On the Power of Evolution

MSc Thesis *(Afstudeerscriptie)*

written by

Georgios Sarailidis
(born November 27th, 1987 in Kozani, Greece)

under the supervision of **Ronald de Wolf**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam.*

Date of the public defense:	**Members of the Thesis Committee:**
August 9th, 2013	Benedikt Löwe Roeland Merks Ronald de Wolf Jelle Zuidema

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

To all the fruit I ate.

Abstract

We introduce a framework designed to mathematically study and model the 'evolvability' of entities. Our use of 'evolvability' is different than the one found in the context of biology. We roughly understand an entity as being 'evolvable' if, with non-negligible probability, it can arise as the result of an evolutionary process. However, this is not a formal definition. An essential part of our work is to examine how one may arrive at a formal definition for evolvability. The exact definition will depend on one's application of interest. With such a definition in hand one may approach questions like "is there a mechanism that can make such-and-such entities co-evolve under such-and-such initial conditions, with non-negligible probability, in a non-astronomical amount of time?" Ultimately, we are interested in assessing the power of a given evolutionary process in terms of the evolutionary products it can give rise to, the conditions and resources necessary for this to happen, and the respective probabilities of those events.

Acknowledgements

Thanx Ronald. Ronald gave me feedback on countless drafts and led me when I was in haze. The freedom in how to do things in this thesis would have been too much for me to handle all by myself. Thanx to my other professors too, it was fun with many, and to my examiners. And to Tanja, naturally. Thanx also to Katelijn, my yoga teacher; helped a lot with this thesis, yes. Thanx to all the mates around too. Thanx to the eastern-Asian movie makers. Thanx to noise-rock, post-punk, etc. Thanx to all the mates and the parties.

Contents

Chapter 1

Introduction

Nowadays, it is widely accepted that life has evolved through the years. According to neo-Darwinian evolution, starting with simple forms, adaptational forces have managed to create increasingly complex organisms. The term 'evolution' is quite an abstract notion; it signifies some kind of change of some population along time. Hence, for example, it does not necessarily entail natural selection, which is an integral part of neo-Darwinism. In this text, however, we will be interested only in evolution in line with neo-Darwinism, encompassing the idea of mutation and natural selection, and we will mean the latter when we say 'evolution'.

In this work we are interested in the *complexity* of evolutionary innovations. For example, how complex is it for a mammal to appear as a result of a certain evolutionary process? Under what kind of initial conditions? Questions such as these serve as motivation and lead our models. However, they may be overwhelmingly intricate to tackle. What are the defining characteristics of a mammal? How can we represent those in a mathematical framework? How can we represent environmental conditions on earth? Can we find a set of such conditions to which we may restrict our attention and expect to get satisfactorily accurate results? The latter questions reside well in the realm of biology.

In this work we abstract away from these details and focus on an abstract process of evolution. We form a Meta-Model which allows for evolution to be studied through different levels of abstraction, depending on the setting. The user of the framework is given the opportunity to define things such as the details of the environment, the evolutionary process, the entities in question, etc.

Note that given the nature of this work we are bound to use some biology jargon early on. As such, the reader may want to jump to Section 2.1, for a clarification of the respective terms, before continuing with this introduction, or she may just go back and forth whenever she comes across an unknown term.

There has been prior work on the issue of the complexity of evolutionary innovations. Leslie Valiant [Val09], through the perspective of computational learning theory, introduced a framework mostly concerned with classifying the complexity of evolutionary innovations, e.g., certain complex organisms. Complexity in Valiant's framework is captured by the notion of *evolvability*, where the latter is roughly understood as the possibility/probability of a certain innovation being the result of an evolutionary process that has started from some random initial state.[1]

Our approach is similar, in the sense that we are interested in the notion of evolvability. In fact, Valiant's work has been our starting point, but we will deviate from it substantially. Our main concern in this work has been to avoid any unnecessary assumptions in order to stay as close as possible to the essentials of the evolutionary process.

Ultimately, we expect the modelling and subsequent understanding of evolvability to give insight into questions in biology and possibly into philosophical questions[2] that have to do with evolution.

The following section sketches our approach.

1.1 The Plan

In this work we try to understand the complexity of macroscopic evolutionary innovation. Let us try to clarify this statement. Firstly, what is 'macroscopic evolutionary innovation'? It is innovation, e.g., the emergence of a certain organism or character, that is the result of an evolutionary process that has been executed for a long period of time.[3]

In the setting of computational complexity we are interested in how 'complex' it is for certain problems to be solved through a computational process. In analogy with that, in this work we are interested in how 'complex' it is for certain innovations to happen as the result of an evolutionary process.

---

[1] Note that the term 'evolvability' is used with different meaning in the context of biology. More on that in Chapter 2.

[2] For example, arguments about evolvability, understood as above, have been used to criticize the trust of biology in neo-Darwinism. Assessing such arguments partly motivates our work, even though in Appendix A.2.2 we argue that such claims tend to be doomed from the beginning because of the need and the difficulty to define the assumed initial conditions of the evolutionary system under question.

[3] For more details, the reader is referred again to Section 2.1 where those notions are defined.

We capture the notion of complexity with the notion of evolvability. In a more technical sense, we understand an innovation to be *evolvable* if it can arise as the result of an evolutionary process. Depending on the application we may restrict the evolutionary process in various ways. For example, we may restrict the possible mutation mechanisms, or require the evolutionary process to be in some sense efficient. One of the main aspects of the process that we may want to bound through the notion of efficiency is *time*, which can be thought of roughly as number of generations. In this setting, if it takes too long for a certain innovation to arise then we may want to deem it non-evolvable.

1.2 Into Perspective

There are probably many ways in which we can relate our work to existing research. In this section we will point out a few of those.

Mathematical work on evolution has assumed (at least) two different, and to a certain extent complementary in nature, trends. One [GS07, Gas07] that is focused on the evolution of actual entities that have existed on earth, their history and patterns, and a less empirical one [Kau93, Goo01, TH12, LOPA03] that is mainly concerned with how order arises as the result of limited interactions between the basic elements of life. Our work is closer to the latter trend.

Our framework is flexible enough so that we can speak about the properties of evolutionary systems in general, that is, not only about biological evolutionary systems. Similarly, Adami et al. [LOPA03], for example, consider the evolutionary origin of complexity in a setting where the entities of interest are computer programs. Fontana and his colleagues work on an abstract level too. In their work at the Fontana Laboratory [Fon13] they define a formal language for representing biochemical and biophysical aspects of interaction, which they use to analyze the effects of adaptational forces.

We understand evolvability as a certain kind of complexity. In biology there is a lot of research that considers the evolution of complexity. Kauffman [Kau93], for example, has taken a dynamical-systems approach in order to analyze the dynamics of evolution by focusing on the implications of the adaptational forces. McShea [McS96] considers four different definitions of complexity and how this complexity can be observed on specific cases of evolution on earth.

1.3 Overview

In Chapter 2 we give a first taste and somewhat clarify many of the basic notions and issues that will concern us in this text. In Chapter 3 we define and analyze Valiant's framework. In Chapter 4 we proceed to point out various shortcomings of this framework. In Chapter 5 we introduce our own framework, and in Chapter 6 we demonstrate how one may restrict our framework so as to explore certain specific applications. In Chapter 7 we introduce various problems and use our framework to formalize and analyze them. Chapter 8 concludes the text. Lastly, Appendix A contains various details and issues complementary to the main body of the text.

If the reader is mainly interested in our framework and not in connections with that of Valiant's, she may skip Chapters 3 and 4. Though, reading those chapters is sure to help one better understand our framework and we would advise at least a quick reading of the introduction of Chapter 3, Section 3.1, and Chapter 4.

Chapter 2

Preliminaries

In this work our focus will be on an abstract process of evolution that is driven mainly through mutation and natural selection. However, the knowledge of this process still comes from the understanding of how this process took place on earth and we should expect that the notion of evolution can become clear only in analogy to its manifestation on earth.

2.1 Concepts in Evolutionary Theory

In this section we give the definitions of various terms that we use throughout the text. Some of the definitions are the standard ones while others, such as that of natural selection and evolvability, are somewhat more particular to our purposes here. Be sure to read the remarks in the end of this section.

Evolution, or *evolutionary process*, is a process that drives the change of biological populations through time.

Evolutionary innovation commonly refers to some change that is the result of an evolutionary process. In general, we use the term to denote a large-scale change, in analogy to macro-evolution—defined later in this section. We may also use the term *innovation* just by itself wherever there is uncertainty whether the corresponding change can be the result of an evolutionary process.

The *genotype* of an organism corresponds to the hereditary information of the organism, i.e., its genetic makeup. The genotype should be distinguished from the phenotype it produces. An organism's *phenotype* is its observable body and behavior. The phenotype of an organism is determined to a large extent from its genotype. However, knowing the phenotype of an organism does not necessarily allow us to directly derive its genotype.

In biological classification, organisms are categorized into groups such as kingdom, genus, and species. These groups, known as *taxa*, form an hierarchy, where, e.g., the organisms of many different species may be members of the same genus. The hierarchy is known as the *taxonomic hierarchy* and has its roots in the work of Carolus Linnaeus.

Natural selection is a principle that automatically occurs given the occurrence of certain conditions. Specifically, “if *variation* exists for some trait, and a *fitness difference* is correlated with that trait, and the trait is to some degree *heritable* (determined by genetics), then the *trait distribution* will change [towards the variations of the trait that result in fitter individuals] over the life history of organisms in a single generation, and between generations.” [Car10] The process of change is called *adaptation*.

Assuming that the more fit an organism is, the more likely it is to reproduce, then natural selection seems somewhat like a tautological principle, since it roughly says that the fitter an organism is, i.e., the more likely to reproduce, the better its chances are to pass its genes to the next generations. All in all, when natural selection is present in a process then adaptation occurs in such a way that fitness increases in the long run.

Microevolution refers to “changes in gene frequencies within populations, under the influence of natural selection and random drift” [Rid04]. It is evolution in the small scale. *Macroevolution*, on the other hand, is evolution on the large scale. It refers to “the origin of higher taxa, such as the evolution of mammal-like reptiles into mammals, fish into tetrapods, and green algae into vascular plants. It also refers to long-term evolutionary trends, and to diversification, extinction, and replacements of higher taxa.” [Rid04]

Neo-Darwinism stands for Darwinian evolution through natural selection coupled with Mendelian genetics. That is, it corresponds to the idea that the dynamics of evolution are characterized by natural selection, and that evolution works by changing the genetic makeup of organisms along reproduction. Specifically, in the context of biology, it is accepted as the essential mechanism that drives (micro)evolution.

In biology, *evolvability* refers to how probable or easy it is for a certain organism to evolve into some new kind of organism. However, in this work we define the term quite differently. In this text, evolvability refers to how probable or easy it is for a certain organism to evolve from certain initial conditions.[1] In the early parts of the text we will mostly use the term in

---

[1] The choice to have this clash of terminology here may be somewhat surprising, but the thing is that the term is very fitting for both notions used (both for ours and Valiant’s, and for the standard one in biology). The issue is the particular nature of the notion ‘to evolve’. Let us first focus on two other notions, on ‘to change’ and ‘to be created’. A certain thing that does not exist may not change, while it is for a thing that does not exist that we care whether it can be created. On the other hand, when it comes to our notion

connection to the informal definition and we will allow the context to clarify those occasions where a technical one is used.

Lastly, we will use the term *evolutionary system*—or sometimes just *system*—to refer to the collection, at some point in time, of all those elements that are part of or affect the evolutionary process, and are of interest to us. Its technical sense will be context-dependent, and it will be determined by the model in question. For example, the system at some time t may refer to some environment at time t, the organisms in that environment, and the properties of the accompanying evolutionary process.

Remark 2.1.1. Our understanding of natural selection may be slightly different than the technical understanding of it in the context of biology. We understand natural selection as a mechanism which spans the whole taxonomic hierarchy of organisms, and does not reside on the lowest level alone, i.e., that of the species. As such, in the context of our work, natural selection is a mechanism that also underlies the interspecies dynamics of evolution, e.g., in our understanding of natural selection, it is the latter that is at work when gazelles and crocodiles clash as a result of both needing access to the same water supply.

You may notice, later on, that this abstraction we make is so natural that it is hardly noticeable if not made explicit, especially for non-biologists whose understanding of natural selection tends to be a direct extrapolation from the fact that competition manifests itself between the members of any group of living organisms who strive for the same resources.[2]

The most widely accepted model for macroevolution in biology, and the only one that can be illustrated with detailed fossil evidence, is the extrapolative model which suggests that "macroevolution proceeds by the same process [as microevolution], that of natural selection and adaptive improvement, as has been observed within species and at speciation; but the process is operating over a much longer period" [Rid04, p. 586].

The details of this process are not yet fully understood. However, these details will not affect our work here. Our understanding of natural selection will be in line with the extrapolative model and possibly will give the freedom for our framework to encapsulate even more intricate theories (if anyone ever wants to consider such a thing).

---

of interest, a certain thing that does exist may have the ability to evolve (in analogy with 'to change'), but also a certain thing that does not exist may have the ability to evolve (in analogy with 'to be created'). This may clarify a bit why the same term may be so fitting for two different notions.

[2] More information on interspecies competition may be found in [Con83].

2.2 Life, Environment, and Evolutionary Mechanism

We start here by formalizing a few ideas that rely on very basic assumptions that are hardly questioned in the context of biology. This will serve as a starting point for our framework in Section 5, it will allow us to better understand Valiant's model, and it will also allow us to compare our framework to that of Valiant's.

Understanding an evolutionary system as an ever-changing system of interrelated components, we differentiate between those components whose evolution we are interested in (we save the symbol L for those; 'L' for 'life'), those components that correspond to the particulars of the evolutionary process (we save the symbol M for those; 'M' for (evolutionary) 'mechanism'), and the rest of the components (we save the symbol E for those; 'E' for 'environment'). For example, in the context of life on earth, L may correspond to all living organisms, E to the physical laws, the state of the earth in terms of its geology and atmosphere, etc., and M to an evolutionary mechanism in accordance with neo-Darwinism.

More precisely, we use L, M, and E to denote representations of the respective components. We also use $S = (L, E, M)$ to denote a state of the system at a given point in time. For example, in a formal framework, L could denote a set of organisms represented by their DNA sequence, and M could denote an evolutionary algorithm.

Moreover, we talk of the classes $\mathcal{L}$, $\mathcal{E}$, and $\mathcal{M}$ that contain all the possible L's, E's, and M's, respectively. For example, say that any specific L has the form of a set of entities, where an entity is represented by a binary string of length n, and as such, $L \subseteq \mathcal{P}(\{0,1\}^n)$, where for some set X, $\mathcal{P}(X)$ denotes the power set of X, i.e., the set of all subsets of X. Then if we allow for any such string to code for an entity, $\mathcal{L}$ will be equal to $\mathcal{P}(\mathcal{P}(\{0,1\}^n))$.

Lastly, note that in general the state $S = (L, E, M)$ of a system is dynamic, that is, it changes over time. In principle all of L, M, and E, may be dynamic. However, in a formal framework it is only L that *must* be dynamic—since we are interested in the evolution of the respective entities, while E and M may be static. For example, in analogy with evolution on earth it makes sense to consider M to be static.[3]

---

[3] For further discussion on this latter assumption the reader is referred to Appendix A.2.1.

Chapter 3

Valiant's Framework of Evolvability

Valiant [Val09] suggests that evolution ultimately is a constrained form of learning. He tries to encapsulate the process of evolution in a setting similar to that of computational learning, and, specifically, in the framework of PAC learning [Val84].[1] The main notion discussed in PAC learning is that of *learnability.* According to the theory, some specific concept is learnable if there exists an algorithm that a computer can use so that it *learns to* (as in, reach a state where it is able to) differentiate between whether some arbitrary example is in line with the concept in question or not.

