Evolvability

LESLIE G. VALIANT

Harvard University, Cambridge, Massachusetts

Abstract. Living organisms function in accordance with complex mechanisms that operate in different ways depending on conditions. Darwin's theory of evolution suggests that such mechanisms evolved through variation guided by natural selection. However, there has existed no theory that would explain quantitatively which mechanisms can so evolve in realistic population sizes within realistic time periods, and which are too complex. In this article, we suggest such a theory. We treat Darwinian evolution as a form of computational learning from examples in which the course of learning is influenced only by the aggregate fitness of the hypotheses on the examples, and not otherwise by specific examples. We formulate a notion of evolvability that distinguishes function classes that are evolvable with polynomially bounded resources from those that are not. We show that in a single stage of evolution monotone Boolean conjunctions and disjunctions are evolvable over the uniform distribution, while Boolean parity functions are not. We 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 the technical sense suggested above, each such target being rendered evolvable by the serendipitous combination of the environment and the outcomes of previous evolutionary stages.

Categories and Subject Descriptors: F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*Self-modifying machines (e.g., neural networks)*; I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms—*Analysis of algorithms*; I.2.6 [**Artificial Intelligence**]: Learning—*Concept learning*; J.3 [**Life and Medical Sciences**]: *Biology and genetics*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Evolvable, PAC learning, SQ learning

ACM Reference Format:

Valiant, L. G. 2009. Evolvability. J. ACM 56, 1, Article 3 (January 2009), 21 pages. DOI = 10.1145/1462153.1462156 http://doi.acm.org/10.1145/1462153.1462156

This work was supported by grants NSF-CCR-0310882, NSF-CCF-0432037 and NSF-CCF-04-27129 from the National Science Foundation.

Preliminary versions of this article appeared as Electronic Colloquium on Computational Complexity Report TR06-120 (2006) and in *Proceedings of the 32nd International Symposium on Mathematical Foundations of Computer Science*, Lecture Notes of Computer Science, vol. 4708, Springer-Verlag, Berlin, Germany, 2007, pp. 22–43.

Author's address: Harvard University, School of Engineering and Applied Sciences, Pierce Hall, 29 Oxford Street, Cambridge, MA 02138, e-mail: valiant@seas.harvard.edu.

1. *Introduction*

We address the problem of quantifying how complex mechanisms, such as those found in living cells, can evolve into existence without any need for unlikely events to occur. If evolution merely performed a random search, it would require exponential time, much too long to explain the complexity of existing biological structures. Darwin suggested selection as the critical controlling principle beyond variation. He also observed that the supposition that the eye could evolve would be " $\cdots$ absurd in the highest possible degree" were it not for the fact that eyes "vary ever so slightly" and might therefore evolve over time by selection [Darwin 1859]. In other words, a *necessary* condition for evolution to a specific target is the *existence* of an evolutionary path towards it consisting of small steps. But what are the conditions that hold in biology that are *sufficient* for such paths to be *taken* as routinely and efficiently as they apparently are?

This article describes a quantitative theory of the possibilities and limitations of what selection can achieve in speeding up the process of acquiring complex mechanisms beyond mere exhaustive search. In particular, we show that, in a defined quantitative sense, selection for a given beneficial behavior can provably support the evolution of certain specific classes of mechanisms, and provably not support that of certain other classes.

We approach this problem by viewing mechanisms from the viewpoint of the mathematical functions they realize. In particular the subject matter of the field of computational learning theory [Valiant 1984; Pitt and Valiant 1988; Blumer et al. 1989; Kearns and Valiant 1994; Kearns and Vazirani 1994] can be viewed as that of delineating limits on the complexity of functions that can be acquired with feasible resources without an explicit designer or programmer. A primary instance studied there is the acquisition of a recognition algorithm for a function given just positive and negative examples of it. The quantitative study of computational learning over the last two decades has shown that certain classes of recognition mechanisms can indeed be learned in a feasible amount of time, while others encounter apparently intractable computational impediments.

Our goal here is to give a quantitative theory of the evolution of mechanisms. What we formalize is concerned with four basic notions. First, since the biology of cells consists of thousands of proteins and operates with circuits with complex mechanisms, we seek mechanisms that can evaluate *many-argument functions*. This permits the behavior of circuits to vary in complex ways depending on the particular combination of values taken by a large number of input parameters. For example, such a function may determine the expression level of a protein in terms of the expression levels of all the other proteins. Second, any particular many-argument function has a measure of *performance* that is determined by the values of the function on inputs from a probability distribution over the conditions that arise. By applying the function to a variety of conditions, each one corresponding to an experience, the organism will enjoy a cumulative expected benefit that is determined by this performance. Third, for any function only a limited number of variants can be explored per generation, whether through mutations or recombination, since the organisms that can exist at any time have a *limited population*. Fourth, there is the requirement that mechanisms with significant improvements in performance evolve in a *limited number of generations*.

We show that our notion of evolvability is a restricted case of PAC learnability. This offers a unifying framework for the fields of evolution and cognition. The behavior of a biological organism is clearly affected both by the results of evolution and those of learning by the individual. Distinguishing between the effects of nature and nurture on behavior has proved problematic, and it will perhaps help to have a unifying viewpoint on them.

While evolvability as we define it is a form of learnability, it is a constrained form. In PAC learning [Valiant 1984], an update to a hypothesis may depend arbitrarily, as permitted by polynomial time computation, on the particular examples presented, on the values computed on them by the hypothesis, and on the values of the functions to be learned on those examples. In the more restricted Statistical Query (SQ) model of Kearns [1998] the updates can depend only on the result of statistical queries on the distribution of examples. However, these queries can ask about the percentage of examples that have certain syntactic properties beside being positive or negative, such as the number of bits that are ones in their description, and hence the course of learning may depend explicitly on these syntactic descriptions. In evolution, we assume that the updates depend *only* on the aggregate performance of the competing hypotheses on a distribution of examples or experiences, and in no additional way on the syntactic descriptions of the examples. This restriction reflects the idea that the relationship between the genotype and phenotype may be extremely complicated, and the evolution algorithm does not understand it. We shall observe that the function classes that are evolvable, in our sense, form a subset of those that are SQ learnable.

As far as the evolvability of specific classes of functions, we have both positive and negative results. On the positive side we show that the classes of monotone Boolean conjunctions and disjunctions are evolvable over the uniform distribution of inputs for the most natural representation of these functions. On the negative side, we observe that the class of Boolean parity functions is not evolvable over the uniform distribution, as it is not SQ learnable. Since the latter class is known to be learnable, we can conclude that evolvability is more constrained than learnability.

Our intention is to leave little doubt that functions in classes that are provably evolvable in the defined sense, do correspond to mechanisms that can logically evolve into existence over realistic time periods and within realistic populations, without any need for combinatorially unlikely events to occur. Previous quantitative theories of evolution had aims other than that of quantifying the complexity of the mechanisms that evolved. The major classical thrust has been the analysis of the dynamics of populations [Fisher 1930; Wright 1968; Roff 1997; Bürger 2000]. The immediate goal of these analyses has been the prediction of the outcome of competition among different populations. Game theory has been one of the approaches followed [Maynard-Smith 1982]. A more recent development is the study of evolutionary algorithms [Bäck et al. 1997; Wegener 2001], a field in which the goal is to develop good computer algorithms inspired by evolution, usually for optimization, but not necessarily those that model biological evolution. For example, these algorithms may choose mutations depending on the current inputs or experiences, may act on exponentially small increases in performance, and may be forced to start from a fixed configuration. We note also that the term *evolvability* has been used in a further different sense, that of measuring the intrinsic capacity of genomes to produce variants [Wagner and Altenberg 1996].

We observe that these and other previous discussions of evolution here emphasized competition and survival. Such discussions have not yielded quantitative explanations of which mechanisms can evolve using only feasible resources, and which cannot. Our purpose in introducing our target-oriented formulation is to address this specific question.

