix A.3 and weak mixability of countable families discussed in Appendix B.3.

Proposition 63. *Let S be the set of superpredictions for $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ with $\Omega = \left\{\omega^{(0)}, \omega^{(1)}, \ldots, \omega^{(M-1)}\right\}$ ($|\Omega| = M$). Let $A = \min\{x \in \mathbb{R} \mid (x, x, \ldots, x) \in S\}$ and $m_i = \min_{\gamma \in \Gamma} \lambda(\omega^{(i)}, \gamma)$ for $i = 0, 1, \ldots, M-1$. If $m_i < A$ for all $i = 0, 1, \ldots, M-1$ and $\mathfrak{G}$ is weakly mixable, then asymptotic complexities $\overline{\mathrm{AC}}$, $\overline{\overline{\mathrm{AC}}}$, $\underline{\mathrm{AC}}$, and $\underline{\underline{\mathrm{AC}}}$ w.r.t. $\mathfrak{G}$ differ from their effective analogues $\overline{\mathrm{ACE}}$, $\overline{\overline{\mathrm{ACE}}}$, $\underline{\mathrm{ACE}}$, and $\underline{\underline{\mathrm{ACE}}}$.*

If, moreover, $\mathfrak{G}$ is computable and computably countably very weakly mixable, and there is a computable enumeration of all polynomial-time computable strategies w.r.t. $\mathfrak{G}$, then asymptotic complexities $\overline{\mathrm{AC}}$, $\overline{\overline{\mathrm{AC}}}$, $\underline{\mathrm{AC}}$, and $\underline{\underline{\mathrm{AC}}}$ and effective complexities $\overline{\mathrm{ACE}}$, $\overline{\overline{\mathrm{ACE}}}$, $\underline{\mathrm{ACE}}$, and $\underline{\underline{\mathrm{ACE}}}$ differ from their polynomial analogues $\overline{\mathrm{ACP}}$, $\overline{\overline{\mathrm{ACP}}}$, $\underline{\mathrm{ACP}}$, and $\underline{\underline{\mathrm{ACP}}}$.

Proof. Let $m = \max_{i=0,1,\ldots,M-1} m_i < A$ and $\mathfrak{A}_1, \mathfrak{A}_2, \ldots$ be an enumeration of all computable strategies w.r.t. $\mathfrak{G}$. Lemma 59 implies that there is a strategy $\mathfrak{A}$ "capturing the power" of the family, i.e., such that

$$\mathrm{Loss}_{\mathfrak{A}}^{\mathfrak{G}}(\boldsymbol{x}) \le \mathrm{Loss}_{\mathfrak{A}_j}^{\mathfrak{G}}(\boldsymbol{x}) + \alpha_j(|\boldsymbol{x}|)$$

for all $\boldsymbol{x} \in \Omega^*$, where $\alpha_j(n) = o(n)$ as $n \to \infty$.

The language differentiating the complexities will consist of a single infinite string $\boldsymbol{x} = \omega_1\omega_2\ldots$. It is constructed by induction. Suppose that we have constructed $\omega_1, \omega_2, \ldots, \omega_{t-1}$. Let $\gamma_t = \mathfrak{A}(\omega_1, \omega_2, \ldots, \omega_{t-1})$ be the prediction output by $\mathfrak{A}$ on step t after seeing $\omega_1, \omega_2, \ldots, \omega_{t-1}$. It follows from the definition of A that there is $\omega_t \in \Omega$ such that $\lambda(\omega_t, \gamma_t) \ge A$; if there are several such ωs, take the first one in the enumeration $\Omega =$

$\{\omega^{(0)}, \omega^{(1)}, \ldots, \omega^{(M-1)}\}$. We get

$$An \le \mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}}(\boldsymbol{x}|_n) \le \mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{A}_j}(\boldsymbol{x}|_n) + \alpha_j(n)$$

for all $n, j = 1, 2, \ldots$ and therefore for the language $L = \{\boldsymbol{x}\}$ we have $\overline{\mathrm{ACE}}(L), \overline{\overline{\mathrm{ACE}}}(L), \underline{\mathrm{ACE}}(L), \underline{\underline{\mathrm{ACE}}}(L) \ge A$.

On the other hand, let $\mathfrak{S}$ be the strategy that on step t predicts γ minimising $\lambda(\omega_t, \gamma)$. We get

$$\mathrm{Loss}^{\mathfrak{G}}_{\mathfrak{S}}(\boldsymbol{x}|_n) \le mn$$

and therefore $\overline{\mathrm{AC}}(L), \overline{\overline{\mathrm{AC}}}(L), \underline{\mathrm{AC}}(L), \underline{\underline{\mathrm{AC}}}(L) \le m < A$.