Similar to the above, in Valiant's evolvability framework an entity is evolvable if there exists an algorithm that, starting from any entity, will converge to the entity in question; with the only catch that the algorithm must be efficient and should work with good probability (similar to the PAC learning framework[2] [KV94] in computational learning, defined in the following section).

The evolutionary process is modelled as a process in which, at each step, the current entity is replaced by a fitter entity—or sometimes by one of similar fitness, the latter being selected from a small pool of mutations of the former. The fitness of an entity is determined relative to some target entity. The target entity serves to lead the evolutionary process towards some "ideal" entity, with some kind of "ideal" behavior.

As such, the starting point in the process is the consideration of a certain ideal function that will lead the evolutionary process. Precisely speaking,

---

[1] The PAC learning framework was proposed in 1984 by Valiant himself, and it is one of the reasons why he was awarded the Turing Award in 2010.

[2] PAC stands for *probably approximately correct.* The corresponding framework allows for the use of algorithms that can learn a concept *approximately* with good *probability.*

we start by considering a class of ideal functions, the functions therein are *independently* considered as targets for independent runs of the evolutionary process. This point will be made clear when we reach the definition of evolvability (Definition 3.1.5).

3.1 Formal Definition

In this section we formally define the framework. We precede the definition of evolvability with that of PAC learning. This will help see how Valiant's framework of evolvability is inspired by (his own) PAC learning framework. The exposition below follows [FV08] closely.

3.1.1 Preliminaries

Let $X = \cup_{n\geq 1} X_n$ be the situation space, where $X_n = \{0,1\}^n$, corresponding to the set of possible situations an entity might find itself in. Entities are modelled using functions from X_n to $\{+1,-1\}$, for some n, where $\{+1,-1\}$ is the set of possible ways an entity may respond to a given situation; in other words, what is modelled is the behavior of an entity.

Let D_n be a distribution over X_n that defines the probabilities of the various possible situations $x \in X_n$. Let $D = \cup_{n\geq 1} D_n$ be an ensemble of distributions.

A *concept/entity class*[3] C over X is a set of $\{+1,-1\}$-valued functions over X.[4] We have that $C = \cup_{n\geq 1} C_n$ where C_n is the restriction of C to the concepts/entities over X_n.

Lastly, we need a way to represent concepts/entities in our formal system. For this reason we use the representation class R that contains the representations of all possible entities that we wish to consider in our system. For example, while C may correspond to some phenotypic characteristic of the entities, R may correspond to their DNA.

3.1.2 PAC Learning

Valiant introduced the PAC (Probably Approximately Correct) learning model in [Val84]. Let C be a concept class over X. Let D be some unknown distribution over X. The objective is to find an algorithm $\mathcal{A}$ that can learn C

---

[3] In the context of learning we are interested in *concepts* to be learned, while in the context of evolvability we are interested in *entities* to evolve.

[4] This definition of an entity class is adequate for the needs in Valiant's framework of evolvability. However, in our own framework later we do not restrict the form of an entity class.

under any D, that is, for every concept $f \in C$, and every distribution D, $\mathcal{A}$ can, with good probability, learn f approximately.

The only access that $\mathcal{A}$ has to f is through an example oracle that returns tuples of the form $\langle x, f(x) \rangle$ with x randomly chosen according to D. In other words, the said tuples are random examples together with the value of f on those examples.

Let R be the hypothesis space, i.e., a class of representations of concepts, that $\mathcal{A}$ is using. What $\mathcal{A}$ does is use hypotheses $r \in R$ in order approximate f. As such, the objective is to find a good $r \in R$ that has approximately the same behavior as f.

Definition 3.1.1. An algorithm $\mathcal{A}$ is said to *PAC learn* C in time t if for every n, $\epsilon \in (0, 1)$, $\delta \in (0, 1)$, $f \in C_n$, and distribution D_n over X_n, $\mathcal{A}$ given ϵ, δ, and access to the example oracle, outputs in time t with probability at least $1 - \delta$, a hypothesis r such that $\mathrm{Pr}_D[f(x) \neq r(x)] \leq \epsilon$.

The learning is called *efficient* if t is upper bounded by some polynomial $p(n, 1/\epsilon, 1/\delta)$. We say C is *learnable* if there exists an algorithm $\mathcal{A}$ that can PAC learn C.[5]

3.1.3 Evolvability

Let C be an entity class. We understand C as the class of *ideal* functions/entities, i.e., the functions that will be used as targets for the evolutionary process.[6] Let $R = \cup_{n \geq 1} R_n$ be a class of representations of functions, where R_n is the restriction of R to functions of n variables. We assume functions and representations are polynomially evaluable, i.e., $f \in C$ and $r \in R$ can be efficiently computed for every $x \in X$.

Similarly to the setting of PAC learning, on each run of an algorithm $\mathcal{A}$ in the current setting, $\mathcal{A}$ tries to converge to some specific function in C. In analogy with evolution, the idea here is that $\mathcal{A}$ is run with some specific $f \in C$ as the ideal function, i.e., the ideal entity towards which all evolution drifts. Having specified that f is the ideal function, we understand that f gives the "best" possible response for each situation $x \in X$. Subsequently, the closer to f the behavior of a representation (of a function) $r \in R$ is, the better the performance of r.[7]

---

[5] In this text when we say 'learnable' we mean 'PAC learnable'.

[6] For each run of the evolutionary process, we will assume that there exists only one such ideal behavior at any point in time, and, as such, we will not require convergence to more than one ideal function at the same time. Every time the algorithm runs, it tries to converge towards some specific ideal function.

[7] Clearly, the performance we are talking about is relative to f, and it may look strange

The idea here is that the algorithm starts from some random representation r_0 which "mutates" to some representation r_1 and so on. At each point in the algorithm we only remember the last representation. If the latter is close in performance to f, the algorithm stops.

Definition 3.1.2. We define the *performance* of a representation r relative to some ideal function f over D to be the correlation of r with f taken over all $x \in X$ weighted according to D. Precisely speaking, we have

$$\mathrm{Perf}_f(r, D) = \mathbf{E}_{x\in D}[r(x) \cdot f(x)],$$

where $\mathbf{E}_{y\in P}[\phi(y)]$ is the expected value of $\phi(y)$ for y chosen according to probability distribution P, and $\cdot$ denotes multiplication.

Moreover, we define the *empirical performance* $\mathrm{Perf}_f(r, D, s)$ which is computed using, instead of all vectors in X, just s random samples from X.[8] Precisely speaking, we have the following random variable

$$\mathrm{Perf}_f(r, D, s) = \frac{1}{s}\sum_{i\le s} r(z_i) \cdot f(z_i),$$

where $z_1, z_2, \ldots, z_s \in X$ are random variables chosen randomly and independently according to D.

As we already mentioned, we want our algorithm to converge to f. Specifically, we want an algorithm that with probability $1 - \delta$ will reach some representation r such that $\mathrm{Perf}_f(r, D) \ge 1 - \epsilon$ for some small $\epsilon > 0$.

We define now a quadruple that holds a vector of parameters for the mutator function Mu (to be defined).

Definition 3.1.3. Let $P = (R, \mathrm{Neigh}, \mu, t)$ be a *parameter vector*[9] where

- R is the set of representations of functions over X that the algorithm may use,

---

to talk about it as if it was absolute performance and still have a whole set, C, of ideal functions. This is a remnant from the setting of computational learning, and later on we will see how the notion of the ideal function is somewhat unnatural for the modelling of evolution.

[8] $\mathrm{Perf}_f(r, D)$ depends on all the situations $x \in X$, which are 2^n in number. However, $\mathrm{Perf}_f(r, D, s)$ depends only on s, in number, of the situations in X, and hence will be efficiently computable if s is polynomially bounded.

[9] The term 'evolutionary algorithm' is used in [FV08] instead of 'parameter vector', but we opt for the latter because what is called an 'evolutionary algorithm' in [FV08] is just a set of parameters for the evolutionary algorithm and not a complete characterization of the algorithm.

- Neigh is a function such that for $r \in R_n$, $\epsilon \in (0,1)$, and $\delta \in (0,1)$, $\mathrm{Neigh}(r,\epsilon,\delta)$ is the neighborhood of r, i.e., the set of representations into which r may "mutate". It holds that $r \in \mathrm{Neigh}(r,\epsilon,\delta)$ and $|\mathrm{Neigh}(r,\epsilon,\delta)| \leq p_P(n, 1/\epsilon, 1/\delta)$ for a fixed polynomial p_P,
- μ is the function such that for $r \in R_n$ and $r_1 \in \mathrm{Neigh}(r,\epsilon,\delta)$, $\mu(r, r_1, \epsilon, \delta)$ is the probability that r "mutates" into r_1,
- $t(r,\epsilon,\delta)$ is the tolerance at $r \in R_n$, that is, the minimum increase in performance that a "mutation" has to attain in order to be considered beneficial. The tolerance is bounded from above and below by a pair of polynomially-related polynomials in the following sense. There exist a polynomial $\mathrm{tu}_N(1/n,\epsilon,\delta)$ and a constant $\eta \geq 1$ such that for all $r \in R_n$ and $\epsilon > 0$, $\mathrm{tu}_N(1/n,\epsilon,\delta) \geq t(r,\epsilon,\delta) \geq \mathrm{tu}_N^{\eta}(1/n,\epsilon,\delta)$.

Functions Neigh, μ, and t need to be computable by a randomized algorithm in time polynomial in n and $1/\epsilon$.

Let us see now how "mutation"—corresponding to one step in the evolutionary process—works.

Definition 3.1.4. For a function f, distribution D, parameter vector $P = (R, \mathrm{Neigh}, \mu, t)$, a representation $r \in R$, accuracy ϵ, probability of failure δ, and sample size s, the *mutator* $\mathrm{Mu}(f, D, m, r, \epsilon, \delta, s)$ is a random variable that takes value r_1 as follows. First the mutator computes an empirical value of $v(r') = \mathrm{Perf}_f(r', D, s)$ for each $r' \in \mathrm{Neigh}(r,\epsilon,\delta)$.[10] Let

$$\mathrm{Bene} = \{r' \mid v(r') \geq v(r) + t(r,\epsilon,\delta)\}$$

and

$$\mathrm{Neut} = \{r' \mid |v(r') - v(r)| < t(r,\epsilon,\delta)\}.$$

Then, in case that Bene is not empty, output some $r_1 \in \mathrm{Bene}$ with probability $\mu(r, r_1, \epsilon, \delta) / \sum_{r' \in \mathrm{Bene}} \mu(r, r', \epsilon, \delta)$. Otherwise, i.e., $\mathrm{Bene} = \emptyset$, output $r_1 \in \mathrm{Neut}$ with probability $\mu(r, r_1, \epsilon, \delta) / \sum_{r' \in \mathrm{Neut}} \mu(r, r', \epsilon, \delta)$.

In other words, the mutator starts by categorizing the representations in Neigh, which comprise the possible mutations, between beneficial and neutral mutations, or nothing of the two. It does this in terms of the performance of those possible mutations, determined through the use of a sample, of X, of size s. If there is at least one possible beneficial mutation, it selects between

---

[10] Note that $\mathrm{Perf}_f(r, D, s)$ only takes as an argument the size of the sample, and not the actual sample. As such, the samples $z_1, z_2, \ldots, z_s \in X$ are chosen independently for the various r.

all the possible beneficial ones according to μ. Otherwise, it uses μ to decide among the mutations in Neut.[11] This process gives rise to a sequence of representations $r_0, r_1, r_2, \ldots$ such that $r_i = \mathrm{Mu}(f, D, P, r_{i-1}, \epsilon, \delta, s)$.

We are ready now to define the evolvability of a class of functions.

Definition 3.1.5. Let C be a class of functions, D a distribution over X, and $P = (R, \mathrm{Neigh}, \mu, t)$ a parameter vector. We say that C is *evolvable* by P over D if there exist polynomials $s(n, 1/\epsilon, 1/\delta)$ and $g(n, 1/\epsilon, 1/\delta)$ such that for every n, $f \in C_n$, $\epsilon \in (0, 1)$, $\delta \in (0, 1)$, and $r_0 \in R_n$, with probability at least $1 - \delta$, the sequence $r_0, r_1, r_2, \ldots$, where $r_i = \mathrm{Mu}(f, D, P, r_{i-1}, \epsilon, \delta, s(n, 1/\epsilon, 1/\delta))$, will have $\mathrm{Perf}_f(r_{g(n,1/\epsilon,1/\delta)}, D) > 1 - \epsilon$.

The polynomial $g(n, 1/\epsilon, 1/\delta)$ is used to bound the number of generations needed for the evolution of f. Note that in the above definition evolvability is distribution-dependent. The respective distribution-independent definition follows.

Definition 3.1.6. A class of functions C is evolvable if there exists a parameter vector P such that for all distributions D, C is evolvable by P over D (using the same polynomials s and g).

Remark 3.1.1. Let us take a moment to note the parameters to the evolutionary algorithm/process that we defined above and that are implicit in the definition of evolvability. First, there are the "global" parameters X and D (understood as the environment where the entities reside), and P (understood as the characterization of the evolutionary mechanism). There is the "local" parameter f, which is the ideal function, and lastly, there is the characterization of the initial state of the system, which amounts to r_0, the initial representation.

Lastly, let us give one more definition that will allow us later to handle some notions with more ease.

Definition 3.1.7. A concept class C is evolvable by a representation class R over distribution D if C is evolvable over D using R as a representation class.

Remark 3.1.2. Evolvability 'by R over D' is a notion in between evolvability as defined in Definition 3.1.6 and Definition 3.1.5. It is similar to evolvability 'by a parameter vector $P = (R, \mathrm{Neigh}, \mu, t)$ over D', only that instead of fixing the whole parameter vector P, we fix only its first element R.

---

[11] Note that there is always at least one representation in Neut, namely the current representation r.

Another way to look at it is as follows. Evolvability 'by R' demands C to be evolvable using R specifically, while evolvability as in Definition 3.1.6 only requires the existence of such an R under which C is evolvable.

3.2 Some Results

In this section we give some results concerning Valiant's framework.

3.2.1 Evolvability Versus Learnability

Here we will present some results on the relative power of the two frameworks, that of evolvability and that of learnability.

If a class is evolvable it is also learnable. The definition of evolvability is closely related to that of learnability [KV94]. However, it comes with the following restrictions. Each step of learning chooses from a hypotheses set of polynomial size, tolerates at most a small decrease in performance, and the next hypothesis is chosen on the basis of its average performance on example situations, and not according to their exact behavior on the example situations.

In fact, every evolutionary algorithm in Valiant's framework can be simulated by a PAC learning algorithm. Specifically, the following holds.

Proposition 3.2.1. Let C be a concept class, R a representation class, and D a distribution over the situation space X. If C is evolvable by R over D then C is learnable by R over D. Moreover, if C is evolvable by R then C is learnable by R.

In light of Remark 3.1.2 we get the following corollary of Proposition 3.2.1.

Corollary 3.2.1. If C is evolvable then C is learnable.

The main idea in the proof of Proposition 3.2.1 is that the evolutionary algorithm can be simulated in the PAC learning framework. At each step of the process, the algorithm takes a sample of $s(n, 1/\epsilon, 1/\delta)$ examples from X according to D, computes the empirical performance for the current hypothesis, forms the neighborhood of the latter, and from that neighborhood it chooses the next hypothesis. All those steps are easily implemented in the PAC framework.

Evolvability is equivalent to CSQ-learnability. Valiant [Val09] proved that if a class is evolvable it is also SQ learnable, i.e., learnable in the statistical queries (SQ) framework [Kea98].[12] Feldman [Fel09] took this result further and proved that evolvability is *equivalent* to a natural restriction of statistical queries, referred to as correlational statistical queries (CSQ). In other words, the following holds.