2. *Many-Argument Functions*

The structures or circuits that are the constituents of living cells have to respond appropriately to wide variations in the internal and external conditions. We shall represent the conditions as the values of a number of variables $x_1, \ldots, x_n$, each of which may correspond, for example, to the output of some previously existing circuit. The responses that are desirable for an organism under the various combinations of values of the variables $x_1, \ldots, x_n$ we view as the values of an *ideal* function $f(x_1, \ldots, x_n)$. This f will have a low value in circumstances when such a low value is the most beneficial, and a high value in those other circumstances when a high value is the most beneficial. It will be beneficial to evolve a circuit that behaves as closely as possible to this f. For simplicity, we shall consider the variables x_i and the functions f to take just two possible values, a low value of -1, and a high value of $+1$. One can, for example, think of x_i having value $+1$ or -1 according to whether the ith protein is being expressed or not, and f having value $+1$ or -1 according to whether a further protein is being expressed. Then, f is an ideal function if for each combination of the conditions represented by the x_i the value of f is the best one for the organism.

We shall consider in this article two particular function classes. The first, the *parity* functions, will be shown in this article not to be evolvable. The second, *conjunctions*, will be shown to be evolvable in the monotone case over the uniform distribution.

A parity function is odd or even. An odd (even) parity function f over $x_1, \ldots, x_n$ has value $+1$ if and only if an odd (even) number of the variables in a subset S of the variables have value $+1$. For example, an odd parity function over $\{x_1, x_2, x_3, x_4\}$ with $S = \{x_1, x_3, x_4\}$ has value $+1$ if $x_1 = x_2 = x_3 = x_4 = +1$, and value -1 if $x_1 = x_2 = x_3 = +1$ and $x_4 = -1$.

We shall show that for evolving arbitrary parity functions over n variables, that is for an arbitrary *unknown* subset S, either the number of generations or the population size of the generations would need to be exponentially large in terms of n.

In contrast, we shall also show that there are classes with similarly substantial structure that are evolvable in a strong sense. An example of such a class is that of monotone conjunctions, defined as conjunctions of a subset S of the literals $\{x_1, x_2, \ldots, x_n\}$. An example of a conjunction is the function that is true if and only if $x_1 = +1$, and $x_4 = +1$. We abbreviate this function as x_1x_4.

We denote by X_n the set of all 2^n combinations of values that the variables $x_1, \ldots, x_n$ can take. For both of the above defined function classes, the set S is unknown to the evolution algorithm and the challenge for it is to approximate it from among the more than polynomially many possibilities. We define D_n to be a probability distribution over X_n that describes the relative frequency with which the various combinations of variable values for $x_1, \ldots, x_n$ occur in the context of the organism. Evolution algorithms that work for all distributions would be particularly compelling.

Definition 2.1. The *performance* of function $r : X_n \to \{-1, 1\}$ with respect to ideal function $f : X_n \to \{-1, 1\}$ for probability distribution D_n over X_n is

$$\mathrm{Perf}_f(r, D_n) = \sum_{x \in X_n} f(x)r(x)D_n(x).$$

The performance is simply a measure of the correlation between the ideal function f and a hypothesis r we have at hand. The value will always be a real number in the range $[-1, 1]$. It will have value 1 if f is identical to r on points with nonzero probability in D_n. It will have value -1 if there is perfect anti-correlation on these points.

The interpretation is that every time the organism has an experience in the form of a set of values $x \in X_n$ it will undergo a benefit amounting to $+1$ if its circuit r agrees with the ideal f on that x, or a penalty -1 if r disagrees. Over a sequence of life experiences (i.e., different points in X_n) the total of all the benefits and penalties will be accumulated. Organisms or groups for which this total is high will be selected preferentially to survive over organisms or groups with lower such totals.

The performance $Perf_f(r, D_n)$ may be viewed as a fitness landscape over the genomes r. Our analysis discusses the viability of this landscape for evolution in terms of the nature of the ideal function f and the distribution D_n of inputs for f. Instead of speculating on the nature of fitness landscapes that may be encountered in the world, we instead discuss which ideal functions give rise to landscapes that allow them to evolve.

An organism or group will be able to test the performance of a function r by sampling a limited set $Y \subseteq X_n$ of size $s(n)$ of inputs or experiences from D_n, where, for simplicity, we assume that the Y are chosen independently for the various r. These experiences may be thought of as corresponding to one or more per organism, so that $s(n)$ certainly upper bounds the population size.

Definition 2.2. For a positive integer s, ideal function $f : X_n \to \{-1, 1\}$ and probability distribution D_n over X_n the *empirical performance* $Perf_f(r, D_n, s)$ of function $r : X_n \to \{-1, 1\}$ is a random variable that makes s selections independently with replacement according to D_n and for the multiset Y so obtained takes value

$$s^{-1} \sum_{x \in Y} f(x)r(x).$$

In our basic definition of evolvability, we insist that evolution be able to proceed from any starting point. Otherwise, if some reinitialization process were permitted, then proceeding to the reinitialized state from another state might incur an arbitrarily large decrease in performance.

The evolution algorithm for monotone conjunctions that we describe in detail in Section 5 behaves as follows. The learning of a target function x_1x_4 will be achieved by an algorithm that maintains a conjunction of a number of literals. Clearly, the aim is to evolve the function x_1x_4, which has performance $+1$ since it is identical with the target, and is the only conjunction with that performance. The mutations will consist of *adding or deleting a single literal* for the current conjunction, or *swapping one literal for another*. Since there are then about n^2 possible mutations

at each step it is feasible to explore the space of all mutations with a population of (polynomial) size, namely n^2.

3. *Definition of Evolvability*

Given the existence of an ideal function f the question we ask is whether it is possible to evolve a circuit for a function r that closely approximates f. Roughly, we want to say that a class C of ideal functions f is evolvable if any f in the class C satisfies two conditions. (i) From any starting function r_0 the sequence of functions $r_0 \Rightarrow r_1 \Rightarrow r_2 \Rightarrow r_3 \Rightarrow \cdots$ will be such that r_i will follow from r_{i-1} as a result of a single step of mutation and selection in a moderate size population, and (ii) after a moderate number i of steps r_i will have a performance value significantly higher than the performance of r_0, so that it is detectable after a moderate number of experiences. The conditions will be sufficient for evolution to start from any function and progress towards f, predictably and inexorably.

While the ideal function may be viewed as an abstract mathematical function, the hypothesis r needs to be represented concretely in the organism and should be viewed as a *representation* of a function. We generally consider C to be a class of ideal functions and R a class of representations of functions from which the organism will choose an r to approximate the f from C. We assume that the representation R is *polynomial evaluatable* in the sense that there is a polynomial $u(n)$ such that given the description of an r in R and an input x from X_n, the value of $r(x)$ can be computed in $u(|r|)$ steps. This reflects the assumption that the processes of biology can be simulated in polynomial time on a computer. (We note, however, that, in this article, this assumption does not arise anywhere except when relating the evolvable class to learnable classes for which corresponding conditions are imposed.) For brevity, and where it introduces no confusion, we shall denote by r both the representation as well as the function that that representation computes. We denote by C_n, R_n, and D_n, the restrictions of C, R, and D to n variables, but sometimes omit these distinctions where the meaning is clear. Also, we shall denote by ε the error parameter of the evolution, which describes how close to optimality the performance of the evolved representation has to be. We shall be prepared to expend resources that are polynomial in n, the number of arguments of the ideal function, and also in $1/\varepsilon$. Hence, our resource bounds will be polynomial functions, such as $p(n, 1/\varepsilon)$ in the definition below. (To be more precise, n here should be interpreted as the maximum of the number of arguments and the size of the smallest representation of an ideal target function, but we will, for simplicity assume here that the representations needed are of size polynomial in the number of variables, which removes the need for this distinction. However, we will imply in all our polynomial bounds in this article where n occurs this more precise meaning that also applies when the number of variables grows more slowly.)

Definition 3.1. For a polynomial $p(\cdot\ ,\ \cdot)$ and a representation class R a *p-neighborhood* N on R is a pair M_1, M_2 of randomized polynomial time Turing machines such that on input the numbers n (in unary) and $\lceil 1/\varepsilon \rceil$ and a representation $r \in R_n$ act as follows: M_1 outputs all the members of a set $Neigh_N(r, \varepsilon) \subseteq R_n$, that contains r and may depend on random coin tosses of M_1, and has size at most $p(n, 1/\varepsilon)$. If M_2 is then run on this output of M_1, it in turn outputs one member

of $Neigh_N(r, \varepsilon)$, with member r_1 being output with a probability $Pr_N(r, r_1) \geq 1/p(n, 1/\varepsilon)$.