To differentiate polynomial-time complexities from effective we use the same argument with slight modifications. Take $\varepsilon = (A - m)/3$ and a computable strategy $\mathfrak{A}$ capturing the power of all polynomial-time computable strategies up to $\varepsilon|\boldsymbol{x}|$ as in the definition of computably countably very weak mixability. The construction of $\boldsymbol{x}$ is again by induction. On step t we calculate $e^{-\lambda(\omega,\gamma_t)}$, where γ_t is the prediction output by $\mathfrak{A}$ given the beginning of $\boldsymbol{x}$ that has been already constructed, accurate to within $e^{-A+\varepsilon} - e^{-A} > 0$. Since there is ω such that $\lambda(\omega, \gamma_t) \ge A$, we will be able to find ω such that $e^{-\lambda(\omega,\gamma)} \le e^{-A+\varepsilon}$. Let ω_t be the first such ω. We get $\overline{\mathrm{ACP}}(L), \overline{\overline{\mathrm{ACP}}}(L), \underline{\mathrm{ACP}}(L), \underline{\underline{\mathrm{ACP}}}(L) \ge A - \varepsilon$.

It is easy to see that the sequence $\boldsymbol{x}$ is computable and therefore there is a computable strategy $\mathfrak{S}$ that on step t outputs a computable approximation $\tilde{\gamma}$ to γ minimising $\lambda(\omega_t, \gamma)$ so that $\lambda(\omega_t, \tilde{\gamma}) \le m + \varepsilon$. We get $\overline{\mathrm{ACE}}(L), \overline{\overline{\mathrm{ACE}}}(L), \underline{\mathrm{ACE}}(L), \underline{\underline{\mathrm{ACE}}}(L), \overline{\mathrm{AC}}(L), \overline{\overline{\mathrm{AC}}}(L), \underline{\mathrm{AC}}(L), \underline{\underline{\mathrm{AC}}}(L) \le m + \varepsilon < A - \varepsilon$. $\square$

Acknowledgements

The authors were supported by EPSRC through the grants GR/R46670 “Complexity Approximation Principle and Predictive Complexity: Analysis and Applications”. Yuri Kalnishkan and Vladimir Vovk were also supported through the grant EP/F002998 “Practical competitive prediction”. Yuri Kalnishkan has been supported by the Leverhulme Trust through the research grant RPG-2013-047 “On-line Self-Tuning Learning Algorithms for Handling Historical Information”.

The authors would like to thank anonymous ALT and Information and Computation reviewers for their observations and suggestions. Yuri Kalnishkan thanks Alexey Savvateev, who introduced the lattice terminology to him, and Dmitry Adamskiy for helpful suggestions.

References

[Bir48] G. Birkhoff. *Lattice Theory.* American Mathematical Society, revised edition, 1948.

[CBL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

[DeG70] M. H. DeGroot. *Optimal statistical decisions.* McGraw-Hill, 1970.

[Egg58] H. G. Eggleston. *Convexity.* Cambridge University Press, Cambridge, 1958.

[FL05] L. Fortnow and J. H. Lutz. Prediction and dimension. *Journal of Computer and System Sciences*, 70(4):570–589, 2005.

[GD04] P. D. Grünwald and A. P. Dawid. Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory. *The Annals of Statistics*, 32(4):1367–1433, 2004.

[GO03] B. R. Gelbaum and J. M. H. Olmsted. *Counterexamples in Analysis.* Dover Publications, 2003.

[Hit03] J. M. Hitchcock. Fractal dimension and logarithmic loss unpredictability. *Theoretical Computer Science*, 304:431–441, 2003.

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

[Hut04] M. Hutter. *Universal artificial intelligence: Sequential decisions based on algorithmic probability.* Springer, 2004.

[Kal02] Y. Kalnishkan. General linear relations among different types of predictive complexity. *Theoretical Computer Science*, 271:181–200, 2002.

[Ko91] Ker-I Ko. *Complexity theory of real functions.* Birkhäuser, 1991.

[KV08] Y. Kalnishkan and M. V. Vyugin. The weak aggregating algorithm and weak mixability. *Journal of Computer and System Sciences*, 74(8):1228–1244, 2008.

[KVV04] Y. Kalnishkan, V. Vovk, and M. V. Vyugin. Loss functions, complexities, and the Legendre transformation. *Theoretical Computer Science*, 313(2):195–207, 2004.

[KVV05] Y. Kalnishkan, V. Vovk, and M. V. Vyugin. How many strings are easy to predict? *Information and Computation*, 201(1):55–71, 2005.

[KVV07] Y. Kalnishkan, V. Vovk, and M. V. Vyugin. Generalised entropy and asymptotic complexities of languages. In *20th Annual Conference on Learning Theory, COLT 2007*, pages 293–307. Springer, 2007.

[LV08] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 3rd edition, 2008.

[Rya86] B. Ya. Ryabko. Noiseless coding of combinatorial sources, Hausdorff dimension, and Kolmogorov complexity. *Problems of Information Transmission*, 22(3):170–179, 1986.

[Rya11] D. Ryabko. On the relation between realizable and nonrealizable cases of the sequence prediction problem. *The Journal of Machine Learning Research*, pages 2161–2180, 2011.

[VW98] V. Vovk and C. J. H. C. Watkins. Universal portfolio selection. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 12–23. ACM Press, 1998.

[Wei00] K. Weihrauch. *Computable Analysis*. Springer, 2000.