Proposition 3.2.2. C is evolvable iff C is learnable by CSQs.

This latter result is not difficult to see. A CSQ allows a learning algorithm to obtain an estimate of the correlation between the current hypothesis/representation and the target/ideal function. But this is exactly the performance metric used in Valiant's framework of evolvability as seen in Definition 3.1.2.

Evolvability is strictly weaker than learnability. Kearns [Kea98] has found a class that is PAC learnable but not SQ learnable. This, together with Proposition 3.2.2 and the fact that CSQs are weaker than SQs, gives us that evolvability is strictly weaker than learnability.

Lastly, note that since the evolutionary process in the framework is a constrained form of PAC learning, it is more difficult to design algorithms that give rise to the evolution of a certain function than designing PAC learning algorithms to learn that function.

3.2.2 Examples of Evolvable Structures

Here we see a few examples of evolvable structures under Valiant's framework. Firstly, let us consider conjunctions and disjunctions on n variables $x_1, \ldots, x_n$. A conjunction or disjunction is *monotone* if it contains no negated literals. Valiant [Val09] proved the following.

Proposition 3.2.3. Monotone conjunctions and disjunctions are evolvable over the uniform distribution by their natural representations (as lists of literals).[13]

This result can be strengthened if initialization is allowed. Valiant defines *evolvability with initialization* similar to evolvability in Definition 3.1.5, but instead of requiring convergence from any starting point $r_0 \in R$, we require only that there exists some specific starting point r_0 such that we have convergence for all targets $f \in C$. Valiant proved that

---

[12] SQ and CSQ are defined in Appendix A.1.

[13] Note that, in line with Remark 3.1.2, this result is stronger than saying that "monotone conjunctions and disjunctions are evolvable over the uniform distribution".

Proposition 3.2.4. Conjunctions and disjunctions are evolvable with initialization over the uniform distribution.

Moreover, Feldman [Fel08], based on the equivalence of evolvability with CSQs, proved that

Proposition 3.2.5. For any fixed distribution D, if a class of functions is learnable by SQs over D then it is also evolvable over D.

Lastly, Michael in [Mic07] gave an algorithm for evolving decision lists using a different fitness/performance metric over hypotheses.

3.3 Robustness of the Model

Feldman [Fel09] examined the robustness of Valiant's model, by comparing some variants of it in order to find which modelling choices affect the power of the model. The two most important parameters were the performance metric and the selection procedure. In this section we sketch some of his results.

3.3.1 Performance Metrics

In learning theory, a loss function is usually used as a measure of the distance between a hypothesis and an ideal concept. For functions with range Y, a loss function L is a non-negative mapping $L : Y \times Y \to \mathbb{R}^+$.

For a certain loss function L, the L-SQ model, defined by Feldman [Fel09], is roughly a statistical query model that can only find the L-distance between a hypothesis and an ideal function; in contrast to the SQ model that can use any loss function.[14]

Previously, we saw that evolvability is equivalent to learning by CSQs. Feldman proved that this equivalence can be extended as follows.

Proposition 3.3.1. For any admissible loss function L, evolvability using L as its performance metric is equivalent to learning by L-SQs.

Moreover, Feldman demonstrated that the power of the model is greatly affected by the performance metric used. We say that a loss function is *quasi-linear* if for every $y \in [-1, 1]$ it holds $L(1, y) + L(-1, y) = L(-1, 1)$. Feldman proved the following.

Proposition 3.3.2. For L quasi-linear, L-SQ learnability is equivalent to CSQ learnability, while for L not quasi-linear, L-SQ learnability is equivalent to SQ learnability.

---

[14] More information on L-SQs can be found in Sections A.1 and A.1.1.

3.3.2 Selection Rules

At each stage of the process a selection rule is used to select the next hypothesis/representation. Consider the following two selection rules:

(i) the selection rule used in Valiant's model, i.e., form the Bene and Neut sets and select a mutation—randomly according to μ—from Bene if not empty, otherwise select from Neut.

(ii) a selection rule which instead of aiming to select some mutation from Bene, which contains all "beneficial" mutations, tries to select one of the mutations that are close to optimal among the pool of mutations available. Specifically, let Opt be the set $\{r' \in \text{Neigh}(r, \epsilon, \delta) \mid v(r') \geq p^* - t(r, \epsilon, \delta)\}$, where $p^* = \max_{r' \in \text{Neigh}(r,\epsilon,\delta) \bigcup \{r\}} \{v(r')\}$ and Neigh, v, and t are as in Definition 3.1.4. The algorithm selects the mutation from Opt randomly according to μ.

Feldman [Fel09] proved that the power of the model is not affected by which of the two selection rules is used.

3.4 On the Structure of the Definition of Evolvability

In this section we discuss Valiant's definition of evolvability in terms of the components that we are mainly interested in ('life', 'evolutionary mechanism', 'environment'), as they were outlined in Section 2.2. This will allow to better understand how the current framework models evolution, and how it relates to our framework.

Recall that the three important components in Section 2.2 were the characterizations/representations of the entities (corresponding to $\mathcal{L}$), the environment (corresponding to $\mathcal{E}$), and the evolutionary algorithm (corresponding to $\mathcal{M}$). Here we assign to $\mathcal{L}$, $\mathcal{E}$, and $\mathcal{M}$ concrete roles in Valiant's framework. We identify $\mathcal{L}$ with the class of representations R, that is, $\mathcal{L}$ contains all the possible representations of entities r. We identify $\mathcal{E}$ with the set of all possible distributions D, where a distribution is interpreted as an environment. Lastly, we identify $\mathcal{M}$ with the set of all possible parameter vectors P, where a parameter vector is interpreted as an evolutionary algorithm.

Given that X is the set of all possible situations into which an entity may find itself, in Valiant's framework we understand the environment as the distribution D on X; that is the reason why we use $\mathcal{E}$ as mentioned. Moreover, a parameter vector P completely determines the evolutionary algorithm; that is why we use $\mathcal{M}$ as mentioned.

From Definitions 3.1.5 and 3.1.6, we have the following for any concept class C:

$$\begin{aligned}
&C \text{ is evolvable } \Leftrightarrow \\
&\quad \exists P \in \mathcal{M}, \forall D \in \mathcal{E}, \exists \text{ polynomials } s(n, 1/\epsilon, 1/\delta), g(n, 1/\epsilon, 1/\delta), \\
&\quad \forall n, \forall f \in C_n, \forall r_0 \in \mathcal{L} \\
&\qquad \text{s.t. } \forall \epsilon, \forall \delta, \text{ with probability at least } 1-\delta, \text{ the sequence } r_0, r_1, \ldots, \\
&\qquad\quad \text{where } r_i = \mathrm{Mu}(f, D, P, r_{i-1}, \epsilon, \delta, s(n, 1/\epsilon, 1/\delta)), \\
&\qquad\quad \text{will have } \mathrm{Perf}_f(r_{g(n,1/\epsilon,1/\delta)}, D) > 1 - \epsilon. \qquad (3.1)
\end{aligned}$$

Note that in Valiant's framework the environment and the evolutionary algorithm are static, while the only dynamic part is the current entity r_i.

One thing we may notice from (3.1) is that Valiant, in order for a class of functions C to be evolvable, demands the existence of an evolutionary mechanism P that results in all the functions in C being able to evolve in all possible environments D, starting from any possible initial entity r_0. More about this in Section 4.3.

Chapter 4

Shortcomings of Valiant's Framework

In this chapter we attempt to point out various shortcomings of Valiant's framework. They constitute mostly discrepancies with standard facts of biology or clash with the intuition we get from evolution as it happened on earth. Later in the text, it will be rather evident that those shortcomings are mainly the result of Valiant's decision to fit his framework of evolvability into that of PAC learning.

The main reason we analyse the framework in this way is in order to better understand the issues at stake. As such, this analysis serves as a starting point for our own framework.

Each choice/assumption made in the framework may give rise to many shortcomings. The following subsections are categorized according to the assumptions made in the framework, rather than according to the shortcomings induced by those assumptions. Sometimes, for example, two different assumptions may force the framework to the same discrepancy, but we still analyse those two assumptions separately.

4.1 One Representation Assumption

Let us denote by ORA the One Representation Assumption of Valiant, which stands for the fact that in his framework he only considers the setting where only one type of entity exists at any given point in time.[1]

---

[1] Recall that he considers successive representations r_i that the evolutionary mechanism tries to push towards some ideal function. Note that the Neigh function allows the mechanism to also consider some neighbours of the current entity, that is, roughly, a limited set of entities that are used to model entities in the same species with the current entity.

Coevolution. It directly follows from ORA that the model of the evolutionary process prescribed in the framework does not consider the *co*existence of many entities of different type.

In order for Valiant to handle the issue that the model of the process cannot account for the possible coevolution of many entities of different type, he needs to consider many ideal functions, each of which acts as a target towards which the evolutionary mechanism must be able to evolve, completely ignoring all other ideal functions. This gives us no guarantee that those ideal functions could have evolved all together in the same system. As such, the framework cannot account for a multitude of entities concurrently evolving in the same system.

Natural Selection. Natural selection forces the evolution of the entities of some environment, at a certain moment in time, to drift towards the fittest of them, while taking into account the interrelationships between all the entities.[2] In other words, natural selection, even if it to a certain extent acts locally, determines the fitness (i.e., the chances of an individual to reproduce) of some individual, relative to the other individuals in the environment.

Given ORA, though, we see that the current framework only allows us to determine fitness in an absolute way (in the sense that the fitness of an individual is always the same regardless of the state of the other entities). This is another reason why Valiant's framework cannot account for macroevolution since it does not consider interspecies relationships.

4.2 Static Environment

Valiant assumes that the environment is static. This is a serious restriction since it does not allow for the coevolution of the entities and the environment they reside in. In the context of microevolution this may be acceptable, but in the context of macroevolution we cannot expect the environment to stay unchanged over time. In particular, it may be the case that the change of the environment is a critical factor in driving evolution in the large scale. As such, a framework that cannot handle this change may fail to predict the specifics of evolution.

Note that the assumption of a static environment in conjunction with ORA allows Valiant to use ideal functions, since the corresponding entities are ideal in that environment. If we were to drop the static environment assumption then clearly a changing environment could give rise to different

---

[2] Since all the entities rely on the same resources for survival.

ideal behaviors, which means that there would be no a priori ideal functions. While if we were to drop ORA then the fact that an entity in the system would not be independent from the rest of the entities, means that we would not be able to pinpoint independent ideal behaviors.

4.3 The Notion of 'Evolvable'

Another questionable point is Valiant's modelling of 'evolvable'. As we can see from (3.1), for a function f to be evolvable it needs to be the case that there exists an evolutionary mechanism P through which f may evolve under any environment D, starting from any initial state r_0. This seems to be a definition for something that is bound to evolve, regardless of environment and initial conditions, that is, regardless of chance. This definition stems from the setting of computational learning, where the objective is to find a mechanism such that the concept will be learned no matter what.

However, one would expect some entity to be 'evolvable' if it can arise with some fair probability[3], corresponding to being able to evolve under some settings, instead of under all settings. The latter is a less strict definition that would allow for more intricate analysis.

Another counterargument for the above definition of evolvability goes as follows. A widely accepted thesis, for example, is that animals have evolved from unicellular organisms, but it may not be the process of evolution that gave rise to the latter. As such, in analogy to evolution on earth, we cannot require of an organism to be able to evolve from any possible initial state in order to call it evolvable. For in that case we would not be able to call animals evolvable, since in the above example animals would not have evolved if starting from some initial state where no unicellular organisms were around, except only if we took into account some other process (other than the evolutionary one) that could give rise to unicellular organisms. Such a process, however, would be out of the scope of this framework.

All in all, what we may safely argue is only that the notion of evolvability is context-dependent and Valiant's definition corresponds to only a special case of it. More on the context-dependent nature of the notion of evolvability can be found in Appendix A.2.2.

---

[3] In the case of Valiant's framework specifically, the random parameters on which we may quantify the said probability are the initial representation r_0 and the distribution D.

4.4 Distant Targets

In Valiant's framework, on every run of the evolutionary algorithm there exists exactly one ideal function f. Evolution is modelled as starting from a certain initial entity r_0 and evolving towards f, while every entity r_i, in between r_0 and f, has f as its target. Let's keep this in mind.

In macroevolution we are interested in the evolution of organisms starting from simple unicellular ones and reaching complex modern-day animals. In such a setting Valiant's framework suggests that the unicellular organism (corresponding to its initial entity r_0) has as its target ideal function a modern-day animal, such as the human. However, it makes no sense to assess the performance of a unicellular organism on the situations $x \in X$ that are important to a human. Adaptational forces that act on a unicellular organism are very different from those that act on a human, as those organisms have largely different needs to satisfy.

Moreover, we would expect the target ideal function of the unicellular organism to be a lot simpler than that of the human, and as such it would be a lot easier for the unicellular organism to evolve to that intermediate target function. In the same vein, it may be also a lot simpler to evolve to many intermediate targets that are successively closer to a human than to evolve towards a human while having the human as a target right from the start.

Lastly, note that Valiant is explicit about the phenomenon we discuss here. He writes: "the theory, analogously to learning theory, analyses only the *granularity* of the structure that can evolve in a single phase with a single ideal function" [Val09, pp. 15]. He adds: "[i]f multiple phases are allowed with different ideal functions in succession, then arbitrarily complex structures can evolve. For example, in response to various initial ideal functions some set of conjunctions and disjunctions may evolve first. At the next phase the outputs of these functions can be treated as additional basic variables, and a second layer of functionality can evolve on top of these in response to other ideal functions. This process can proceed for any number of phases, and build up circuits of arbitrary complexity, as long as each layer is on its own beneficial. We call this *evolvable target pursuit.*"

It is even more interesting to note what Valiant writes in the Abstract of [Val09]: "[w]e suggest that the mechanism that underlies biological evolution overall is 'evolvable target pursuit', which consists of a series of evolutionary stages, each one inexorably pursuing an evolvable target in our technical sense, each such target being rendered evolvable by the serendipitous combination of the environment and the outcomes of previous evolutionary stages." However, the two quotes above are the only places where Valiant mentions 'evolvable target pursuit'. As such, at this point his framework seems to overlook some macroevolution.

4.5 Monotonicity

An entity being able to evolve to some target does not necessarily mean that it can evolve to it monotonically, in terms of performance. In his model Valiant allows for a mutation to be selected only from the beneficial or neutral mutations. Hence, he effectively bounds the decrease in performance, for any given mutation, by the tolerance $t(r, \epsilon, \delta)$.

A disadvantageous mutation may arise recurrently [Rid04]. Natural selection may not be able to eliminate such a mutation. Given also exaptation, where that certain trait that resulted from the mutation becomes later a beneficial trait, it is not necessarily a good idea to ignore non-monotonicity of performance.[4]

Let us note here, though, that we have not analyzed the issue in detail and it may be the case that Valiant's restriction does not hinder his framework in the way hinted above. The reason being that the small decrease in performance allowed by his framework may be sufficient to model any other decrease in performance; at least to the extent that evolvability is concerned.

4.6 Conclusion

Valiant tried to model evolution staying close to the setting of computational learning. We find this resulted in some unnatural assumptions. Besides the fact that he does not handle evolution of the environment, which would probably be a very difficult task anyway, his notion of 'evolvable', the fact that he uses distant targets, and especially the fact that he does not model coevolution we find to be major restrictions.

Aside from the static environment, Valiant's most important assumption is ORA. Under the latter, the rest of his modelling choices are fairly reasonable. For example, the choice to use ideal functions in order to navigate the course of the representations, i.e., each new mutation tries to approach the ideal function. However, this choice feels fairly unnatural since we do not understand in general the evolutionary process as having some distant innovation as a target, but more like as if it tries to adapt to the issues at hand, i.e., the current problems that the organisms face at each point in time.