The interpretation here is that for each genome the number of variants, determined by M_1, that can be searched effectively is not unlimited, because the population at any time is not unlimited, but is polynomial bounded. But a significant number of experiences with each variant, generated by M_2, must be available so that differences in performance can be detected reliably. One possible implementation, clearly, is to regard R as the set of possible genomes, restrict mutations to a fixed constant number of base pairs, and regard the genome length as a polynomial in the relevant n. We consider exponentially many such variants to be impractical, while modest polynomial bounds such as n or n^2 are feasible. As in other areas of algorithmic analysis natural polynomially bounded processes usually have reasonably modest polynomial bounds, and hence such results are meaningful [Garey and Johnson 1979; Papadimitriou 1994]. The theory, as presented here, aims to distinguish between polynomial and exponential resources, insisting as it does that population sizes, numbers of generations, and numbers of computational steps all have to be upper bounded by a polynomial in the number of variables on which a circuit depends, and in the inverse of the error. Clearly, using more careful analysis finer distinctions can also be made. We note that estimates of actual mutation rates in various organisms are available [Drake et al. 1998; Kimura 1968; Kumar and Subramanian 2002].

Definition 3.2. For error parameter ε, positive integers n and s, an ideal function $f \in C_n$, a representation class R with $p(n, 1/\varepsilon)$-neighborhood N on R, a distribution D, a representation $r \in R_n$ and a real number t, the *mutator* $Mu(f, p(n, 1/\varepsilon), R, N, D, s, r, t)$ is a random variable that on input $r \in R_n$ takes a value $r_1 \in R_n$ determined as follows: For each $r_1 \in Neigh_N(r, \varepsilon)$ it first computes an empirical value of $v(r_1) = Perf_f(r_1, D_n, s)$. Let *Bene* be the set $\{r_1 \mid v(r_1) \geq v(r) + t\}$ and *Neut* be the set difference $\{r_1 \mid v(r_1) \geq v(r) - t\} - Bene$. Then

(i) if $Bene \neq \phi$ then output $r_1 \in Bene$ with probability

$$Pr_N(r, r_1) / \sum_{r_1 \in Bene} Pr_N(r, r_1)$$

(ii) if $Bene = \phi$ then output an $r_1 \in Neut$, the probability of a specific r_1 being

$$Pr_N(r, r_1) / \sum_{r_1 \in Neut} Pr_N(r, r_1).$$

In this definition, a distinction is made between beneficial and neutral mutations as revealed by a set of s experiences. In the former, the empirical performance after the mutation exceeds that of the current representation r by an additive tolerance of at least t, a quantity which will, in general, be large enough, in particular some inverse polynomial, that it can be reliably distinguished from a zero increase in performance. In neutral mutations, no significant increase in performance is expected, but it is expected that the performance is not worse than that of the current r by more than t. If some beneficial mutations are available, one is chosen in accordance with the relative probabilities of their generation by N as allowed by machine M_2 in Definition 3.1. If none is available, then one of the neutral mutations is taken

in accordance with the relative probabilities of *their* generation by N. Since in Definition 3.1, we insist that $r \in Neigh_N(r, \varepsilon)$, r will always be empirically neutral, by definition, and hence *Neut* will be nonempty.

Definition 3.3. For a mutator $Mu(f, p(n, 1/\varepsilon), R, N, D, s, r, t)$ a *t-evolution step* on input $r_1 \in R_n$ is the random variable $r_2 = Mu(f, p(n, 1/\varepsilon), R, N, D, \mathrm{s}, r_1, t)$. We then say $r_1 \rightarrow r_2$ or $r_2 \leftarrow Evolve(f, p(n, 1/\varepsilon), R, N, D_n, s, r_1, t)$.

We say that polynomials $tl(x, y)$ and $tu(x, y)$ are *polynomially related* if for some $\eta > 1$ for all x, y $(0 < x, y < 1)$ $(tu(x, y))^\eta \leq tl(x, y) \leq tu(x, y)$. We now define an evolution sequence as a sequence of t-evolution steps where the t at each step is bounded between two polynomially related quantities $t\ell(1/n, \varepsilon)$, $tu(1/n, \varepsilon)$ and computable in polynomial time by a Turing machine T that takes $r \in R$, n and ε as inputs.

Definition 3.4. For a mutator $Mu(f, p(n, 1/\varepsilon), R, N, D, s, r, t)$ a $(t\ell, tu)$-*evolution sequence* for $r_1 \in R_n$ is a random variable that takes as values sequences $r_1, r_2, r_3, \ldots$ such that for all i $r_i \leftarrow Evolve(f, p(n, 1/\varepsilon), R, N, D, s, r_{i-1}, t_i)$, where $t\ell(1/n, \varepsilon) \leq t_i \leq tu(1/n, \varepsilon)$, tl and tu are polynomially related polynomials, and t_i is the output of a TM T on input r_{i-1}, n and ε.

We shall find that if we want to evolve to performance very close to one, say $1 - \varepsilon$, we shall need numbers of experiences s or numbers of generations g that grow inversely with ε, and tolerances t that diminish with ε. We therefore regard these as functions of n and ε, and denote them by $s(n, 1/\varepsilon)$, $g(n, 1/\varepsilon)$, $tl(1/n, \varepsilon)$ and $tu(1/n, \varepsilon)$.

Definition 3.5. For polynomials $p(n, 1/\varepsilon)$, $s(n, 1/\varepsilon)$, $t\ell(1/n, \varepsilon)$ and $tu(1/n, \varepsilon)$, a representation class R and $p(n, 1/\varepsilon)$-neighborhood N on R, the class C is $(t\ell, tu)$-evolvable by $(p(n, 1/\varepsilon), R, N, s(n, 1/\varepsilon))$ *over distribution* D if there is a polynomial $g(n, 1/\varepsilon)$ and a Turing machine T, which computes a tolerance bounded between tl and tu, such that for every positive integer n, every $f \in C_n$, every $\varepsilon > 0$, and every $r_0 \in R_n$ it is the case that with probability greater than $1 - \varepsilon$, a (tl, tu)-evolution sequence $r_0, r_1, r_2, \ldots$, where $r_i \leftarrow Evolve(f, p(n, 1/\varepsilon), R, N, D_n, s(n, 1/\varepsilon), r_{i-1}, T(r_{i-1}, n, \varepsilon))$, will have $Perf_f(r_{g(n,1/\varepsilon)}, D_n) > 1 - \varepsilon$.

The polynomial $g(n, 1/\varepsilon)$, the generation polynomial, upper bounds the number of generations needed for the evolution process.

Definition 3.6. A class C is *evolvable by* $(p(n, 1/\varepsilon), R, N, s(n, 1/\varepsilon))$ *over* D iff for some pair of polynomially related polynomials $t\ell$, tu, C is (tl, tu)-evolvable by $(p(n, 1/\varepsilon), R, N, s(n, 1/\varepsilon))$ over D.

Definition 3.7. A class C is *evolvable by* R *over* D iff for some polynomials $p(n, 1/\varepsilon)$ and $s(n, 1/\varepsilon)$, and some $p(n, 1/\varepsilon)$-neighborhood N on R, C is evolvable by $(p(n, 1/\varepsilon), R, N, s(n, 1/\varepsilon))$ over D.

Definition 3.8. A class C is *evolvable over* D if for some R it is evolvable by R over D.

Definition 3.9. A class C is *evolvable* if it is evolvable over all D.

Our definition of evolvability is closely related to that of learnability [Valiant 1984; Kearns and Vazirani 1994], but includes the extra ingredients that each step of learning (i) chooses from a polynomial size set of hypotheses, (ii) tolerates at most a small decrease in performance, and further (iii) the choice of the next hypothesis from among the candidate hypotheses is made on the basis of their aggregate performance on inputs, and not differentially according to the values of the various inputs.

