CONVERGENCE LAWS FOR RANDOM WORDS

James F. Lynch *
Department of Mathematics and Computer Science
Clarkson University
Potsdam, N. Y. 13699-5815, U. S. A.

Abstract

A random word is a finite sequence of symbols chosen independently at random from some finite alphabet. The probability distribution of the symbols may be constant, or it may depend on the length of the word, analogous to the way that edge probabilities of a random graph depend on the number of vertices of the graph. Two formal languages which express properties of words are considered: a first-order predicate calculus, and a monadic second-order calculus. It is shown that every sentence in the first-order language has probability that converges to a limiting value as the length of the word increases. This extends the known convergence result for first-order sentences about random words with constant probabilities. A weaker form of convergence law is proven for the monadic second-order language. The proofs rely on a combinatorial game (the Ehrenfeucht game), and results on the asymptotic behavior of finite Markov chains with variable transition probabilities.

1 Introduction

A word is a finite sequence of symbols taken from some finite alphabet, which is assumed to be fixed. A property of words is a set of words. In this paper, we will give a characterization of properties that are expressible in certain formal languages. For simplicity, we shall assume the words are over the two symbol alphabet $\{0, 1\}$. All of our results easily extend to words over any finite alphabet. Thus a *word of length* n is a sequence $w = w_1 \ldots w_n$ of 0's and 1's. We put $|w|$ for the length of w. Let p be some fixed function of the natural numbers whose range is in $[0, 1]$. The *random word of length* n is $w = w_1 \ldots w_n$ where for each $x = 1, \ldots, n$ independently, the probability that $w_x = 0$ is $p(n)$. Given a property $\mathcal{P}$ of words, $\mathrm{pr}(\mathcal{P}, n)$ is the probability that the random word of length n has property $\mathcal{P}$.

*This article is based on a lecture presented at the Eighteenth Australasian Conference on Combinatorial Mathematics and Combinatorial Computing. The research was supported by NSF Grant CCR-9006303.

One language we shall use to describe properties of words is a particular first-order predicate calculus. It involves *variables* x, y, z (with or without subscripts) that represent indices in a word. The variables $y, y_1, y_2, \ldots, z, z_1, z_2, \ldots$, and unsubscripted x are not necessary; they are only to improve the readability of formulas. The essential feature is that the language must have an infinite number of distinct variables. The building blocks of all formulas are the *atomic formulas.* In our language, we have $x \leq y$, $x < y$, $x = y$, and all similar formulas with other pairs of variables. They have their usual meaning when x and y are interpreted as indices in a word. The remaining atomic formulas are $Z(x)$ and all similar formulas with other variables. $Z(x)$ means that $w_x = 0$ where w_x is the xth symbol in the word w.

Longer formulas are constructed from the atomic formulas by combining them with the boolean operators $\neg$ (not), $\vee$ (or), $\wedge$ (and), $\Rightarrow$ (implies), and $\Leftrightarrow$ (if and only if), and by binding variables to the quantifiers $\forall$ (for all) and $\exists$ (there exists). Again, we could dispense with many of these constructs; for example $x = y$ is equivalent to $x \leq y \wedge y \leq x$. A thorough introduction to the syntax of first-order predicate calculi may be found in Mendelson [21].

In general, the truth of a formula depends on the values assigned to its variables, e.g. $x \leq y$ obviously depends on the relationship between x and y. However, a formula in which all variables are bound to quantifiers is true or false for any given word. Such a formula is a *sentence.* We will write $w \models \sigma$ if the word w *satisfies* the sentence σ, i.e. σ is *true* for w. For example,

$$00011 \models \exists x(\neg Z(x) \wedge \forall y(Z(y) \Leftrightarrow y < x)).$$

That is, there exists x such that $w_x = 1$ and for all y, $w_y = 0$ if and only if $y < x$.

The other language we will consider is a monadic second-order predicate calculus. In addition to all of the constructs of our first-order language, it has monadic second-order variables $X, X_1, X_2, \ldots$ that represent sets of indices, and the atomic formulas $x_i \in X_j$ for all $i, j = 1, 2, \ldots$. Also the quantifiers $\forall$ and $\exists$ may be applied to the second-order variables. The term monadic refers to the fact that the second-order variables are unary or one place. More general second-order languages have variables that represent relations of higher arity.

Given a class of structures or models and a language pertaining to the structures, a fundamental problem is to characterize those properties expressible in the language. A property is expressible if there is a sentence that is true for precisely those structures that have the property. Some simple examples of properties expressible in our first-order language are the following:

"All the symbols in w are the same."
$(\forall x)(Z(x)) \vee (\forall x)(\neg Z(x))$

"The first symbol in w is 1."
$(\exists x)(\forall y)(x \leq y \wedge \neg Z(x))$

"The word w has a substring 00100."
(Straightforward but tedious to write.)