---

[4] Note that these phenomena may be too rare, and as such the argument about monotonicity weakens. But if we are to be precise we should probably just model things in such a way that is in line with decrease in performance having a low probability.

Another unnatural assumption that makes sense under ORA is that, without loss of generality, all the resources evolution has in the current framework is one processor, instead of something more in the lines of a massively parallel computer as one would expect.

All in all, the fact that the framework resides in the setting of computational learning resulted in getting many results about evolvability as defined. However, we feel that there is a need for a more precise modelling of the evolutionary process, one that stays closer to our intuitions about evolution.

Chapter 5

Definition of our Framework

In this chapter we define our framework. Our object of study is evolution that employs mutation and some kind of natural selection; in line with the theory of neo-Darwinism. This determines to a certain extent the details of our framework. However, it still leaves a large degree of freedom. We are interested in formulating a framework that is abstract enough to allow for the analysis of most of the interesting cases of evolutionary systems that fit the picture as described above.

The idea is that the user of the framework should be able to specialize it to her needs. The main property that is analyzed in the framework is the evolvability of entities. However, there are many ways in which one may understand evolvability. As such, we need to give the user a certain amount of freedom in choosing what she considers evolvability to be. We discuss the issue further in Section 6.2.

However, before deciding on a working definition of evolvability, the user needs to specify the details of the evolutionary process[1]. To that end we define the Meta-Model (MM) in Section 5.1.1 which describes the basics of the evolutionary process. The user is supposed to specialize MM in order to get to a model of the evolutionary process that satisfactorily captures the application in mind.

In the next chapter we discuss further how the user may specialize MM so that it satisfactorily models her specific evolutionary system of interest. Moreover, we discuss how the user may design a definition of evolvability that fits her needs and we give some interesting examples of such designs.

---

[1] In this chapter we use the terms ‘evolution’ and ‘evolutionary process’ interchangeably.

5.1 Modelling Evolution

In this section we introduce the Meta-Model (MM), which is a high-level meta-model that describes evolution. MM determines only the essential parts of an evolutionary system. Therein one can find our basic assumptions. This is the basis of our framework and the user may specialize MM in order to get to a model that better fits her needs.

Throughout the framework we use the symbols S, L, E, and M in line with Section 2.2. As such, S denotes the state (L, E, M) of an evolutionary system at a certain point in time. However, L, E, and M are still open to a concrete definition. Roughly, the only thing that holds true for L, E, and M throughout the framework is that their concrete definitions are to correspond to modellings of the 'life', 'environment', and 'evolutionary mechanism' components of the system of interest.[2] In general we would like to allow for the possibility for L, E, and M to range along many different objects. Hence, we will opt to define the classes $\mathcal{L}$, $\mathcal{E}$, and $\mathcal{M}$ that contain all the possible L's, E's, and M's, respectively. This point will be made clear in the discussion about the definition of evolvability later.

We accompany the description of a couple elements of MM with descriptions in pseudocode that briefly encapsulate the processes in question. The pseudocode is quite simple to understand and should be quite clear in conjunction with the accompanying verbal descriptions. A few notes on the format of our pseudocode can be found in Appendix A.3.

5.1.1 Meta-Model

The Meta-Model (MM) defines certain details about the form of the components of an evolutionary system L, E, and M. Specifically, L is a multiset that contains representations of entities. The possible representations are elements of a representation class R which is not yet defined. E and M are also not yet defined, except for the fact that M contains the functions b, d, Mu, and N, which correspond to the probability that an entity reproduces ('b' for birth), the probability that an entity dies ('d' for death), the mutator function, and the function that returns the multiset of the "significant" neighbors of an entity ('N' for neighborhood), respectively. All those undefined details will be left to the user who is supposed to define them so that she gets a model of evolution appropriate to her needs.

Time is discretized into timeslots t. Let S^t be the state of the system during the t-th timeslot, with S^0 being the initial state. For $S = (L, E, M)$ a

---

[2] In addition we restrict L to be a multiset.

state of the system, we denote by $S.L$ the first component of the tuple S. We use similarly $S.E$ and $S.M$ for the other two components of S. Moreover, we use the same convention to point to the components of L, E, and M themselves, e.g., $S.M.Mu$ denotes the mutator function of $S.M$.

Some burdensome notation, such as the above, is unavoidable. The reason is that we allow for the possibility for all the components of a system to be dynamic (i.e., varying with time), and as such, we need some way to specify exact components at various points throughout the process. We hope that the reader will appreciate this notation when definitions start getting somewhat complex. This notation carries some redundancy but demands less of the reader in terms of the depth of clarity in which she has to remember what the symbols stand for.

A Step in the Process

The Meta-Model defines functions **step** and **stepL**, which correspond to a step in the evolution of the system and a step in the evolution of the life parameter L of the system, respectively. We have $S^{t+1} = \mathbf{step}(S^t)$. Functions **step** and **stepL** are defined as follows.

Function **step** is quite simple. It takes a state S as an input and outputs a new state formed by the three components of S updated by the functions **stepL**, **stepE**, and **stepM**, respectively. The latter two will not be defined in MM, only **stepL** will be. The corresponding pseudocode follows.

```
IN: state. OUT: state.
step($S$)
  $L' \leftarrow$ stepL($S$)
  $E' \leftarrow$ stepE($S$)
  $M' \leftarrow$ stepM($S$)
  return ($L'$,$E'$,$M'$)
```

Function **stepL** is a bit more complicated. Before starting with its description, note that the pseudocode below uses the random function flip, where flip(x) is a random variable that takes the value TRUE with probability x, and FALSE with probability $1 - x$.

Function **stepL** takes as input a state S and updates its life component $S.L$ to give rise to a new life component L'. L' is the same as L except for the entities that die and those new entities that are the result of reproduction.

Each entity may reproduce at most once in a given timeslot; the probability of reproduction of an entity in $l \in L$ is given by applying the function $S.M.b$ to l. Note that $S.M.b$, as with all the other components of $S.M$, is not defined by MM; the only part defined about the functions that are components of $S.M$ is the type of their arguments and of their output. The probability that an entity dies is given by applying $S.M.d$.

One way to describe the mapping that results from **stepL** is the following. L' is initialized to $S.L$. Then, for each $l \in S.L$, **stepL** forms the neighborhood neigh of l using the function $S.M.N$, and based on l, neigh, and the environment $S.E$, it computes (probabilistically—using $S.M.b$, $S.M.d$, and flip) whether l is to reproduce and/or to die. If l is to reproduce, L' gains a new element $S.M.Mu(l, S)$, which is a mutation of l. If l is to die, l is removed from L'.

```
IN: state. OUT: life.
stepL($S$)
  $L' \leftarrow S.L$
  for each $l \in S.L$
    neigh $\leftarrow S.M.N(l, S)$
    if flip($S.M.b(l, \text{neigh}, S.E)$) then $L' \leftarrow L' \cup \{S.M.Mu(l, S)\}$
    if flip($S.M.d(l, \text{neigh}, S.E)$) then $L' \leftarrow L' \setminus \{l\}$
  end
  return $L'$
```

The pseudocode above only defines the mapping that results from **stepL** and not the actual way in which the process happens. For example, if we were to understand the process as a computational one, we would have that if the computation for each $l \in S.L$ was done in parallel, then **stepL**(S) could still be computed in finite time even if $S.L$ was infinite.

MM also defines the form of functions **stepE** and **stepM**. Those two functions are similar to **stepL**. They take a state S as input and return the updated $S.E$ and $S.M$, respectively, as may be seen in the following abstract pseudocode.

```
IN: state. OUT: environment.
stepE(S)
  ...
```

```
IN: state. OUT: evolutionary mechanism.
stepM(S)
  ...
```

5.1.2 Remarks on MM

In this section we present various remarks on MM considering various possibilities, potential limitations, and implicit assumptions.

Limits on the size of L. Notice that the size of L can grow in ways that are perhaps unnatural. Consider, for example, the case where $S.M.b$ always returns 1 and $S.M.d$ always returns 0. In that case, $|L|$ would double on each timeslot. However, there are more interesting modelling possibilities. In analogy with a natural system we would expect that a population can only grow to the extent that the resources of its environment allow. As such, a more natural way to model evolution may be to have a negative correlation between the probability that an entity reproduces $S.M.b$ and $|L|$, and a positive correlation between $S.M.d$ and $|L|$.

Sexual reproduction. It is a basic assumption of our model that a new entity is mainly based on *one* already existing entity (evident in the function call $S.M.Mu(l, S)$). This means, for example, that there is no sexual reproduction. Nevertheless, this assumption can be overridden since $S.M.Mu$ takes the whole of S as an argument and we allow for S to store any kind of information. Note that this has not been our intention initially and it is only a side effect of our design.

One birth per entity per timeslot. Each already existing entity can reproduce at most once in a given timeslot. However, the timeslot may correspond to a very small interval in "real" time and as such we still allow for many reproductions in a short length of time.

Fitness correlation between different entities. An important aspect of MM is that it allows for the reproduction/birth and the death probabilities of a certain entity to be affected by the presence of other entities. That is the main reason for the existence of the neighborhood function $S.M.N$, whose output is fed to $S.M.b$ and $S.M.d$. The main idea behind this choice is that in analogy to evolution on earth, entities who live in the same environment need to share common resources.

Note, however, that sometimes there is no such idea of common resources shared by entities. In case, for example, of the evolution of monotone conjunctions there is no natural environment into which we may consider those abstract objects/entities to reside.

Moreover, note that the other factor that affects the $S.M.b$ and $S.M.d$ probabilities, which essentially determine the fitness of the various entities, is the environment $S.E$. This is similar to what Valiant does if we interpret the distribution D over the situation space X as the environment of the entities. However, Valiant does not consider the coevolution of entities, so in his model there is no issue of the fitness of different entities being correlated or not.

Chapter 6

Specializations of our Framework

In this chapter we discuss specializations of our framework. This is the first step towards exploring specific applications. In Section 6.1 we discuss how the user may specialize the Meta-Model (MM) in order to reach a model of evolution. We also give an example of an interesting but quite high-level specialization. In Section 6.2 we discuss how the user may design a definition for evolvability and we also give some interesting examples. Lastly, in Section 6.3 we demonstrate how Valiant's framework is a special case of our framework.

In the next chapter we will make all these more concrete by examining some applications of our framework.

6.1 Specializations of the Meta-Model

For a complete modelling of evolution, as far as MM is concerned, one thing we need to do is specify completely functions **stepE**, and **stepM**. Moreover, we need to specify the exact form of L, E, and M, which essentially means that we need to specify $\mathcal{L}$, $\mathcal{E}$, and $\mathcal{M}$, defined in Section 2.2.

The first step in specifying the form of L is deciding on a representation class R. At this point of the specialization, we have that L can be any multiset that contains elements from R, as defined in Section 5.1.1. The user, however, may want L to always have exactly one entity (as in Valiant's model). In that case, since as we already mentioned in the previous chapter L is a multiset, if we want to force it to have exactly one element at all times, we will have to do that by the appropriate definitions of $S.M.b$ and $S.M.d$. On the other hand, the user may want to handle L's of infinite size. She may represent an entity as a string, attempting maybe to represent its phenotype. It may be that any string in $\{0,1\}^n$ could represent an entity, or it may be that only a set of strings that satisfy some intricate pattern can be regarded as entities.

The environment E may be dynamic and represent, for example, just the humidity and the temperature of the physical environment of the entities. Or it may be static, as in Valiant's model where the environment is a distribution on the various situations the entities may find themselves into. Note that modelling the environment in a manner similar to Valiant is quite general and for that reason it will be of interest to us.

The evolutionary mechanism may also be either static or dynamic. It is difficult to imagine a model where the mechanism itself evolves into many different manifestations of different kind, however, it is not as strange to imagine a system where some parameters of the evolutionary mechanism change along time.

An example of a partial specialization of MM follows.

6.1.1 Resource-Bounded Meta-Model

MM does not restrict the resources of the evolutionary process in any way. Though, the user may be interested in some process that is efficient in some sense. MM gives a lot of freedom to the user on how to define efficiency. In this section we sketch one of the options. We introduce the Resource-Bounded Meta-Model (RBMM) which restricts the power of MM in order to come closer to modelling the fact that resources in the physical setting are limited. We will bound the resources of the evolutionary process using bounds of a computational nature. As such, we will also understand the evolutionary steps as results of computation.

As we mentioned above, somewhere along the specialization of MM the user needs to define the form of the representations of entities, i.e. the representation class R. Given R we can always define a length function; for an entity l we denote its length as $|l|$. For example, if $R = \cup_{n \geq 0}\{0,1\}^n$ then, for $l \in R$, we may define $|l|$ to be equal to the length of the string l.

The Resource-Bounded Meta-Model (RBMM) is characterized by the following. $S.M.b$, $S.M.d$, $S.M.Mu$ and $S.M.N$ are required to run in polynomial time, and also the sets to which $S.M.N$ maps should be polynomially bounded in size. The said polynomial bounds, of either time or size, are in $|l|$ where l is the first argument of the above functions.[1]

---

[1] It is standard practice in the context of computational complexity to model efficiency by using polynomial bounds.

Note here that we have not yet decided on a specific model of computation. However, the one that naturally arises given our above definitions is one which understands efficiency as *local* efficiency. Specifically, the computations are done in parallel, with one processor per entity. In this setting, for example, $|L|$ could be exponential and the computation could still be efficient.

6.2 Modelling Evolvability

The user needs to take into consideration what is the notion of evolvability that she wants to use. The highest-level variable of interest is S^t. The evolutionary process gives rise to a certain random sequence $S^0, S^1, S^2, \ldots$. A definition of evolvability will require certain properties of this sequence. Moreover, the user may quantify along the classes $\mathcal{L}$, $\mathcal{E}$, and $\mathcal{M}$ in various ways when setting up a definition for evolvability. She may pick out specific elements from those classes, or even take probabilities on them.

For example, concerning the evolutionary mechanism, the user would need to choose whether the latter is static or dynamic. In case it is static, she may want to be able to choose it a priori (e.g., in the case of evolution on earth, one may consider that the evolutionary mechanism is set in stone and she wants to work with that mechanism only, or maybe work with just a small number of mechanisms), or she may just care about the existence of some mechanism that gives rise to the evolution of a certain entity (e.g., in case where the mechanism is a computer program, she may just want to see whether a program exists under which a certain entity evolves), or maybe she wants a bunch of mechanisms that all together have a fair probability to occur (e.g., in a setting where she would have no control on which mechanism is used and she wants to see whether if things are left to luck, there is still a chance that some certain entity will evolve).

In the next section we consider evolvability under the assumption that the evolutionary mechanism M is static. In this setting we give a definition of evolvability that tries to capture the notion as it relates to evolution on earth.

6.2.1 Evolvability under a Static M

In this section we concern ourselves with evolution under a static evolutionary mechanism M. In Appendix A.2.1, we argue why a static M is a reasonable assumption in the setting of evolution on earth.

Let's start with our basic assumption, i.e., M is static. This is captured by the fact that **stepM** is the identity function. We want to get now to a definition of evolvability that is in line with evolution on earth.

First we need a distance measure $\text{dis}(\cdot\,,\cdot\,)$ that maps from $C \times R$ to $[0, 1]$, where C is a class of entities[2], and $\text{dis}(l, l')$ is the distance between l and l'.[3] Given that, we may say that an entity $l \in C$ has evolved if there exists an entity (technically, a representation of an entity) $l' \in L$ whose distance from l is very small.

The intuition that we want to serve here is that a class of entities C is evolvable if there exists an evolutionary mechanism $M \in \mathcal{M}$ that can give rise to C. Let us stress here that the mere existence of such a mechanism would be enough to give us evolvability. Note that MM allows for an enormous class of different evolutionary mechanisms to be used. However, we expect that the user will restrict the class of possible mechanisms $\mathcal{M}$ so that it represents only those mechanisms that make sense in her application of interest.