PROPOSITION 3.10. *If C is evolvable by R over D, then C is learnable by R over D. In particular, if C is evolvable by R then C is learnable by R.*

PROOF. If C is evolvable over D, then, by definition, for some polynomials $p(n, 1/\varepsilon)$, $s(n, 1/\varepsilon)$, $g(n, 1/\varepsilon)$, $t\ell(1/n, \varepsilon)$ and $tu(1/n, \varepsilon)$, some representation R and some $p(n, 1/\varepsilon)$-neighborhood N on R, C is $(t\ell, tu)$-evolvable by $(p(n, 1/\varepsilon), R, N, s(n, 1/\varepsilon))$ over distribution D with generation polynomial $g(n, 1/\varepsilon)$. The main observation here is that we can replicate this evolution algorithm exactly in terms of the PAC learning framework. At each stage, the evolution algorithm takes fixed size samples of $s(n, 1/\varepsilon)$ labeled examples from the distribution, computes for its current hypothesis the empirical performance, and from that generates the next hypothesis in a polynomially bounded fashion. But computing this performance is equivalent to computing the fraction of examples on which the hypothesis predicts correctly. Hence, the access required to examples is that of random labeled examples from D, and every step is a polynomial-time computational process. All this is permitted within the PAC model. Also the final hypothesis of the evolution model satisfies the requirements of the learning model since it ensures that the performance is at least $1 - \varepsilon$, and hence accurate on at least $1 - \varepsilon/2$ of D. □

We can strengthen the above statement by observing that evolvability implies learnability in the more restricted sense of statistical queries defined by Kearns [1998]. In that model oracles provide not individual examples but estimates, to within arbitrary inverse polynomial additive error, of the fraction of examples that satisfy polynomially evaluatable properties. For simulating an evolution algorithm we need to be able to simulate the evaluation of *Bene*, or in other words for pairs r, r_1 the probability that the empirical performances for samples of a certain size s give $v(r_1) \geq v(r) + t$. This can be done by asking the SQ oracle to estimate the probabilities of the four events that: $(r = r_1 = f)$, $(r = f, r, \neq f)$, $(r \neq f, r_1 = f)$ and $(r \neq f, r_1 \neq f)$. These need to be simulated only to sufficiently small inverse polynomial accuracy

PROPOSITION 3.11. *If C is evolvable by R over D, then it is efficiently learnable from statistical queries using R over D.*

Evolvability for all D is a very strong and desirable notion. As mentioned previously, it guarantees evolution independent of any assumptions about the distribution. It also means that evolution can continue even if the D changes. Of course, a change in D can cause a reduction in the value of Perf for any one r, and hence may set back the progress of evolution. However, the process of finding improvements with respect to whatever the current D is will continue. It remains an open problem as to whether such distribution-free evolution is possible for a significant class of functions.

Note also that the representation class R may represent a class of functions that differs from C. For example, an R richer than C may be helpful. Alternatively, a weaker class may still produce good enough approximations and may have better properties for evolution. In general, if we wish to identify or emphasize the class R that supports an evolution algorithm, we say that *C is evolvable by R for D*, or *C is evolvable by R*.

The purpose of the main definition above of evolvability is to capture the notion of evolution towards a target under stable conditions. In biological evolution, other phenomena are involved also, and, we believe, many of these can be discussed in the language of our formalism by appropriate variants. One restriction is *evolvability with initialization*. In that case, in Definition 3.5, instead of requiring convergence from any starting point $r_0 \in R_n$, we require only that there is convergence from one fixed starting point r_0 for all targets $f \in C_n$. The more general definition given is more robust, allowing for successive phases of evolution, each one with a different target ideal function, for example. The evolved representation for one phase can then serve as the starting representation for the next, without a decrease in performance at any step. In evolution with initialization, the steps of going from the end of one phase to a reinitialized new state may suffer an arbitrary performance decrease. In our definition of evolvability, we seek to avoid allowing any mechanism that would provide for such initialization by a back door. We therefore insist that the tolerance be bounded between two polynomially related functions. Allowing the tolerance to be arbitrarily large would allow initialization in one step via an arbitrarily large drop in performance.

Another variation is *variable population evolution.* In this, the sample size s may vary. In particular, if it is made small then random variations in the empirical performance may make a low performance mutation appear as neutral or even beneficial, and be adopted. This permits reinitializations, for example, for a subsequent phase of evolution with initialization. In biology evolution in small populations is believed to play a special role.

A further variant is *evolvability with optimization*. Here, we insist that, in Definition 3.2, the representation r_1 selected is any one with empirical performance within tolerance t of the best empirical performance in $Neigh_N(r)$. However, it is easy to see that this variant is no more powerful than the main definition. One can simulate the search for the best representation, as required in one step of the optimized evolution, in no more than $6/t$ basic steps of looking for a representation with an empirical additive improvement of at least $t/2$, each step using a new sample. (Note that the actual performance can increase cumulatively by at most 2. Using the Hoeffding Bound (Fact 2) one can show that the cumulative empirical performance increase on the different samples can be limited to 3 with overwhelming probability.) For this simulation, we change the representation to $i \cdot r^M \cdot r^P$, where $i \leq 6/t$ is an integer denoting which basic step we are in, $r^M \in R$ is the representation that generates the mutations (using the M_1 of Definition 3.1) for each of the up to $6/t$ basic steps, and $r^P \in R$ is the one with best performance found so far. (In other words, r^P is the function this representation is computing, but the representation also has a memory of r^M from which it can generate new mutations in R, that may not be generatable from r^P alone.) After $i = 6/t$ basic steps, the final r^P is adopted as the starting r^M and r^P of the next step of the optimized evolution. Note that the constructed representation in this reduction is a *redundant* representation in the sense that there are many representations that correspond to the same function r^P.

It illustrates the power of storing history, namely R^M, in addition to the active part, R^P.

PROPOSITION 3.12. *If C is evolvable with optimization over D, then C is evolvable over D. If C is evolvable with initialization and optimization over D, then C is evolvable with initialization over D.*

A simpler variant is that of *fixed-tolerance* evolvability, obtained if the bounds tl, tu on the tolerance are the same.

We note that the aspect of evolution that is outside the control of the evolution algorithm itself is the population size. Thus, evolvability guarantees inexorable convergence only if the population is appropriate. Our algorithms require only that the population, as represented by s, be large enough. The variable population variant defined earlier permits schedules of varying population sizes.

4. *Limits to Evolvability*

The obvious question arises as to whether the converse of Proposition 3.1 holds: does learnability imply evolvability? Our next observation answers this in the negative, saying as it does that for a certain function class there is a distribution that defeats all combinations of representations and neighborhoods.

We define Lin_n to be the set of odd parity functions $f(x_1, \ldots, x_n)$ over $\{-1, 1\}^n$. Each such f corresponds to some subset of the variables $x_{i[1]}, \ldots, x_{i[k]} \in \{x_1, \ldots, x_n\}$. The function f has value 1 if and only if an odd number of the variables $\{x_{i[1]}, \ldots, x_{i[k]}\}$ have value 1. Clearly, there are 2^n functions in Lin_n. We define U to be the uniform distribution over $\{-1, 1\}^n$. We note that the functions in Lin are easy to compute, and further the class is known to be learnable not only for U but for all distributions [Fischer and Simon 1992; Helmbold et al. 1992].

PROPOSITION 4.1. *Lin is not evolvable for U by any representation R.*

PROOF. Kearns [1998] shows that Lin is not efficiently learnable from statistical queries over U using any representation. The result then follows from Proposition 3.2 above. □

The class Lin may appear to be biologically unnatural. That is exactly the prediction of our theory, which asserts that evolution cannot be based on the evolvability of such a class.

An important class of functions that is known to be learnable is that of linear halfspaces $\{\mathbf{a}.\mathbf{x} \leq b \mid \mathbf{a} \in \mathbf{R}^n, b \in \mathbf{R}\}$ in n-dimensional space $\mathbf{R}^n$. This class is learnable for all distributions by the natural representation of linear halfspaces if the coefficients $\mathbf{a}$, b are represented as rational numbers with n digits of accuracy, by virtue of the existence of polynomial time algorithms for linear programming [Blumer et al. 1989]. However, if both the class and its representation is restricted to {0,1} coefficients, then we have the following.

PROPOSITION 4.2. *If C is the class of Boolean Threshold Functions* $\{\mathbf{a}.\mathbf{x} \leq b \mid \mathbf{a} \in \{0, 1\}^n, b \in \mathbf{R}\}$ *in* $\mathbf{R}^n$ *and R is the given representation of it, then C is not evolvable by R, unless NP = RP.*

PROOF. In Pitt and Valiant [1988], it is shown that this class is not learnable by its natural representation unless NP = RP. (The proof there shows that an

NP-complete problem, integer programming, can be mapped to instances of learning Boolean threshold functions for a certain distribution to accuracy better than $1/n$.) The result follows from Proposition 3.1 above. □

There appear to be at least four impediments that can be identified to evolvability in our sense, the first three of which derive from general impediments to learnability, while the last is particular to evolvability: (i) A purely information theoretic impediment [Ehrenfeucht et al. 1989]: the complexity of the mechanism that is to evolve exceeds the number of experiences, (ii) A representational limit such as Proposition 4.2 above, where learnability by a fixed representation would imply solving a computational problem that is believed to be hard, (iii) An intrinsic complexity limitation [Kearns and Valiant 1994]: the function class is so extensive that learning it by *any* representation would imply an efficient algorithm for a problem believed to be hard to compute, (iv) Limits such as Proposition 4.1 above, that show that for information theoretic reasons evolvability cannot proceed because no empirical test of a polynomial number of hypotheses in a neighborhood can guarantee sufficient convergence in performance. Note that impediments (i) and (iv) are absolute, requiring no unproven computational assumptions.

5. *Some Provably Evolvable Structures*

We now describe some basic classes of Boolean functions and distributions that are provably evolvable. Here, disjunction or Boolean "or" is denoted by $+$, conjunction or Boolean "and" by the multiplication sign, and Boolean negation of a variable x_i by x_i'. In general, we shall have n variables $x_1, \ldots, x_n$. A *q-disjunction* is a disjunction of $k \leq q$ of the n variables or their negations, while a *q-conjunction* is a conjunction of $k \leq q$ of the n variables or their negations. Thus a q-disjunction is $y_{i[1]} + \cdots + y_{i[k]}$ where $1 \leq i[1], \ldots, i[k] \leq n$ and $y_{i[j]} \in \{x_1, \ldots, x_n, x_1', \ldots, x_n'\}$, and a q-conjunction is $y_{i[1]} \ldots y_{i[k]}$. The uniform distribution over $\{-1, +1\}$ will be denoted again by U. A conjunction or disjunction is *monotone* if it contains no negated literals. We note that Ros (Section B2.8 in Bäck et al. [1997], Ros [1993]) has analyzed evolutionary algorithms for learning conjunctions and disjunctions. However, a step of his algorithm is allowed to depend not just on the value of his current hypothesis on an input, but on more detailed information such as the number of bits on which the hypothesis and input differ. Such dependence on the input condition we consider unrealistic for evolution, and is outside our model. With regard to the literature on evolutionary algorithms [Wegener 2001], we also note that the functions being evolved there are often real rather than Boolean valued, and that provides more feedback to the process.

Fact 1. Over the uniform distribution U for any conjunction $Pr_U(y_{i[1]} \cdots y_{i[k]} = 1) = 2^{-k}$ and for any disjunction $Pr_U(y_{i[1]} + \cdots + y_{i[k]} = 1) = 1 - 2^{-k}$.

For our probabilistic arguments below, it will be sufficient to appeal to the following:

Fact 2 [*Hoeffding* 1963]. The probability that the mean of s independent random variables each taking values in the range $[a, b]$ is greater than or less than the mean of their expectations by more than δ is at most $\exp(-2s\delta^2/(b-a)^2)$, where $\exp(x)$ denotes e^x.

Fact 3 (*Coupon Collector's Problem*). Suppose that there is a bucket of n balls and M is a subset of m of them. Then after $j = CC(n, m, \eta) = n(\log_e m + \log_e(1/\eta))$ samples with replacement the probability that some member of the chosen set M has been missed is less than η.

PROOF. This probability is upper bounded by $m(1-1/n)^j < m(1-1/n)^{jn/n} = me^{-j/n} < \eta$. □

We note that an evolution algorithm for a representation R needs to have defined at each step (i) the neighborhood N, (ii) the tolerance t and (iii) the sample sizes, so that the mutator random variable can be evaluated at that step.

THEOREM 5.1. *Monotone conjunctions and disjunctions are evolvable over the uniform distribution for their natural representations.*

PROOF. We first note that for our definition of evolvability it is sometimes advantageous for a local search procedure to introduce literals that do not appear in the ideal function f. For example, suppose $f = x_1x_2x_3$ and we start with a hypothesis 1, the conjunction of zero literals. Then, the hypothesis will disagree with f on $7/8$ of the distribution, and the introduction of the literal x_4 will be an improvement, reducing this probability from $7/8$ to $1/2$.

If we are evolving to accuracy ε and have n variables, we let $q = \lceil \log_2(dn/\varepsilon) \rceil$ for some constant d. We choose the effective representation class R to be monotone q-conjunctions.

We first assume that both the ideal function f and the initial representation r_0 have at most q literals. We denote by r^+ and r^- the sets of conjunctions consisting of the literals of r with one literal added, and with one taken away, respectively. In the case that r has the maximum number q of literals then r^+ is empty. In the case that $|r| = 0$, r^- is empty. Also we define r^{+-} to be the conjunctions consisting of the literals in r with one further literal added and then one literal taken away. Clearly, $r \in r^{+-}$. We then choose the neighborhood structure N to be such that

$$Neigh_N(r, \varepsilon) = r^+ \cup r^- \cup r^{+-}.$$

Finally, r and the members of r^+ and r^- will each have equal probabilities, so that their total is $1/2$, while the remaining members of r^{+-} will also have equal probabilities, again totaling $1/2$. Clearly, N is a $p(n, 1/\varepsilon)$-neighborhood structure where $p(n, 1/\varepsilon) = O(n^2)$.

The construction will ensure that every mutation in N either causes an improvement in performance of at least 2^{-2q} or causes no improvement (and a possible degradation.) We choose the tolerance $t(1/n, \varepsilon) = 2^{-2q-1}$ and the number of samples $s(n, 1/\varepsilon) = t^{-3}$. It will then follow that the empirical test will, except with exponentially small probability, correctly identify an available mutation that has true improvement $2t$, distinguishing it from one that gives no improvement in performance. This can be seen by substituting $a = -1, b = 1, \delta = t$ and $s = \delta^{-3}$ in the Hoeffding Bound above to obtain that the probability of s trials each with expected improvement 2δ will produce a mean improvement of less than $t = \delta$ is at most $\exp(-2s\delta^2/(b-a)^2) = \exp(-(dn/\varepsilon)^2)$. A similar argument also shows that the same test will not mistakenly classify a mutation with no performance increase with one with an increase of $2t$. In the same way, the same tolerance will distinguish a mutation with a nonnegative performance increase from one whose performance decreases by at least $2t$.

In a run of the evolution algorithm, there will be $g(n, 1/\varepsilon)$ stages, and in each stage up to $p(n, 1/\varepsilon)$ mutations will be tested, where g and p are polynomials. We will want that in all $p(n, 1/\varepsilon)g(n, 1/\varepsilon)$ empirical tests the probability of even one failure to make such a distinction be less than $\varepsilon/2$. But $p(n, 1/\varepsilon)g(n, 1/\varepsilon)\exp(-(dn/\varepsilon)^2) < \varepsilon/2$ for all n, ε for a suitable constant d.

Suppose that W is the true set of m literals in the ideal conjunction, and that the current hypothesis r is the conjunction of a set V of k literals. We claim that:

CLAIM 1. *For* $(a) - (g)$ *suppose that* $k \leq q$. *Then*

(a) *If* $k < q$, *then adding to* r *any literal* z *in* $W - V$ *will increase the performance of* r *by at least* 2^{1-q}.