Moreover, we understand evolvability to represent the fact that an entity can evolve with a non-negligible probability. The probabilities in question apply to the initial conditions of the system. Specifically, the initial set of entities, L^0, and the initial environment, E^0, may be chosen probabilistically, and this is what we do. What we need here is for the user to define two probability measures μ_{L^0} and μ_{E^0} which dictate the probability of certain L^0 and E^0 being the case, respectively.[4]

Now, the last step before we define evolvability is to define what it means for a class of entities C to have evolved. We want something like a predicate that represents the event that a class of entities C has evolved. We are going to define two different predicates $e^{\text{qual}}(C, S, \epsilon)$ and $e^{\text{quant}}(C, S, \epsilon)$. The first predicate will correspond to a qualitative understanding of evolvability which roughly says that a class C has evolved in state S, if for every entity l in C there is an entity in the system (during state S) that is similar to l. The second predicate is similar to the first one, however, it also cares about the number of entities in C and in the system. Let us give now the formal definitions.

Definition 6.2.1. For an entity class C, a state of the system S, and $\epsilon \in$

---

[2] Note that similar to Valiant we differentiate between entities (as found in C) and their representations (as found in L).

[3] We may use here a function similar to Valiant's and Feldman's performance metrics, as introduced in 3.1.3 and 3.3.1. Only we are going to call them distance measures since we do not understand the target entity as an "ideal" entity.

[4] If the environment is static then E^0 would actually correspond to the environment along the whole run of evolution.

$(0,1)$ we have

$$e^{\text{qual}}(C,S,\epsilon) \Leftrightarrow \forall l \in C \; \exists l' \in S.L : \text{dis}(l,l') \leq \epsilon.$$

Definition 6.2.2. For an entity class C, a state of the system S, and $\epsilon \in (0,1)$ we have

$$e^{\text{quant}}(C,S,\epsilon) \Leftrightarrow \exists f : C \to S.L, \; \forall l \in C : \text{dis}(l,f(l)) \leq \epsilon,$$

where f is one-to-one.

Thus, whereas in the qualitative understanding we care for every $l \in C$ to be mapped to some entity $l' \in S.L$, in the quantitative one we require the whole of C to be mapped one-to-one on $S.L$. The first predicate corresponds to questions of the type "can lions evolve in the system?" while the second predicate may answer the question "can a thousand lions evolve in the system at the same time?" Note that the second predicate may be quantitative when considering the evolution of entities, but if we consider the evolution of the system as a whole then the second predicate may too be understood to have a qualitative nature. Through the latter interpretation we would be answering a question of the type "can the system evolve so that it contains a thousand lions?"[5]

In other words, whereas qual-evolvability is truer to our initial understanding of evolvability and focuses on the possibility of a certain entity to evolve, quant-evolvability has a flavor of population dynamics since it is also concerned with numbers of entities.

With all the above in hand we can define evolvability for classes of entities C of finite size as follows. Let us start with the qualitative interpretation. We remind the reader that $\mathcal{M}$ is the class of all possible evolutionary mechanisms, $\mathcal{E}$ is the class of all possible environments, and $\mathcal{L}$ is the class that contains all possible multisets of entities.

Definition 6.2.3. Let μ_{L^0} and μ_{E^0} be probability measures on $\mathcal{L}$ and $\mathcal{E}$, respectively. A finite entity class C is *qual-evolvable* w.r.t. μ_{L^0} and μ_{E^0} if there exist $M \in \mathcal{M}$, and polynomials $g(1/\epsilon)$ and $h(1/\epsilon)$ such that for all $\epsilon \in (0,1)$,

$$\Pr_{\substack{L^0 \sim \mu_{L^0} \\ E^0 \sim \mu_{E^0}}} \left[\exists t \leq g(\tfrac{1}{\epsilon}) : \; e^{\text{qual}}(C,S^t,\epsilon)\right] > \frac{1}{h(1/\epsilon)},$$

where $S^0 = (L^0, E^0, M)$ and $S^{t+1} = \mathbf{step}(S^t)$.

---

[5] Clearly, we assert here that we understand a system with one lion to be qualitatively different from a system with a thousand lions.

Remark 6.2.1. In the evolvability definition above we require the existence of a polynomial h that bounds the rate of decrease of the success probability as ϵ grows smaller. In the context of computational complexity it is standard practice to bound probabilities like that in order to model the fact that the probability does not decrease too fast.

However, in concrete applications of our framework the user may find this to be too restrictive and not representative of the needs of the application. In such a case, the user may want for example to allow the probability to decrease exponentially with n. In Section 7.3 we will examine a certain system and see how the above choice affects the evolvability of classes in the system.

Similarly, we require evolution of the entities of interest to happen in a timeslot t that is bounded by a polynomial g in order to model for the fact that we do not want the process to take too long. Similar concerns with h apply here too.

We also define evolvability for classes C_n of entities of arbitrary complexity. Here n corresponds to the complexity of the entities in C_n. As we said before, evolvability is also in a sense a kind of complexity, however, here when we say complexity we are thinking more of a superficial kind of complexity as in the number of traits of a certain organism, or the length of its DNA.

Definition 6.2.4. Let $\mathcal{L}_n$ and $\mathcal{E}_n$ be such that $\mathcal{L} = \bigcup_{n\geq 1} \mathcal{L}_n$ and $\mathcal{E} = \bigcup_{n\geq 1} \mathcal{E}_n$. Let μ_{L^0} and μ_{E^0} be such that $\mu_{L^0,n}$ and $\mu_{E^0,n}$ are probability measures on $\mathcal{L}_n$ and $\mathcal{E}_n$, respectively. A sequence of entity classes $\{C_n\}_{n\geq 1}$ is *qual-evolvable* w.r.t. μ_{L^0} and μ_{E^0} if there exist $M \in \mathcal{M}$, and polynomials $g(n, 1/\epsilon)$ and $h(n, 1/\epsilon)$ such that for all n, and $\epsilon \in (0, 1)$,

$$\mathop{\mathbf{Pr}}_{\substack{L^0\sim\mu_{L^0,n}\\ E^0\sim\mu_{E^0,n}}} \left[\exists t \leq g(n, \tfrac{1}{\epsilon}) :\ e^{\text{qual}}(C_n, S^t, \epsilon)\right] > \frac{1}{h(n, 1/\epsilon)},$$

where $S^0 = (L^0, E^0, M)$ and $S^{t+1} = \mathbf{step}(S^t)$.

Now we can extend both of the above two definitions to the quantitative interpretation.

Definition 6.2.5. The definition of *quant-evolvable* is the same as *qual-evolvable* except that we use e^{quant} instead of e^{qual}.

Our definition here of evolvability, in line with Section A.2.2, makes clear the importance of being explicit about the initial state of the system, here captured by μ_{L^0} and μ_{E^0}, when talking about evolvability. Moreover, note that our definition is more flexible than Valiant's, whose definition renders entities *generically* evolvable, under any environment, starting from any initial state.

We also define here a stronger version of evolvability which captures the idea that there exists an evolutionary mechanism under which a certain class of entities is bound to evolve.

Definition 6.2.6. A class of entities C is *universally evolvable* if C is evolvable w.r.t. all pairs of μ_{L^0} and μ_{E^0} (using the same evolutionary mechanism).

Universal evolvability is a property that in general is very hard to satisfy, and is fairly similar in its interpretation of evolvability to the interpretation implicit in Valiant's definition of evolvability.

Remark 6.2.2. In certain applications we want to know how the probability of evolution is affected by the size of the entity classes C under consideration. In the next chapter we will consider in several examples qual-evolvability as in Definition 6.2.3, but slightly altered so as to render a class evolvable only if the probability of evolution decreases at most polynomially as a function of the size of the entity class under question. Note that in order to do that we have to consider entity classes of some certain type but of arbitrary size.

Remark 6.2.3. Note that in our definitions of evolvability we find it enough, roughly, if a class of entities C shows up in only one timeslot. That is, we do not require for the entities of C to stick around for a long time in order to call C evolvable. However, we could instead have required for the entities in C to be in the system for $q(\epsilon)$ subsequent timeslots, with $q(\epsilon)$ possibly getting larger as ϵ is getting smaller. Such a solution may make sense especially in cases where the the probability of an entity to go extinct given that it has existed for one timeslot (or even $q(\epsilon)$ timeslots) is non-negligible.

In general, it is an interesting question to examine the sustainability of certain classes. This could be a direction for future work.

Remark 6.2.4. The reader may have noticed some kind of directedness in our basic question in this work. Instead of asking what is evolvable, we ask for specific entities whether they are evolvable. Even though those specifics can be quite general, e.g., we do not need to ask whether the great white shark is evolvable, we can ask whether fish are evolvable, our question still is directed. However, in the setting where the entities under consideration are one-to-one with the elements of the representation class R that we use (let us assume that f is the corresponding bijection from the set of all entities under consideration to R), we can easily reconcile this through the following definition. The class of all evolvable entities is $\{f^{-1}(r) \mid r \in R \wedge (f^{-1}(r) \text{ is evolvable})\}$.

One may wonder whether we really address the issue here. Note that what we do is formalize a certain notion, but we do not force any way of analysis of the system at hand so as to find the corresponding classes of interest. The user of the framework is free to analyse the system in a way that makes it easier either to find what entities are evolvable in general or to find for specific entities whether they are evolvable.

6.3 Fitting Valiant's Framework into Ours

In this section we specialize RBMM introduced in Section 6.1.1 in order to get Valiant's model. As the reader may guess, we need to define two things, the evolutionary process and the definition of evolvability. In the following whenever we use S^t with t undefined we mean 'for an arbitrary t'.

Consider the following specialization. Functions **stepE** and **stepM** are the identity functions. $S^t.E$ contains the distribution D over X. $S^t.M.N$ returns the empty set[6], and $S^t.M.b$ and $S^t.M.d$ are always 1. $S^t.M.Mu$ is the mutator function of Valiant.[7] $S^t.M$ also contains an ideal function f so that the process may use Perf_f, a parameter vector $P = (R, \mathrm{Neigh}, \mu, t)$, and a polynomial s that determines the sample size. Note that f, P, and s are variables and are only specified through the definition of evolvability.

In this context Valiant's definition of evolvability (Definition 3.1.6) can easily be translated to our framework as follows.

Definition 6.3.1. We say that a class of functions C is *evolvable* if there exists a parameter vector $P = (R, \mathrm{Neigh}, \mu, t)$, and polynomials $s(n, 1/\epsilon, 1/\delta)$ and $g(n, 1/\epsilon, 1/\delta)$ such that for all distributions D, for every n, every $f \in C_n$, every $\epsilon \in (0, 1)$, every $\delta \in (0, 1)$, and every $r_0 \in R_n$, with probability at least $1 - \delta$, the unique $r \in S^{g(n,1/\epsilon,1/\delta)}.L$ will have $\mathrm{Perf}_f(r, S^t.E.D) > 1 - \epsilon$, where $S^0.L = \{r_0\}$, $S^t.M.P = P$, $S^t.M.s = s$, and $S^t.E.D = D$.

Remark 6.3.1. The reader may be a bit puzzled on how our framework whose outlook is so much different than Valiant's can accommodate the latter. Specifically, Valiant makes use of oracles and has ideal functions that drive evolution. He essentially models what is referred to in biology as "directed evolution" [Rid04, pp. 340], an idea that clashes with that of natural selection.

---

[6] Note that $S^t.M.N$ is a different function than Valiant's $\mathrm{Neigh}(r, \epsilon)$ function.

[7] Precisely speaking, our mutation function $S^t.M.Mu(l, S^t)$ is equal to $Mu(S^t.M.b, S^t.M.D, S^t.M.P, l, \epsilon, \delta, S^t.M.s(n, 1/\epsilon, 1/\delta))$ where Mu is Valiant's mutation function.

The thing, though, is that even though we designed a framework without expecting to use any oracles and ideal functions we also did not restrict it so as to not allow for the use of those. All in all, this example serves to demonstrate how flexible our framework really is.

Chapter 7

Applications of our Framework

In this chapter we examine various systems to illustrate our framework. As the reader will notice, sometimes we simplify our notation in order to avoid clutter. For example, if the function giving the probability of reproduction is static we may denote it as $S.M.b$ instead of $S^t.M.b$, or we may even denote it as b. Moreover, if arguments to functions like $S.M.b$ or $S.M.Mu$ are redundant or implicit we may omit them, and write, e.g., $S.M.b(l)$ instead of $S.M.b(l, \text{neigh}, S.E)$.

Let us start with a couple of definitions that will make our exposition clearer. A *discrete trait* is a trait that can assume finitely many manifestations, e.g., a trait that corresponds to the color of the eyes assuming that the latter can either be brown, blue, or green. A *continuous trait* is a trait that can vary continuously, e.g., a trait that corresponds to the color of the eyes assuming that the color of the latter can be any combination of different proportions of brown, blue, and green. Lastly, two numbers x and y are α-*close* if $|x - y| \leq \alpha$.

7.1 Peppered Moth

We will start with a trivial example in order to clarify the use of our framework. Suppose that there is a large population of white (light-colored) peppered moths, maybe with the occasional black (dark-colored) moth. We model the color of the moth as a discrete trait that can only take two values, black and white. Specifically, we use the representation class $R = \{0, 1\}$, where the 0 stands for black and the 1 for white. We remind the reader that the set of entities at timeslot t (which we denote by $S^t.L$) is a multiset that only contains elements from R.

Given that the trait of interest is discrete, we also use a discrete distance function, specifically, $\text{dis}(l, l') = 0$ if $l = l'$ and $\text{dis}(l, l') = 1$ otherwise. Moreover, note that since the distance function is discrete, the accuracy parameter ϵ has no significance in this setting.

The white moths are well protected on the light-colored trees that occupy the area since they are hidden from their predators. However, with the coming of the industrial revolution the trees assume a very dark color, say, overnight. And now the white moths can be easily spotted by their predators. The probability that a black moth dies $S.M.d(0)$ is equal to some small real number $p_d \leq 1/2$. Since we will not allow for those probabilities to change over time, which is a result of assuming a static environment, we will write d instead of $S.M.d$ and similarly for $S.M.b$ and $S.M.Mu$. Hence, we have $d(0) := p_d$. Moreover, we opt for $d(1) := 1 - p_d$.[1] Note that $d(1) \geq d(0)$, meaning that white moths are likelier to die than black moths. The probability that a black or white moth gives birth are the same and equal to some $p_b \in (0, 1)$, i.e., $b(l) := p_b$ for all $l \in R$. We assume that births happen at the start of a timeslot and deaths at the end of a timeslot.

Moreover, there is a very small probability p_m that a white moth will reproduce a black one, and the reverse. Specifically, for all $l \in R$ we have $Mu(l) = l$ with probability $1 - p_m$, and $Mu(l) = 1 - l$ with probability p_m.

We will consider the case where the initial set of entities L^0 contains no black moths. Is the class C that contains only black moths evolvable? Let us first consider qual-evolvability.

Qualitative evolvability. The case for qual-evolvability is trivial and we only mention it here in an effort to showcase more options in our framework. It is enough that with positive probability[2] there exists some timeslot t such that $S^t.L$ contains a black moth.

Remember from the definition of qual-evolvability that $S^0.L = L^0$. It is easy to show that there is a positive probability that $S^1.L$ contains a black moth. Specifically, suppose that $S^0.L$ contains at least one white moth. This moth will give birth to a black moth with probability $p_b p_m$. And this gives us qual-evolvability. Precisely speaking, any class of black moths C is qual-evolvable, according to Definition 6.2.3, w.r.t. any μ_{L^0} and μ_{E^0} such that there exist a set of entities L and an environment E with $\mu_{L^0}(L) > 0$, $\mu_{E^0}(E) > 0$, $|L| > 0$, $E.p_b > 0$, and $E.p_m > 0$.[3]

---

[1] One way to interpret the choice of $d(\cdot)$ is that a predator will either eat a black moth or a white moth. However, the main reason behind this modelling choice is to simplify the exposition.