(b) *Removing from* r *any literal* z *in* $V \cap W$ *will decrease the performance of* r *by at least* 2^{1-q}.

(c) *Adding to* r *a literal in* $W - V$ *and removing from* r *a literal in* $V - W$ *will increase the performance by at least* 2^{-q-m}.

(d) *Adding to* r *some literal not in* W, *and removing from* r *a literal in* $V \cap W$ *will decrease the performance of* r *by at least* 2^{-q-m}.

(e) *Adding to* r *a literal in* $W - V$ *and removing from* r *a literal in* $V \cap W$ *will leave the performance unchanged, as will also adding to* r *a literal not in* W, *and removing one in* $V - W$.

(f) *If* r *contains all the literals in* W, *then removing a* z *in* $V - W$ *will increase the performance of* r *by at least* 2^{1-q}.

(g) *If* r *contains all the literals in* W, *then adding a* z *in* $V - W$ *will decrease the performance of* r *by at least* 2^{-q}.

(h) *If* $m > q$, *then adding a* z *to an* r *of length at most* $q - 2$ *will increase performance by at least* 2^{1-q}, *and removing a* z *from an* r *of length at most* $q - 1$ *will decrease performance by at least* 2^{1-q}.

To verify the above eight claims, suppose that r is $y_{i[1]} \cdots y_{i[k]}$.

(a) Consider a z in $W - V$. Then, conjoining z to $y_{i[1]} \cdots y_{i[k]}$ will change the hypothesis from $+1$ to the value of -1 on the points satisfying $z' y_{i[1]} \cdots y_{i[k]}$. Clearly, the ideal function f takes value -1 at all these points since $z = -1$. These points will, by Fact 1, have probability $2^{-(k+1)} \geq 2^{-q}$. Hence, the performance will improve by at least twice this quantity, namely 2^{1-q}.

(b) Suppose that $z = y_{i[1]}$. Removing it from r will change the hypothesis from -1 to the value of $+1$ on the points satisfying $z' y_{i[2]} \cdots y_{i[k]}$. Clearly, the ideal function takes value -1 at all these points. These points will, by Fact 1, have probability $2^{-k} \geq 2^{-q}$ and hence the performance will degrade by at least twice this quantity.

(c) Suppose the added literal is z and the removed literal is $y_{i[1]}$. Then, the hypothesis changes (i) from 1 to -1 on the points where $z' y_{i[1]} \cdots y_{i[k]} = 1$, and (ii) from -1 to $+1$ on the points such that $z y'_{i[1]} y_{i[2]} \cdots y_{i[k]} = 1$. Now (i) changes from incorrect to correct at all such points, and applies with probability $2^{-(k+1)}$. Also (ii) applies to a set of points with the same total probability $2^{-(k+1)}$, but the change on some of the points may be from correct to incorrect. To show that the net change caused by (i) and (ii) in combination is beneficial as claimed it is sufficient

to observe that (ii) is nondetrimental on a sufficient subdomain. To see this, we consider the literals Z in W that are missing from r but other than z, and suppose that there are u of these. Then, on the domain of points $zy'_{i[1]}y_{i[2]}\cdots y_{i[k]} = 1$ that specify (ii) we note that on the fraction 2^{-u} of these the correct value of the ideal function is indeed 1. Hence, the improvement due to (i) is not completely negated by the degradation due to (ii). The improvement in performance is therefore at least $2^{-u-k} \geq 2^{-m-q}$.

(d) Suppose the added literal is z and the removed literal is $y_{i[1]}$. Then, the hypothesis changes (i) from 1 to -1 on the points where $z'y_{i[1]}\cdots y_{i[k]} = 1$, and (ii) from -1 to $+1$ on the points such that $zy'_{i[1]}y_{i[2]}\cdots y_{i[k]} = 1$. Now (ii) is an incorrect change at every point and applies with probability $2^{-(k+1)}$. Also (i) applies to a set of points with the same total probability $2^{-(k+1)}$. To show that the net change caused by (i) and (ii) in combination is detrimental to the claimed extent, it is sufficient to observe that (i) is detrimental on a sufficient subdomain. To see this, we consider the literals Z in W that are missing from r, and suppose that there are u of these. Then, on the domain of points $z'y_{i[1]}y_{i[2]}\cdots y_{i[k]} = 1$ that specify (i) we note that on the fraction 2^{-u} of these the correct value of the ideal function is indeed 1. Hence, (i) suffers a degradation of performance on a fraction 2^{-u} of its domain, and hence the rest cannot fully compensate for the degradation caused in (ii). The combined decrease in performance is therefore at least $2^{-u-k} \geq 2^{-m-q}$.

(e) Suppose the added literal is z and the removed literal is $y_{i[1]}$. Then, the hypothesis changes (i) from 1 to -1 on the points where $z'y_{i[1]}\cdots y_{i[k]} = 1$, and (ii) from -1 to $+1$ on the points such that $zy'_{i[1]}y_{i[2]}\cdots y_{i[k]} = 1$. Now, (ii) is an incorrect change at every point and applies with probability $2^{-(k+1)}$, and (i) applies to a set of points with the same total probability $2^{-(k+1)}$ but is a correct change at every point. The second part of the claim follows similarly. Again each of the two conditions holds with probability 2^{-k-1}. But now, if there are u literals in W missing from r, then over each of the two conditions stated in (i) and (ii) function f is true on a fraction z^{-u}. Hence, the effect of the two changes is again to cancel and keep the performance unchanged.

(f) Suppose that $z = y_{i[1]}$. Removing z from $y_{i[1]}\cdots y_{i[k]}$ will change the value of the hypothesis from -1 to $+1$ on the points satisfying $z'y_{i[2]}\cdots y_{i[k]}$. But all such points have true value $+1$ if r contains all the literals in W. Hence, this gives an increase in performance by an amount $2^{1-k} \geq 2^{1-q}$.

(g) Consider a z in $V - W$. Then, conjoining z to $y_{i[1]}\cdots y_{i[k]}$ will change the hypothesis from $+1$ to -1 on the points satisfying $z'y_{i[1]}\cdots y_{i[k]}$. But all such points have true value $+1$ if r contains all the literals in W. Hence, conjoining z will cause a decrease in performance by an amount $2^{-k} \geq 2^{-q}$.

(h) If $m > q$, then the hypothesis equals -1 on a large fraction of at least $1-2^{-2-q}$ of the points. A conjunction of length $k \leq q-2$ will equal -1 on $1-2^{-k} \leq 1-2^{2-q}$ points, and a conjunction of length $k \leq q-1$ on $1-2^{-k} \leq 1-2^{1-q}$ of the points. Hence, the fraction of points on which the -1 prediction will be made increases by $(1-2^{-k-1})-(1-2^{-k}) = 2^{-k-1} \geq 2^{1-q}$ if $k \leq q-2$ and a literal is added,

and decreases by $(1-2^{-k})-(1-2^{-k+1})=2^{-k}\geq 2^{1-q}$ with the removal of one, if $k\leq q-1$. If $m>q$, then the corresponding increase/decrease in the fraction of points on which predictions are correct is at least 2^{-q}, since the fraction of predicted -1 points changes by twice this quantity, and the true $+1$ points amount to at most a half this quantity.

To prove the proposition, we are first supposing that the number m of literals in the ideal function is no more than q. Then, the intended evolution sequences will have two phases. First, from any starting point of length at most q the representation will increase the number of its literals that are in W by a sequence of steps as specified in Claims (a) and (c). Interspersed with these steps, there may be other steps that cause similar inverse polynomial improvements, but add or remove non-ideal literals. Once the conjunction contains all the literals of the ideal conjunction, it enters into a second phase in which it contracts removing all the non-ideal literals via the steps of Claim (f).

The assertions of the above paragraph can be verified as follows. Claims (a) and (c) ensure that *as long as some ideal literal is missing from r*, beneficial mutations, here defined as those that increase performance by at least 2^{-2q} will be always available and will add a missing ideal literal. Further, Claims (b), (d) and (e) ensure that mutations that remove or exchange ideal literals will be deleterious, reducing the performance by at least 2^{-2q}, or neutral, and hence will not be executed. Some beneficial mutations that add or remove nonideal literals may however occur. However, since each literal not already in the conjunction, will be generated by N with equal probability as a target for addition or swapping in, the Coupon Collector's model (Fact 3) can be applied. If the ideal conjunction contains m literals, then after $\mathrm{CC}(n,m,\varepsilon/2)=O((n\log n+n\log(1/\varepsilon))$ generations all m will have been added or swapped in, except with probability $\varepsilon/2$.

Once r contains all the ideal literals, then the only beneficial mutations are those that remove nonideal literals (Claim (f)). Adding nonideal literals (Claim (g)), replacing an ideal literal by a nonideal literal (Claims (d)), or replacing a nonideal literal by a nonideal literal (Claim (e)) are all deleterious or neutral. Hence, in this second phase after $\mathrm{O}(n)$ steps, the ideal conjunction with perfect performance will be reached.

We conclude that, in the case that the number of literals in both r_0 and the ideal conjunction is at most q, then the evolution will reach the correct hypothesis in the claimed number of stages, except with probability ε, which accounts for both the empirical tests being unrepresentative, as well as the evolution steps failing for other reasons.

In case the initial conjunction is of length greater then q, we allow for a prologue phase of evolution of the following form. We define the neighborhood of each such long conjunction to be all the conjunctions obtained by removing one or none of its literals, each one being generated with equal probability. Clearly, the removal of a literal from a hypothesis having $k>q$ literals will change its value on at most $2^{-k}<2^{-q}=2\varepsilon/(dn)$ of the distribution and hence the performance, if it decreases, will decrease by no more than $2\varepsilon/(dn)$. Hence, if we set the tolerance to $t=4\varepsilon/(dn)$ then a mutation that decreases the number of literals will be available, and will be detected as a neutral mutation as long as its empirical performance is not less than its true performance by more than $\delta=2\varepsilon/(dn)$. The probability of this happening is small, as can be seen by substituting

$a = -1, b = 1, \delta = t/2$, and $s = \delta^3$ in the Hoeffding bound (Fact 2), yielding $\exp(-dn/4\varepsilon)$. After this process runs its course, which will take $O(n)$ stages except with exponentially small probability, a short conjunction of length at most q will be reached.

Finally, we consider the alternative case that the number m of literals in the ideal conjunction is more than q. If r_0 has at most q literals, then in the evolution beneficial steps (h) that add literals will be always available until the hypothesis becomes of length $q - 1$. Further, steps (h) that remove literals will be never taken since these are deleterious. Once length $q - 1$ is achieved the length can change only between length $q - 1$ and q, and the performance will be at least $1 - 2(2^{1-q} + 2^{-m}) = 1 - 5.2^{-q} = 1 - 5\varepsilon/(dn)$. In the alternative case that r_0 has more than q literals, the prologue phase will be involved as before until length at most q will be reached, and the previous condition joined.

The result for conjunctions therefore follows with $g(n, 1/\varepsilon) = O(n \log(n/\varepsilon))$. We note that $s = \Omega((n/\varepsilon)^6)$ is sufficient for both phases.

The claimed result for disjunctions follows by Boolean duality: Given an expression representing a Boolean function, by interchanging "and" and "or" operators and negating the inputs will yield the negation of the original function. □

The algorithm as described above may be applied to conjunctions with negated variables, but will then fail sometimes. For example, if the starting configuration contains many literals that are negations of literals in the ideal function, then it may have high performance because it predicts -1 everywhere. However, it would appear difficult in that case to find an improved hypothesis by local search.

If initialization is allowed, then the above results can be obtained much more easily, and then also allow negations.

PROPOSITION 5.2. *Conjunctions and disjunctions are evolvable with initialization over the uniform distribution.*

The reader can verify this by considering conjunctions with initial representation 1. If the hypothesis is of length k and consists only of literals that occur in the true conjunction of length $m > k$, then adding a literal from the true conjunction will increase performance by 2^{-k}, while adding one not in the true conjunction will increase performance by $2^{-k} - 2^{1-h}$. If $m = k$, then adding a literal will decrease performance by at least 2^{-k}. Then, if we let $q = \log_2(d/\varepsilon)$ for an appropriate constant d, choose tolerance 2^{-q-1}, have a neighborhood that either adds a literal or does nothing, and stop adding new literals if the conjunction reaches length q, then evolution with optimization will proceed through conjunctions of literals exclusively from W until performance at least $1 - \varepsilon$ is reached. It follows that this evolution algorithm will work with optimization and initialization, and hence, by Proposition 3.3, it is evolvable with initialization alone, but for a redundant representation.

6. *Discussion*

We have introduced a framework for analyzing the quantitative possibilities of and limitations on the evolution of mechanisms. It can be weakened in several ways, separately and in combination, to yield notions that impose less onerous

requirements. First, one can entertain the definition for just one specific distribution as we did for our positive results in Section 5. The question whether significant classes are provably evolvable for all distributions is one of the more important questions that our formulation raises. Second, the requirement of having the performance be able to approach arbitrarily close to the best possible can be relaxed. This permits processes where computations are feasible only for obtaining approximations to the best possible. Third, the starting point need not be allowed to be arbitrary. There may be a tradeoff between the robustness offered by allowing arbitrary starting points, and the complexity of the mechanisms that can evolve. Wider classes may be evolvable in any of these less onerous senses than in the most robust sense. We can equally study, in the opposite direction, the quantitative tradeoffs obtained by constraining the model more, by disallowing, for example, neutral mutations or redundant representations, or by insisting on a fixed tolerance. We note that our algorithm for conjunctions as describe does exploit neutral mutations. Also, it uses a fixed tolerance for the main phase, and a different tolerance in the prologue.

Many important questions are left unresolved regarding the robustness of the model and its variants. Which variants leave the class unchanged and which do not? In a very recent work, Feldman [2008] makes significant progress on this question.

Our result that some structures, namely monotone conjunctions and disjunctions are evolvable over the uniform distribution, we interpret as evidence that the evolution of significant algorithmic structure is a predictable and analyzable phenomenon. This interpretation is further supported by the observation that the theory, analogously to learning theory, analyzes only the *granularity* of the structure that can evolve in a single phase with a single ideal function. If 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*.

In general, our study can be viewed as an attempt to capture the Darwinian notion of random variation with selection in terms of a computationally compelling model. It is an open question, of course, whether biological evolution has greater power than this in some direction. We conjecture, however, that any such extra power, if such exists, also lies within the realm of the PAC learnable.

We note that our model makes no assumptions about the nature of a mutation, other than that it is polynomial time computable by a randomized Turing machine. Thus, the biological phenomena found in DNA sequences of point mutations, copying of subsequences, and deletion of subsequences, are all easily accommodated in the model.

The proof of Proposition 3.3 hints at one possible purpose of redundancy that is also believed to occur widely in biology. In that construction, two near-identical copies of a subsequence are maintained, one of which controls behavior, while the other acts as a reservoir for recording history and thereby offers expanded possibilities for future mutations.

In our model, in each generation just one representation will survive. This is sufficient for our goal of studying which function classes are evolvable. In principle, more traditional issues concerning variability within populations are also expressible in our model. In that case, the surviving representation in our model should be viewed as the genome of the whole population. For example, the idea that diversity in the gene pool of a population serves the purpose of protecting a species against unpredictable changes in the environment can be expressed as follows: We represent the hypotheses of all N members of a population by a hypothesis that concatenates them all but has a distinguished first member. The total hypothesis would still be of polynomial size if N is. The distinguished first member determines the performance while the rest form a reservoir to facilitate future mutations. In a mutation, the subhypotheses would be cyclicly shifted by an arbitrary amount so that any one of them can come into first place, and only this first one would undergo mutation. In this way, the diverse gene pool of a population can be represented. In one phase of evolution, the hypotheses that have their first subhypotheses best fitted to the then current environment would win, but they would retain diversity in their reservoir. Of course, once we regard the genome of a population as a single genome, then there may be useful operations on them beyond cyclic shifts, such as operations that splice together parts of the individual subgenomes. The latter operations correspond to recombination.