[2] We do not care about the rate of decrease of the probability in terms of ϵ, because as we mentioned above, ϵ has no significance in this setting.

[3] Note that $E.p_b$ stands for what we denoted by p_b above. However, since p_b is a

Let us now turn to quant-evolvability.

Quantitative evolvability. We consider here finite entity classes C that, as we said above, contain only black moths. The probability that C evolves, in the quant-evolvability sense, is again positive and constant since it is not affected by ϵ. The exact probability is a function of $|L^0|$ and $|C|$. To be more precise we have that, for a given L^0, there exists $\alpha \in (0,1)$ such that the probability that some arbitrary entity class C evolves is lower bounded by $\alpha^{|C|}$. The derivation of this result can be found in Appendix A.4.

This result gives us quant-evolvability but it does not really tell us much. In a sense it tells us that we can get an arbitrarily large class C to evolve, but it ignores the fact that the probability decreases exponentially as the size of C increases.[4]

In line with Remark 6.2.2 we would rather have a lower bound that decreases at most polynomially. Let us try to get such a bound now. Let X_t be a random variable that corresponds to the number of black moths in the system at timeslot t. Let $I_t := X_t/X_{t-1}$ be the ratio of black moths in the system between timeslots t and $t-1$.

We have that

$$\begin{aligned}\mathbf{E}[I_t] \geq & 2(p_b(1-p_m)(1-p_d))+ \\ & 1(p_b(1-p_m)p_d)+ \\ & 1(((1-p_b)+p_bp_m)(1-p_d))+ \\ & 0(((1-p_b)+p_bp_m)p_d),\end{aligned}$$

where the summands correspond to the events that a black moth reproduces and does not die (results in 2 moths for the next timeslot), it reproduces and dies (results in 1 moth), it does not reproduce or reproduces a white moth and does not die (results in 1 moth), it does not reproduce or reproduces a white moth and dies (results in 0 moths), respectively. The expression is an inequality, instead of an equality, because we do not take into account that some black moths are the result of white moths giving birth. If we tidy this up we get

$$\mathbf{E}[I_t] \geq 1 + p_b(1-p_m) - p_d := \lambda. \tag{7.1}$$

---

parameter defined by the environment we may also denote it by $E.p_b$. Similarly for $E.p_m$.

[4] In most of the examples that we consider in this chapter we will be able to get similar lower bounds, i.e., lower bounds that give us the evolvability of arbitrarily large entity classes C but with a probability that decreases exponentially as a function of $|C|$. In the case of continuous traits, ϵ is also a significant parameter that would affect the rate of decrease of such trivial lower bounds.

Suppose that $\lambda > 1$, i.e., $p_b(1 - p_m) > p_d$. We will demonstrate quant-evolvability of any entity class C with a probability that decreases at most polynomially with $|C|$. From (7.1) we have that[5]

$$\mathbf{E}[X_t] = \mathbf{E}[\prod_{i \le t} I_t]X_0 \ge \mathbf{E}[\prod_{i \le t} I_t] = \prod_{i \le t} \mathbf{E}[I_t] \ge \lambda^t. \tag{7.2}$$

Moreover, we have that since the population can at most double at each timeslot that

$$X_t \le 2^t. \tag{7.3}$$

For any random variable Y that takes values from the natural numbers and there exists Y_{max} such that $Y \le Y_{max}$ with probability 1, we have that for every natural number $k \le Y_{max}$ it holds

$$\begin{aligned}
\mathbf{E}[Y] &= \sum_{y=0}^{Y_{max}} y\mathbf{Pr}[Y = y] \\
&= \sum_{y=0}^{k-1} y\mathbf{Pr}[Y = y] + \sum_{y=k}^{Y_{max}} y\mathbf{Pr}[Y = y] \\
&\le (k-1)\mathbf{Pr}[Y < k] + Y_{max}\mathbf{Pr}[Y \ge k] \\
&= (k-1)(1 - \mathbf{Pr}[Y \ge k]) + Y_{max}\mathbf{Pr}[Y \ge k],
\end{aligned}$$

which gives us that

$$\mathbf{Pr}[Y \ge k] \ge \frac{\mathbf{E}[Y] - k + 1}{Y_{max} - k + 1}.$$

Hence together with (7.2) and (7.3) we get for the probability of the evolution of C that

$$\mathbf{Pr}[X_t \ge |C|] \ge \frac{\lambda^t - |C| + 1}{2^t - |C| + 1}.$$

Now we can easily find a timeslot t such that $\lambda^t > |C| + 1 \Leftrightarrow t > \log_\lambda(|C| + 1)$ that will give us our result. For example, for $t' = \log_\lambda(|C| + 2)$ we get

$$\mathbf{Pr}[X_{t'} \ge |C|] \ge \frac{1}{(|C| + 2)^{1/\log_2 \lambda} - |C| + 1},$$

which is a bound that fits what we were after, i.e., C is evolvable with a positive probability that decreases at most inverse-polynomially in $|C|$.

---

[5] Here we assume for simplicity that L^0 contains at least one black moth and as such $X_0 \ge 1$. In case it does not, it takes only constant probability, namely, $p_b p_m$, for a black moth to arise at the first timeslot as a result of a birth of a white moth.

7.2 A Continuous Trait

We will examine here a system where entities have one continuous trait. Moreover, we will introduce a simple fitness landscape with respect to that trait.

A specific amount of energy, same for all entities, is available to each entity. Each entity l has a specific efficiency l_f in using that energy. The higher the efficiency, the better the fitness of an entity, which translates to higher probability to reproduce and smaller probability to die.

Our initial expectation here, is that the system will converge to a state were most entities l have l_f close to 1. However, we cannot say that before defining an evolutionary mechanism. We will actually consider a simple and natural evolutionary mechanism under which we will demonstrate speciation, in the sense that entities with high energy efficiency will be extremely more prevalent than other entities. Moreover, our analysis will also address the population dynamics in the system.

We define $R = \{l = \langle l_f \rangle \mid l_f \in [0,1]\}$, where l_f is the energy efficiency of l. We remind the reader that $S.L$ is a multiset with elements in R. Let $\mathrm{dis}(l, l') = |l_f - l'_f|$ be the distance function that we will use.

For a real number $x \in [0,1]$, let $U(x)$ be a random variable that takes values uniformly randomly from $[x-\eta, x+\eta]$, where η is a small real number such that $\eta \leq 1/2$. Let $U^*(x) := \min(\max(U(x), 0), 1)$ be the function that for all $x \in [0,1]$ brings the value of $U(x)$ in $[0,1]$; note that if x is η-close to one of the boundaries, i.e., 0 or 1, then there is a peak in the probability density of $U^*(x)$ to equal that boundary.

We will consider the evolutionary mechanism where $S.M.Mu(l, S) = l'$ with $l'_f = U^*(l_f)$ being the energy efficiency of the mutated entity. Moreover, for all entities l we have $S.M.d(l) := 1 - S.M.b(l)$ and $S.M.b(l) := l_f$.[6] In the following we will abbreviate the above functions by omitting '$S.M$'.[7]

Note that the above probabilities mean that the population can start growing exponentially in size when we have entities l with l_f close to 1. However, in a real system we would expect the limited resources to bound the maximum possible size of the population. We will assume that this bound is large, i.e., the system can accommodate a large number of individuals.[8]

---

[6] Note that this is a simple and somewhat extreme model. A nicer model would be to have $S.M.b(l) := p + (1-2p)l_f$, for some $p \in [0, 1/2)$, which would yield $S.M.b(\langle 0 \rangle) := p$ and $S.M.b(\langle 1 \rangle) := 1 - p$. We will consider this model later.

[7] Moreover, we should have written here $S.M.b(l, \mathrm{neigh}, S.E)$, instead of $S.M.b(l)$, according to the form of $S.M.b$ in our framework; analogously for $S.M.d$. But for the sake of simplicity we do not.

[8] This is a restriction that could possibly follow from modifying the definitions of $S.M.b$

Then if we focus on the distribution of the entities and not on the actual number of entities, the analysis for the unbounded-population case and for the bounded one should converge to the same results.[9]

Evolvability. The most useful results in terms of understanding the behavior of the model we obtained through simulations of the system. Our system has only one parameter, namely, η.

For a small η, as seen in Figure 7.1, we have that all the entities l have energy efficiency l_f really close to 1.

While for larger values of η, as seen in Figure 7.2, we have also entities with energy efficiencies much smaller than 1 in the system. The reason is that any given entity has a good chance to give birth to entities with fairly smaller energy efficiency. The small peak that we see in the ratio of entities a little before 1 has to do with the fact that our mutation mechanism is defined using function $U^*(x)$ that, as we mentioned before, gives a quirky behavior to the mutation probabilities near the boundaries.

All in all, our results suggest that there exists an interval around the ideal manifestation of the trait (in our case the trait is energy efficiency and the ideal value is 1) whose size depends on η, i.e., the parameter which determines the variability of the mutations, wherein most of the entities reside. Moreover, the relative frequencies of the entities along the interval converge very fast in the simulation.

---

and *S.M.d.* However, for simplicity we do not make the latter two functions account for the fact that resources are limited and as such the population will be bounded at some point.

[9] Note that a good reason to examine the unbounded-population case directly, instead of the bounded-population case, is that in the former entities are independent from each other, apart from the interrelation that arises from the act of reproduction.

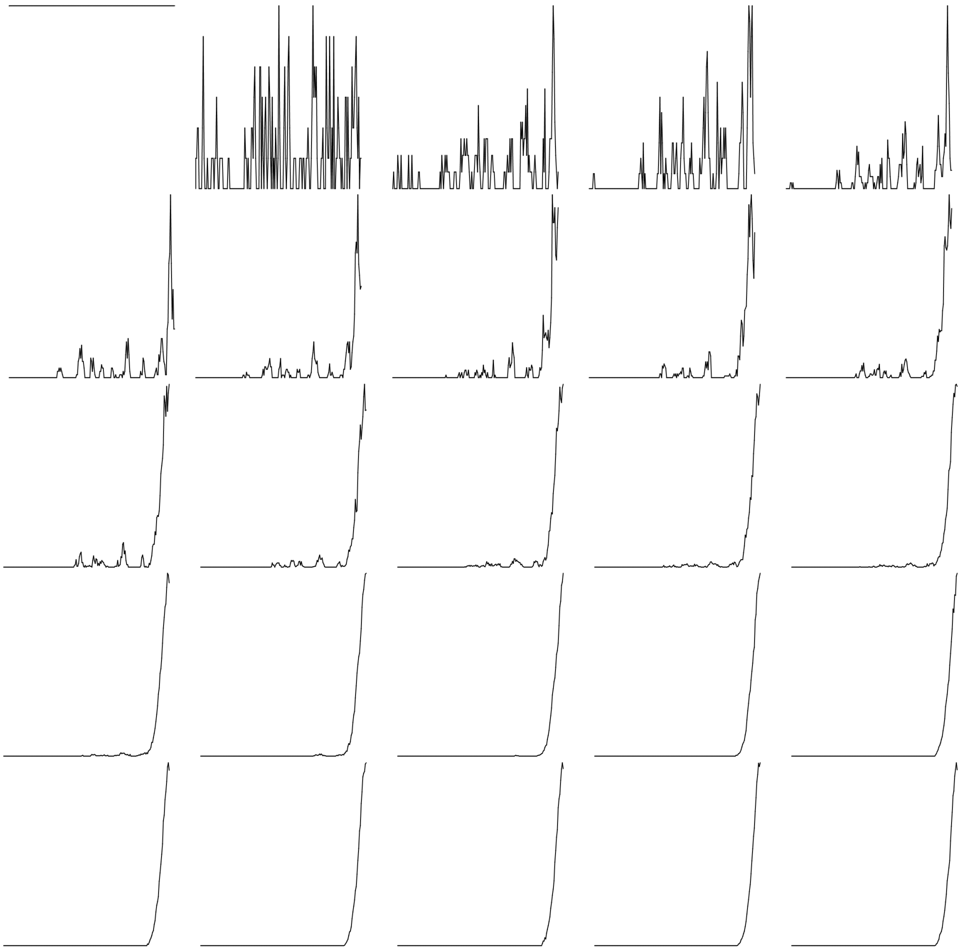

Figure 7.1: Simulation of the system for $\eta = 1/133$ for 300 timeslots. There are 25 graphs in this figure, each of which corresponds to a point in time along the first 300 timeslots; the time interval between successive images is roughly 300/25 timeslots. The horizontal axis corresponds to the energy efficiency interval $[0, 1]$. The vertical axis depicts the relative frequencies of entities with different energy efficiencies. In the initial distribution the entities are uniformly distributed along $[0, 1]$.

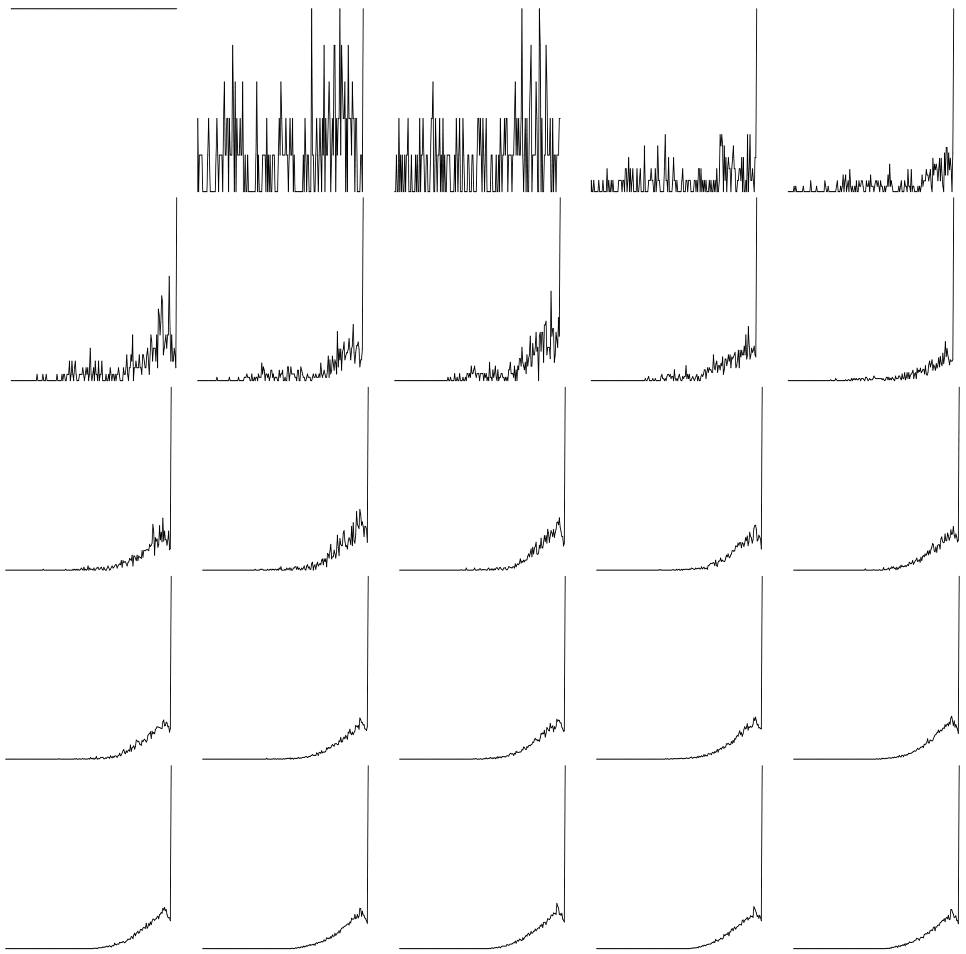

Figure 7.2: Simulation of the system for $\eta = 1/20$ for 300 timeslots. There are 25 graphs in this figure, each of which corresponds to a point in time along the first 300 timeslots; the time interval between successive images is roughly 300/25 timeslots. The horizontal axis corresponds to the energy efficiency interval $[0, 1]$. The vertical axis depicts the relative frequencies of entities with different energy efficiencies. In the initial distribution the entities are uniformly distributed along $[0, 1]$.