In our model, large populations are useful when small improvements in performance need to be detected reliably. Small populations can also have a role in permitting deleterious mutations to be adopted, ones which would not be adopted in larger populations.

It is natural to ask what is the most useful view of the correspondence between our view of circuits and what occurs in biology. What do the nodes, and the functions that evolve, in our model correspond to in biology? It seems plausible to suppose that at least some of the nodes correspond to the expression of particular proteins. Then, the regulatory region associated with each protein coding region would correspond to the function evaluated at that node. Possibly such regions may have to be subdivided further into nodes and functions. The fact that there are highly conserved regions in the genome that code for proteins, and some other regions that are not so conserved [Bejerano et al. 2004; Dermitzakis et al. 2005], is consistent with this viewpoint.

In the case that a node corresponds to the expression of a fixed protein, the interpretation of the ideal functions in our model is particularly simple. Suppose the genome has an evolution algorithm for the class C of functions, such as disjunctions. Then, the ideal function f simply expresses for each combination of other variables the best choice of whether to, or whether not to (or how much to) express that protein. Evolution will then be guaranteed to proceed towards f provided f lies within C. Presumably biology has made a sophisticated selection in its choice of C. If C is too restrictive, then the complex functions of biology may not be expressible. If C is too extensive, then it would be infeasible for evolution to successfully navigate among its members as changing environments may require.

Modularity, in biology or engineering, is the idea that systems are composed of separate components that have identifiable separate roles. If evolvability is severely constrained to limited classes of functions as our theory suggests, then systems that evolve would be constrained to be modular, and to consist of many identifiable small modules. Hence, modularity in biology would be a consequence of the limitations of evolvability.

A unified theory for learning and evolution is of potential significance to the studies of cognition and of its emulation by machine. A major challenge in understanding cognition is that in biological systems the interplay between the knowledge that is learned through experience by an individual and the knowledge inherent in the genes, is complex, and it is difficult to distinguish between them. In attempts to construct computer systems for cognitive functions (e.g., for vision) this challenge is reflected in the difficulty of providing an effective split between the preprogrammed and the learning parts. The unified viewpoint on learning and evolution that we have developed here, suggests that cognitive systems can be viewed as pure learning systems. The knowledge and skills a biological organism possesses can be viewed as the accumulation of what has been learned by its ancestors over billions of years, and what it has learned from its individual experience since conception. Robust logic [Valiant 2000] is a mathematical framework based on learning that aims to encompass cognitive tasks beyond learning, particularly reasoning. The pragmatic difficulty of finding training data for systems to be built along such principles has been pointed out [Valiant 2006]. By acknowledging that the training data may also need to cover knowledge learned through evolution one is acknowledging what happens in existing cognitive systems, namely the biological ones. It is possible that learning is the only way of guaranteeing sufficient robustness in large-scale cognitive systems. In that case, it would follow that the construction of cognitive systems with human level performance should be conceptualized as a learning task that encompasses knowledge acquired in biological systems through evolution as well as experience.

We have shown that with regard to the acquisition of complex mechanisms evolvability can be viewed as a restricted form of learnability. If it is technically the more constrained, it should not be viewed as the more mysterious.

ACKNOWLEDGMENTS. The author is grateful to Daniel Fisher and Martin Nowak for stimulating discussions and encouragement, to Vitaly Feldman, Brian Jacobson, Loizos Michael and Rocco Servedio for their technical comments on earlier drafts of this article, and to Gill Bejerano and Richard Karp for some helpful observations.

REFERENCES

BÄCK, T., FOGEL, D. B., AND MICHALEWISZ, Z. (EDS.). 1997. *Handbook of Evolutionary Computation*. Oxford Univ. Press, Oxford, UK.

BERJERANO, G. PHEASANT, M., MAKUNIN, I., STEPHEN, S., KENT, W. J., MATTICK, J. S., AND HAUSSLER, D. 2004. Ultraconserved elements in the human genome. *Science 304*, 1321–1325.

BLUMER, A., EHRENFEUCHT, A., HAUSSLER, D., AND WARMUTH, M. K. 1989. Learnability and the Vapnik Chervonenkis dimension. *J. ACM 36*, 4, 929–965.

BÜRGER, R. 2000. *The Mathematical Theory of Selection, Recombination, and Mutation*. Wiley, Chichester, UK.

DARWIN, C. 1859. *On the Origin of Species by Means of Natural Selection*. John Murray, London, UK.

DERMITZAKIS, E. T., RAYMOND, A., AND ANTONARAKIS, S. E. 2005. Conserved nongenic sequences—An unexpected feature of mammalian genomes. *Nat. Rev. Genet. 6*, 2(Feb.), 151–157.

DRAKE, J. W., CHARLESWORTH, B., CHARLESWORTH, D., AND CROW, F. J. 1998. Rates of spontaneous mutation. *Genetics 148*, 1667–1686.

EHRENFEUCHT, A., HAUSSLER, D., KEARNS, M., AND VALIANT, L. G. 1989. A general lower bound on the number of examples needed for learning. *Inf. Comput. 82*, 2, 247–261.

FELDMAN, V. 2008. Evolvability from learning algorithms. In *Proceedings of the 40th ACM Symposium on Theory of Computing*. ACM, New York, 619–628.

FISCHER, P., AND SIMON, H. U. 1992. On learning ring-sum expressions. *SIAM J. Comput. 21*, 1, 181–192.

FISHER, R. A. 1930. *The Genetical Theory of Natural Selection*. Oxford Univ. Press, Oxford, UK.

GAREY, M. R., AND JOHNSON, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA.

HELMBOLD, D., SLOAN, R., AND WARMUTH, M. K. 1992. Learning integer lattices. *SIAM J. Computing 21*, 2, 240–266.

HOEFFDING, W. 1963. Probability inequalities for sums of bounded random variables. *J. Amer. Stat. Assoc. 58*, 13.

KEARNS, M. 1998. Efficient noise tolerant learning from statistical queries. *J. ACM 45*, 6, 983–1006.

KEARNS, M., AND VALIANT, L. G. 1994. Cryptographic limitations on learning Boolean formulae. *J. ACM 41*, 1, 67–95.

KEARNS, M., AND VAZIRANI, U. 1994. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA.

KIMURA, M. 1968. Evolutionary rate at the molecular level. *Nature 217*, 624–626.

KUMAR, S., AND SUBRAMANIAN, S. 2002. Mutation rates in mammalian genomes. *Proc. Nat. Acad. Sci. 99*, 803–808.

MAYNARD-SMITH, J. 1982. *Evolution and the Theory of Games*. Cambridge Univ. Press, Cambridge, UK.

PAPADIMITRIOU, C. H. 1994. *Computational Complexity*. Addison-Wesley, Reading, MA.

PITT, L., AND VALIANT, L. G. 1988. Computational limitations on learning from examples. *J. ACM 35*, 4, 965–984.

ROFF, D. A. 1997. *Evolutionary Quantitative Genetics*. Chapman & Hall, New York.

ROS, J. P. 1993. Learning Boolean functions with genetic algorithms: A PAC analysis. In *Foundations of Genetic Algorithms*, L. D. Whitley, Ed., Morgan-Kaufmann, San Mateo, CA, 257–275.

VALIANT, L. G. 1984. A theory of the learnable. *Comm. ACM*, *27*, 11, 1134–1142.

VALIANT, L. G. 2000. Robust logics. *Artif. Intell. J. 117*, 231–253.

VALIANT, L. G. 2006. Knowledge infusion. In *Proceedings of the 21st National Conference on Artificial Intelligence. (AAAI06)*. 1546–1551.

WAGNER, G. P., AND ALTENBERG, L. 1996. Complex adaptations and the evolution of evolvability. *Evolution 50*, 3, 967–976.

WEGENER, I. 2001. Theoretical aspects of evolutionary algorithms. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming*. Lecture Notes in Computer Science, vol. 2076, Springer-Verlag, Berlin, Germany, 64–78.

WRIGHT, S. 1968–1978. *Evolution and the Genetics of Populations, A Treatise*. University of Chicago Press, Chicago, IL.

RECEIVED JUNE 2006; ACCEPTED MARCH 2008