7.3 Food on a Grid

Here we examine an example mainly concerned with biogeography. In this system entities live on the nodes of a grid and they eat the food given to them by some process that places food on the grid; we will use the term *food-process* for the latter. We want to examine how the entities will spread in space depending on the specifics of the food-process.

In Section 7.3.2 we will examine two extreme, and somewhat uninteresting cases. The two cases are (i) the situation where the food-process lays food at all timeslots on the same nodes of the grid, and (ii) the situation where food is uniformly randomly dropped on the grid. We will use our results on those two cases to examine a more interesting case in Section 7.3.3, where we will introduce a food-process that brings about the idea of a fitness landscape more clearly than the food-processes in cases (i) and (ii).

We start with some preliminaries.

7.3.1 Preliminaries

Consider the grid $G_m = \{(x, y) \mid x, y \text{ are integers such that } 0 \leq x \leq m - 1, 0 \leq y \leq m - 1\}$ of size $m \times m$, and a process that places food on the nodes of G_m at the start of each timeslot. Let $F^t \subseteq G_m$ be the set of nodes where the food-process places food at timeslot t. Let the representation class $R_m = G_m$ where we view entities $l \in R_m$ as $l = (l_x, l_y)$. $\mathcal{L}_m$ is the class of all possible L's, i.e., all possible subsets of R_m. Hence, for all timeslots t we have that $S^t.L$, i.e., the population at timeslot t, is a subset of R_m.

On each node of the grid there may live at most one entity; if an entity l is reproduced and is supposed to go on a node that already carries another entity then l is discarded. An entity needs to eat every c timeslots in order to sustain itself, where c is an integer such that $c \geq 1$, and it can only eat food present at its own location.[10] An entity will always eat the food if it is available. Moreover, food goes bad if not consumed in 1 timeslot and is discarded from the system. Reproductions happen at the start of a timeslot and deaths at the end of a timeslot.

Let $w((x, y))$ be the set of neighboring nodes to (x, y); every node has four neighboring nodes and we think of the boundaries of the grid in such a

---

[10] This rule determines $S.M.d$, the function that determines the probability for an entity to die.

way that the grid forms a torus. Specifically,

$$\begin{aligned}w((x,y)) = \{&((x-1+m) \bmod m, y)\\ &((x+1+m) \bmod m, y)\\ &(x,(y-1+m) \bmod m)\\ &(x,(y+1+m) \bmod m)\}.\end{aligned}$$

Let Q denote the predicate such that for every $X \subseteq G_m$, and $(x,y),(x',y') \in G_m$, $Q(X,(x,y),(x',y'))$ is true iff there exists a chain of neighboring nodes in X that starts from (x,y) and finishes with (x',y').

Let $v(l)$ be the set of neighbors of entity l. At timeslot t we have that $v(l) = w((l_x,l_y)) \cap S^t.L$.[11]

7.3.2 First Results

We will consider here evolvability under the class of evolutionary mechanisms $\mathcal{M}$ that contains all those mechanisms that use a mutation function Mu such that $Mu(l) \in w((l_x,l_y))$.[12] Specifically, we will consider qual-evolvability as defined in Section 6.2.1. Our distance function $\text{dis}(l,l')$ equals 0 if $l_x = l'_x$ and $l_y = l'_y$, and 1 otherwise.[13]

In the following when we refer to Manhattan distance between two nodes we refer to the distance of the shortest strictly horizontal and/or vertical path between the two nodes. Note that $m-1$ is the maximum Manhattan distance between two nodes on the grid. The Manhattan distance between two nodes (x,y) and (x',y') lower bounds the number of timeslots needed for an entity to come alive on node (x',y') as a result of life being propagated among neighboring nodes starting from an entity that resides on node (x,y). This is a direct result of the fact that $Mu(l) \in w((l_x,l_y))$.

A simplified analysis. Here we will examine the qual-evolvability of all entity classes C such that $C \in \mathcal{L}_m$. In contrast with the next section, we will not be concerned with how the size of the entity classes affects their probability of evolution, or how the latter behaves in relation to m, i.e., the parameter that determines the size of the grid and consequently the possible distances among nodes.

As we mentioned above, we will consider two cases concerning the food-process. In both cases $|F^t| = k$, for some positive integer $k \leq m^2$. That is,

---

[11] We remind the reader that $S^t.L$ is the set of entities in the system during timeslot t.

[12] Here $Mu(l)$ stands for $S.M.Mu(l,S)$.

[13] Given that $\mathcal{L}_m$ contains no multisets and our distance function is discrete, qual-evolvability is equivalent to quant-evolvability.

the density q of food on the grid is constant and equal to k/m^2. In case (i) we will have that $F^t = F^{t+1}$, for all $t \geq 0$. In case (ii) we will have that F^t is a subset of G_m chosen uniformly at random among all subsets of G_m of size k.[14]

For both cases (i) and (ii) the following holds. For $c = 1$, i.e., if the entities need to eat at every timeslot, no entity class C such that $|C| > k$ is evolvable, since there is not enough food to sustain as many entities as the size of C. More generally, if C is evolvable then $|C| \leq ck$.

Let us focus on case (i) now. We have that if C is evolvable then for all $l \in C$ there exists $l' \in L^0$ such that $Q(F^0, l, l')$.[15] What is more, we can also easily find a mechanism that makes the above statement work the other way around too. One such mechanism is the mechanism where $b(l)$ is a positive constant, i.e., there exists $p_b \in (0, 1]$ such that $S^t.M.b(l) = p_b$ for all timeslots t, and $Mu(l)$ chooses uniformly randomly among the entities on the four neighboring nodes to (l_x, l_y).

Let us demonstrate why this mechanism works as promised, i.e., if for all $l \in C$ there exists $l' \in L^0$ such that $Q(F^0, l, l')$ then C is evolvable. Suppose that for all $l \in C$ it holds that there exists $l' \in L^0$ such that $Q(F^0, l, l')$, then the above mechanism has as a result that all $l \in C$ have a positive probability to evolve, and that probability is independent of any accuracy parameter ϵ. Moreover, because of the nature of F^t in case (i), if an entity evolves it will be sustained forever. Hence, C is evolvable, according to Definition 6.2.3.

Taking m into account. Here we take into account how the probability of evolution is affected by the size of the class to evolve. The size of the classes C could be equal to m or larger[16].

In line with Remark 6.2.2, we address this concern by parameterizing all of our components with m and focusing on how things turn out in relation to m. Hence, instead of $C \subset G_m$, we consider $C_m \subset G_m$.[17] We will consider only the most interesting of the two cases described above, i.e., case (ii) where for some density $q \in [0, 1]$ we will have that F_m^t is a subset of G_m chosen uniformly at random among all subsets of G_m of size $k_m = \lceil qm^2 \rceil$.

Let us start by demonstrating that $|C_m| \leq ck_m$ is not enough to guarantee

---

[14] For case (i) the environment $S.E$ would need to contain (as information) both k and F^0, while for case (ii) only k will be enough.

[15] We remind the reader that L^0 is the initial set of entities.

[16] The size of C can be as large as m^2.

[17] Note here that the subscript m on C_m does not signify the complexity of the entities in C_m as n does in Definition 6.2.4 or as in the example of Section ??. Here m is just the parameter that determines the size of the grid.

the evolvability of C.[18] Let r^t be a predicate such that $r^t((x,y))$ is true iff at timeslot t an entity l such that $(l_x, l_y) = (x,y)$ is reproduced (clearly, reproduced by an entity that resides on a node neighboring (x,y)). We have that $\mathbf{Pr}[r^t((x,y))] \leq 1-(3/4)^4$, where $1-(3/4)^4$ is the probability for $r^t((x,y))$ to happen given that there are entities on all neighboring nodes of (x,y).

Assuming that the density $q \neq 1$, then we have for all $t \geq 1$ that

$$\begin{aligned}
&\mathbf{Pr}[C_m \text{ evolves at timeslot } t] \\
&\quad \leq \mathbf{Pr}[\forall l \in C_m : r^t((l_x,l_y)) \vee (\neg r^t((l_x,l_y)) \wedge (l_x,l_y) \in \bigcup_{t' \in \{t-c,\dots,t-1\}} F^{t'})] \\
&\quad \leq \prod_{l \in C_m} \mathbf{Pr}[r^t((l_x,l_y)) \vee (\neg r^t((l_x,l_y)) \wedge (l_x,l_y) \in \bigcup_{t' \in \{t-c,\dots,t-1\}} F^{t'})] \\
&\quad \leq ((1-(3/4)^4) + (3/4)^4(1-(1-q)^c))^{|C_m|} \\
&\quad \leq \alpha^{|C_m|},
\end{aligned}$$

for some $\alpha \in (0,1)$.

Subsequently, assuming that $L^0 \neq C_m$,

$$\begin{aligned}
&\mathbf{Pr}[C_m \text{ evolves in } g(m,1/\epsilon) \text{ timeslots}] \\
&\quad = \mathbf{Pr}[\bigvee_{t \in \{1,\dots,g(m,1/\epsilon)\}} C_m \text{ evolves at timeslot } t] \\
&\quad \leq \alpha^{|C_m|} g(m,1/\epsilon)
\end{aligned}$$

As such, if $|C_m| \geq m$ then the probability of success decreases exponentially with m. Let us summarize by saying that if the elements of a sequence $\{C_m\}$ are evolvable with a probability that is decreasing at most polynomially with m then $|C_m| \in O(\log m)$. This result is a consequence of the cost that we have to pay (in terms of probability) in order to keep the entities fed. It is still possible though for C_m of size $O(\log m)$ to be evolvable.

Specifically, given the above result the strongest thing that we may possibly prove is that $|C_m| \in O(\log m)$ and $|C_m| \leq ck_m$ is enough to render C evolvable. However, a simple consideration will indicate this to be unlikely. Consider $c=1$, $m=1024$, $|C_m| = \log_2 m = 10$, and say that we have just enough food, i.e., $k_m = 10$. Thus, we ask whether C_m can evolve in a place where we have 10^6 nodes and there is only food for 10 of them. A good guess

---

[18] Note that this restriction, which practically says that there is at least the bare minimum of food on the grid in order to sustain as many entities as those in C_m, was enough to give us evolvability when we did not take m into account.

would be that we cannot get evolvability in this setting; we would instead need to boost k_m, and probably have it in $\Omega(m^2)$.

In order to address this questions we made some simulations of the system.[19] Those indicate the following. It turns out that the system has a threshold behavior. Specifically, there is a threshold value, say q^*, for q that determines whether the probability that everyone dies is negligible or it is close to 1. For example, for $c = 1$ this threshold value of q lies in the interval $[0.59, 0.6]$. Naturally, for larger values of c the value of the threshold q^* gets smaller and smaller.

Simulations show that the density of alive entities in L^0 does not make much difference. Moreover, the density of alive entities, say q_A, for a specific q converges as m gets larger.[20] The fact that F_m^t is uniformly random and $S^t.L \subseteq F_m^t$ for all timeslots t, indicates that the system reaches a state where, in the long run, every configuration of density roughly equal to q_A is pretty much as probable as any other. Or, more precisely, there exists some small real number η such that the great majority of the configurations have density in $[q_A - \eta, q_A + \eta]$. This is the key point in our analysis here.

We have that as m gets larger η gets smaller and the number of nodes on the grid gets larger. As such, we may regard the system to have a behavior where at each timeslot, each node carries an alive entity with probability q_A, independent from the other nodes.[21] And in this setting the probability that a certain entity class $|C_m|$ evolves is $(q_A)^{|C_m|}$. Since $|C_m| \in O(\log m)$ we have that the probability of evolution decreases at most polynomially with m. All in all, the above indicate that for all C_m such that $|C_m| \in O(\log m)$, if $k_m > m^2 q^*$ then C_m is evolvable.

7.3.3 Among the Oases

Now we consider the scenario where there are oases along the grid. There is always plenty of food in an oasis but there is not always life. While in those parts of the grid where there are no oases, there only exist some scarce resources of food. The latter scarce resources may be used by entities so that life may propagate from one oasis to another. We want to see here to what extent this propagation can be successful.

Let us see now how we will model this situation. Our food-process will

---

[19] We simulated the system for grids of dimensions 30×30 to 100×100, and food processes in line with case (ii) for various food densities q between 0 and 1.

[20] For example, for $c = 1$, q_A converges to roughly 0.08 for $q = 0.6$, to roughly 0.44 for $q = 0.7$, to roughly 0.66 for $q = 0.8$, and to roughly 0.84 for $q = 0.9$.

[21] We remind the reader that $k_m > m^2 q^*$ can guarantee to us that q_A is a positive probability.

be a hybrid of the two food-processes of cases (i) and (ii). Specifically, there will be some nodes on the grid that always have food (as in case (i)), corresponding to the oases, while all the rest of the nodes will have food randomly dropped on them (as in case (ii)).

Precisely speaking, we define an oasis as follows. A set of nodes $X \subseteq G_m$ is an *oasis* if it is maximum among all the subsets Y of G_m such that for all timeslots t we have $F^t \cap Y = F^{t+1} \cap Y$, i.e., there is always food on the nodes of Y, and for $(x, y), (x', y') \in Y$ we have $Q(Y, (x, y), (x', y'))$, i.e., Y is neighbor-connected. Note that once an oasis is filled with life, it will remain so intermittently.

Let $O_m \subseteq G_m$ denote the set of nodes that are in oases. We will only consider the evolvability of classes C_m such that $C_m = O_m$. We have for all timeslots t that $F^t \cap O_m = F^{t+1} \cap O_m$, while $F^t \setminus O_m$ is chosen among all subsets of $G_m \setminus O_m$ of size $k_m = \lceil q(m^2 - |O_m|) \rceil$.

We will consider the same evolutionary mechanism as in the previous section. Moreover, we will focus only on settings where the number of different oases in O_m is in $o(m)$, i.e., sublinear. The latter assumption allows us to simplify our problem as follows. A class of oases is evolvable (in the sense that the class of entities on those oases is evolvable) if the most difficult to evolve among those oases is evolvable. This is because if the latter is evolvable then we can always boost the probability for the rest of the oases to evolve by using a polynomial g in $|C_m|$, that determines the number of timeslots available to the process, that increases faster. Moreover, following the results of Section 7.3.2, it is trivial to see that an oasis is evolvable if one of the entities in it is evolvable.

We will assume that there is at least one alive oasis at the initial state, i.e., $|L^0 \cap O_m| \neq 0$ which stands for the fact that there exists at least one alive entity in the oasis. Now, we will focus on the difficult case where there is an oasis that is very far from any other live oases. Given our discussion above this problem boils down to the following. Consider O_m such that $|O_m| = 2$, $|L^0 \cap O_m| = 1$. Let us denote by l^0 the entity that is in $L^0 \cap O_m$ and l the other entity. We want to see whether l can evolve. In order to make the problem difficult, i.e., consider an oasis that is maximally difficult to evolve, as we mentioned above, we assume that the Manhattan distance between the nodes (l_x, l_y) and (l^0_x, l^0_y) is in $\Theta(m)$.[22]

Ultimately, what we want to find is whether having l^0 as a source of

---

[22] Note that the analysis of this setting will give us the behavior of the system, in terms of evolvability, for all situations where there is at least one oasis $\Theta(m)$ steps away from the entities in L^0. This is because if there exists an oasis in O_m that is $\Theta(m)$ steps away from L^0 then, since $|O_m| \in o(m)$, there must exist some oases in O_m that are $\Theta(m)$ steps away from all the other oases in O_m.

life, the scarce food-process between the two oases is enough to make life propagate and reach l with a non-negligible probability. Clearly from our simulation results in the previous section, we have that if $k_m > m^2 q^*$, where q^* is the same threshold (which is a function of c) as in the previous section, then l is evolvable. To see that, note that in the case we are considering now evolvability is easier than in the case before. This is because apart from the random food-process, now we also have at least one entity that never dies; since we assumed that $|L^0 \cap O_m| \neq 0$.

The next question is whether we can expect better evolvability results, than in the previous section, since there are now oases and entities in an oasis never die. Specifically, can we expect here evolution of classes even with smaller density q (of the random food-process) than in the previous setting? Clearly, the presence of oases boosts the probability for evolution, however, asymptotically it makes no difference.[23] The main impediment in the evolution of entities in the previous setting was that life could not propagate large distances if the density was not large enough. In our setting here, as m gets larger, distances grow large as in the previous setting and the same impediment holds here too.

---

[23] Note that this only holds in the case where $|O_m| \in o(m)$, and there exists an oasis in O_m that is $\Theta(m)$ steps away from all other entities in L^0, which roughly means that there is an oasis that is quite far away from all other oases where there is life and as such it is very difficult for the said oasis to evolve. Otherwise, the problem gets easier and we can expect better results.

Chapter 8

Conclusion

Our work borders on philosophy since it is interested in the analysis of a specific notion, that of evolvability. However, we have not tried to answer what evolvability is. Our aim was more practical. We wanted to assist in the formalization of the notion when a specific application is in mind.

Our starting point was Valiant's framework where he formalized the evolvability of functions under a certain model of evolution, with the assumptions that there exists an ideal function and that evolution is executed by an algorithm that has access to an oracle which provides the algorithm with examples of the said ideal function.

Trying to be consistent with the essence of the workings of evolution, we did not force our framework to contain oracles and ideal functions. In general, we did not make very restrictive decisions and even our model for evolution (the Meta-Model) is a very general one. The reader may have noticed that in Chapter 6, where we discussed the definition of evolvability in our framework, our focus was not on how evolvability can be defined on top of Meta-Model but on how it may be defined on top of an arbitrary model of evolution.[1] The Meta-Model was there only to assist in doing this and if one wants to use our framework, one does not need to limit herself to using our Meta-Model as is.

8.1 Future Work

Expanding on the work in Chapter 6, a direction for future work could be to examine more specializations of our framework. Certain specializations could

---

[1] The only hard assumptions we made here was to understand an evolutionary system as consisting of three parts, namely, life L, environment E, and evolutionary mechanism M.

actually give rise to interesting connections, such as the one that Valiant's framework does to computational learning. One may also examine other definitions of evolvability. Maybe certain definitions that are fitting to a certain type of applications.

Expanding on the work in Chapter 7, the most straightforward direction for future work is the actual use of our framework in order to analyse specific evolutionary systems, such as real biological systems or computational evolutionary systems.

Another interesting project would be to analyse more notions in a manner similar to the one that we analyzed evolvability here. For example, one may analyse the notion of sustainability, where, for example, an entity may be rendered sustainable, under certain conditions, if with non-negligible probability it can be sustained in the system for a respectable amount of time[2].

Taking our last point a bit further, a higher-level direction for future work could be to investigate which are all those notions, such as evolvability or sustainability, that may be of interest to biologists. There is the possibility actually that such work could find something new; if it fails to do that, it will at least make more concrete the connections between all those notions.

---

[2] This 'respectability' could be modelled for example by requiring the said amount of time to be superlinear in the inverse of the accuracy parameter $1/\epsilon$.

Appendix A

Appendix

A.1 Statistical Queries

In this section we introduce some frameworks of computational learning. In Section 3.1.2 we introduced the PAC learning framework. Here we introduce the closely related statistical query (SQ) framework and some specializations of it.

Kearns introduced the *statistical query* model in [Kea98] as a special case of PAC learning where the algorithm does not have direct access to the examples, as returned by the example oracle, but only has access to statistical queries on the concept f in question.

Precisely speaking, an algorithm has access to f only through the *statistical query oracle*. The latter, when passed a function $\psi : X \times \{-1, 1\} \to \{-1, 1\}$ as an argument, returns a value v such that $\left| \mathrm{E}_D[\psi(x, f(x))] - v \right| \leq \tau$ where $\tau \in [0, 1]$ is a real number called the tolerance of the query.

The definitions of *learnability* are identical to that of PAC learning, only that now the algorithm has access to the statistical query oracle instead of the example oracle. However, efficiency also requires that τ is bounded from below by $1/p(n, 1/\epsilon)$ for some polynomial p.

A.1.1 Specialized Statistical Queries

Let us start by defining the *correlational* statistical query (CSQ) model which is of particular interest in connection to Valiant's evolvability framework. In this model the function passed to the statistical query oracle is of the form $\psi(x, f(x)) \equiv \phi(x) \cdot f(x)$, where $\cdot$ is multiplication. That is, an algorithm may only query the correlation of some function $\phi : X \to \{-1, 1\}$ to the concept of interest f, where the correlation can be written as $\mathrm{E}_D[\psi(x, f(x))] =$

$\mathrm{E}_D[\phi(x)\cdot f(x)]$. In computational learning, f is the ideal concept and ϕ the current hypothesis.

Feldman [Fel09] also defined L-SQ, an intermediate concept between SQ and CSQ. The CSQ model, on one hand, restricts the SQ model by only allowing queries about the correlation between a concept and a hypothesis. The L-SQ model, on the other hand, restricts the SQ one by only allowing queries about the L-distance between a concept and a hypothesis. A description of the notion of L-distance and the definition of the L-SQ model follow.

In learning theory, a loss function is usually used to formalize the notion of "closeness". For functions with range Y, a loss function L is a non-negative mapping $L : Y \times Y \to \mathbb{R}^+$. The value of the function represents the loss suffered because of the distance between its two inputs. Feldman only considers admissible loss functions $L : \{-1, 1\} \times [-1, 1] \to \mathbb{R}^+$ as performance metrics.[1]

Feldman defined the class of L-SQ models where the statistical query oracle is restricted to answering questions about "closeness" according to L alone. For any admissible loss function L, we define an L-SQ to be a statistical query which may only query the value of $1-2\,\mathrm{E}_D[L(f(x), \phi(x))]/L(-1, 1)$ for some $\phi : X \to \{-1, 1\}$. The important quantity here is $\mathrm{E}_D[L(f(x), \phi(x))]$; the rest of the factors are there to normalize the value of this quantity.

A.2 Remarks in Connection to Certain Biological Assumptions

A.2.1 Assuming a Static Evolutionary Mechanism

Some skeptics might question the assumption of a static evolutionary mechanism. They may argue as follows. Biology tends to focus on a certain limited era along the existence of the universe. Specifically, it focuses on that latest part of it where there has been life on earth. But a more abstract discussion with philosophical concerns, like our discussion, can start from an earlier point. One where maybe there was no life or evolutionary mechanisms, and the latter only came about over time.

In other words, our current evolutionary mechanism may not have been ever-present and, as such, we may not consider it as static. In the context of biology this is not such an important concern since biology supports the

---

[1] He calls a loss function L *admissible* if it is efficiently computable, monotone, non-degenerate, $L(1, -1) = L(-1, 1)$ and $L(-1, -1) = L(1, 1) = 0$. More details in [Fel09].

existence of a certain static evolutionary mechanism along a huge timespan, starting from the time where only unicellular organisms were around. And that timespan is enough for most concerns of biology. It seems that this assumption may need to be questioned mainly only when the discussion reaches into philosophy.

The issue fits in the higher level discussion of the following section.

A.2.2 Using Evolvability to Question Neo-Darwinism

At times neo-Darwinism has been questioned on the basis of evolvability. In this section we criticize this kind of criticism. Let us start by noting a couple of examples where this has happened. According to the Irreducible Complexity [Pen01] argument of the proponents of Intelligent Design, certain biological systems are too complex to be the result of neo-Darwinism acting on "simpler" systems. This is one of the main arguments of the creationist community but no backing evidence has ever been brought up that was publishable in the scientific community.

Moreover, Aaronson [Aar11][2] writes "Interestingly, in a 1972 letter to Hao Wang (see [130, p. 192]), Kurt Gödel expressed his own doubts about evolution as follows: 'I believe that mechanism in biology is a prejudice of our time which will be disproved. In this case, one disproof, in my opinion, will consist in a mathematical theorem to the effect that the formation within geological time of a human body by the laws of physics (or any other laws of similar nature), starting from a random distribution of the elementary particles and the field, is as unlikely as the separation by chance of the atmosphere into its components.'"

Now, let us try to retrace the argument. Roughly, if evolvability was to be used to question neo-Darwinism, one would assert that the probability that modern-day animals, or, say, humans evolved from some initial state, say S, is very small. That is, human evolution, starting from S, implies the occurrence of a very unlikely event. Nevertheless, in general, in such arguments questioning neo-Darwinism the details of S tend to be only sloppily specified, if at all. Our main point here is that this latter step is pivotal and cannot be rushed.

For example, consider the following. Humans have evolved from unicellular organisms (or some other entity, but, for the sake of the argument, we will stick to unicellular organisms). However, it may not be the process of

---

[2] Aaronson included a section on Valiant's work [Val09] in a text of his [Aar11] wherein he argues why computational complexity may offer valuable insight into philosophical issues.

evolution that gave rise to the latter. As such, on one hand, we may need different means to determine the existence and form, and the corresponding probabilities, of those unicellular organisms that could give rise to humans.[3] On the other hand, we should not inquire the probability of the evolution of the human starting from some *random* initial state, as in Gödel's argument above; starting from the latter the probability of humans having evolved may be small, but this is not necessarily the case on the assumption that unicellular organisms existed far in the past.

Or consider this. We already know that humans exist. So maybe the question should be, what is the probability that humans have evolved from S, given that humans exist (and unicellular organisms too)?[4] The fact that humans exist may already be a very unlikely event, and we cannot argue that it is not correlated with the existence of other favorable conditions that were crucial in the emergence of humans. For example, what if neo-Darwinian forces are very common, i.e., systems where those forces could manifest themselves have often come into existence, but those other additional favorable conditions did not came about as often, and that is why humans did not evolve in the past? What if existence had its chance again and again (and in countless different locations in the universe) to give rise to humans, except that it was not lucky enough to have those favorable conditions around? It is not at all simple to rule out such a possibility.

The above should be enough to demonstrate that there tends to be a certain amount of naivity in arguments questioning neo-Darwinism on the basis of evolvability. The reason being that it tends to be problematic to determine the appropriate conditioning for the event that neo-Darwinism underlies evolution.

A.3 Pseudocode

In Section 5.1 we use some pseudocode to clarify some processes of the Meta-Model. Here we give a few notes on how to read this pseudocode.

We use pseudocode to define certain functions. The first line of each definition has an **IN** and an **OUT** statements defining the type of the input and the output, respectively, of the function. The second line has the name of the function and in parentheses the variable names used for its input components.

---

[3] Regardless of whether this is a matter that is under question in the context of biology.

[4] This concern here is related to the "anthropic principle" [HK13] that states that our observations are a result of our existence, or in other words, our understanding of our environment is filtered by the limitations and peculiarities of our own nature.

The next lines contain the body of the function. If we do not want to define the exact workings of the function yet, the body has the following form:

$$\ldots$$

While, if we do want to define the body we use some additional constructs.

We use $\leftarrow$ as the assignment operator; it assigns objects or values to variables of the corresponding type. We use **for each** as our looping construct; "**for each** $x \in X$" results in an iteration through all elements of X with x being the iterator. Lastly we use **if-then-else** as our conditional construct; consider "**if** c **then** b_1 **else** b_2": c must have a boolean value, if it is TRUE then b_1 is executed, otherwise b_2 is executed.

A.4 Details of Section 7.1

Consider the setting of the example in Section 7.1. Let us consider the quant-evolvability of an arbitrary finite entity class C. We can get the following trivial lower bound on the probability of the evolution of C.

Given that $|L^0| \geq 1$ we have that the probability that $|C|$ evolves is lower bounded by

$$\begin{aligned} b(0)p_m(1-d(1))^{|C|} \prod_{i=1}^{|C|-1} b(1)(1-p_m)(1-d(1))^{|C|-i} \geq \\ p_m(p_b(1-p_m)(1-p_d))^{|C|}. \end{aligned} \tag{A.1}$$

The quantity in (A.1) is the probability that the following happens. A white moth in L^0 gives birth to a black moth (happens with probability $b(0)p_m$) and the black moth survives at least $|C|$ timeslots (happens with probability $(1-d(1))^{|C|}$). At the next timeslot this black moth gives birth to another black moth that survives for at least $|C|-1$ timeslots. At the next timeslot one of black moths give birth to another black moth that survives at least $|C|-2$ timeslots, and so on. After $|C|$ timeslots we will have that $S^{|C|}.L$ contains at least $|C|$ black moths.

Bibliography

[Aar11] S. Aaronson. Why philosophers should care about computational complexity. *arXiv preprint arXiv:1108.1791*, 2011.

[Car10] Steven M. Carr. The mathematical theory of natural selection. `www.mun.ca/biology/scarr/2900_Natural_Selection.html`, 2010.

[Con83] J.H. Connell. On the prevalence and relative importance of interspecific competition: evidence from field experiments. *American Naturalist*, pages 661–696, 1983.

[Fel08] V. Feldman. Evolvability from learning algorithms. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 619–628. ACM, 2008.

[Fel09] V. Feldman. Robustness of evolvability. In *Proceedings of COLT*, pages 277–292, 2009.

[Fon13] Walter Fontana. Laboratory. `fontana.med.harvard.edu/www/Documents/Lab/research.signaling.htm`, 2013.

[FV08] V. Feldman and L.G. Valiant. The learning power of evolution. In *Proceedings of COLT*, pages 513–514, 2008.

[Gas07] Olivier Gascuel, editor. *Mathematics of Evolution and Phylogeny.* Oxford University Press, USA, 12 2007.

[Goo01] Brian Goodwin. *How the Leopard Changed Its Spots : The Evolution of Complexity.* Princeton University Press, 3 2001.

[GS07] Olivier Gascuel and Mike Steel, editors. *Reconstructing Evolution: New Mathematical and Computational Advances.* Oxford University Press, USA, first edition edition, 8 2007.

[HK13] Hans Halvorson and Helge Kragh. Cosmology and theology. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2013 edition, 2013.

[Kau93] Stuart A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, USA, 1 edition, 6 1993.

[Kea98] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

[KV94] M.J. Kearns and U.V. Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

[LOPA03] Richard E Lenski, Charles Ofria, Robert T Pennock, and Christoph Adami. The evolutionary origin of complex features. *Nature*, 423(6936):139–144, 2003.

[McS96] Daniel W. McShea. Metazoan complexity and evolution: is there a trend? *Evolution*, 50:477–492, 1996.

[Mic07] L. Michael. Evolving decision lists. Manuscript, 2007.

[Pen01] Robert T. Pennock, editor. *Intelligent Design, Creationism, and Its Critics: Philosophical, Theological, and Scientific Perspectives*. A Bradford Book, first edition, 12 2001.

[Rid04] Mark Ridley. *Evolution (Oxford Readers)*. Oxford University Press, USA, 2 edition, 2 2004.

[TH12] Nobuto Takeuchi and Paulien Hogeweg. Evolutionary dynamics of RNA-like replicator systems: a bioinformatic approach to the origin of life. *Physics of Life Reviews*, 9(3):219–263, 2012.

[Val84] L.G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Val09] L.G. Valiant. Evolvability. *Journal of the ACM (JACM)*, 56(1):3, 2009.