On the other hand, there are many simple properties that cannot be expressed in this language, for example, "w has an even number of 1's." However, it is expressible in our monadic second-order language:

$$\begin{aligned}
\exists X[\ & \forall x(\forall y(x \leq y) \Rightarrow (x \in X \Leftrightarrow Z(x))) \\
& \wedge \forall x \forall y(y < x \wedge \forall z(y \leq z \wedge z \leq x \Rightarrow z = x \vee z = y) \\
& \qquad\qquad \Rightarrow (x \in X \Leftrightarrow (y \in X \wedge Z(x)) \vee (\neg y \in X \wedge \neg Z(x)))) \\
& \wedge \forall x(\forall y(y \leq x) \Rightarrow x \in X)]
\end{aligned}$$

This sentence σ is true for a word w if and only if there exists a relation $X \subseteq \{1, \ldots, |w|\}$ such that for every index x, $x \in X$ if and only if the number of 1's in $w_1 \ldots w_x$ is even, and $|w| \in X$.

Yet there are still properties not expressible in our monadic second-order language, such as "w is a palindrome." We will not prove the inexpressibility of these properties here. There are several ways of doing this, one of which uses the Ehrenfeucht game that will be described in the next section.

A major motivation for finding characterizations of expressible properties of words comes from computational complexity. The now well-developed theory of descriptive complexity has shown that many of the important complexity classes can be described as classes of properties expressible in various formal languages. A complexity class $\mathcal{C}$ is a set of properties of words that are recognized by some class of automata. A formal language $\mathcal{L}$ *captures* $\mathcal{C}$ if for every property $P \in \mathcal{C}$ there is some $\sigma \in \mathcal{L}$ that expresses P. That is, $P = \{w : w \models \sigma\}$. In addition, if for every sentence $\sigma \in \mathcal{L}$ there is $P \in \mathcal{C}$ such that σ expresses P, then P *corresponds* to $\mathcal{L}$.

One of the earliest examples of this kind of correspondence is due to Büchi [2] and Elgot [8]. It states that the class of properties recognizable by finite state automata (the class of regular languages) corresponds to our monadic second-order language. More recently, there have been characterizations of parallel complexity classes in terms of first-order languages. The class AC^0 of properties recognized by bounded-depth parallel networks with a polynomial number of processors has been studied by Furst, Saxe, and Sipser [12], Ajtai [1] and many others. It corresponds to a first-order language like ours, but in addition to $\leq, <, =$ it has other relational symbols with fixed interpretations. Likewise, the class of properties recognized in nondeterministic linear time is captured by a monadic second-order language with an additional relational symbol with a fixed interpretation (Lynch [19]). More generally, the class NP of properties recognized in nondeterministic polynomial time corresponds to general monadic second-order languages (Fagin [9]). These and many other theorems of descriptive complexity theory are surveyed in Immerman [15]. A feature of virtually all of the languages that capture complexity classes is that they include the $\leq$ relation. Thus, the languages in this paper are the common basis of the languages of descriptive complexity.

One of the most widely studied characteristics of expressible properties is their asymptotic behavior, in particular whether they have a 0-1 law. The general form of a 0-1 law is as follows. There is a class of finite structures, a language pertinent to the structures, and for every natural number n, a probability measure defined on all structures of size n. For every sentence in the language, the asymptotic probability that the sentence is true for structures of size n approaches 0 or 1 as n gets large. The first 0-1 laws were proven by Gaifman [13] for countable structures and independently by Glebskiĭ et. al. [14] and Fagin [10] for random finite structures with uniform probability distributions. In random graph terminology, these would be random graphs with constant edge probabilities. Most of the 0-1 laws are relatively recent, however (see the survey by Compton [3]). A major motivation behind studying the asymptotic behavior of expressible properties is the hope that the techniques being developed will illuminate or perhaps even solve some of the important problems of computational complexity. So far, these problems, such as whether P is properly contained in NP, have resisted solution by any means, but the area is still quite new. At the conclusion of this paper, we will describe some other questions which may be easier but are still important, and are potential applications of the techniques we will present, via the results of descriptive complexity.

The class of words and our languages do not have a 0-1 law, even for constant p. In fact, the property that the first symbol in w is 1, which we saw was first-order expressible, has probability $1-p$. However, we will prove the following *convergence law.* The theorem covers the cases when $p(n)$ converges to a value less than 1. For $p(n)$ converging to 1, we replace $p(n)$ by $1-p(n)$ and use symmetry.

Theorem 1.1 *Let σ be a first-order sentence. Then there is a positive integer K depending on σ such that for any probability function $p(n)$ that satisfies one of these conditions:*

(i) $p(n) \ll n^{-1}$,

(ii) $n^{-1/k} \ll p(n) \ll n^{-1/(k+1)}$, $1 \leq k < K$,

(iii) $n^{-1/K} \ll p(n)$ *and* $\lim_{n\to\infty} p(n) < 1$, *or*

(iv) $p(n) \sim cn^{-1/k}$ *for some constant* c *and* $1 \leq k \leq K$,

$$\lim_{n\to\infty} \mathrm{pr}(w \models \sigma, n)$$

exists.

For our stronger monadic second-order language, we have a weaker convergence law.

Theorem 1.2 *Let σ be a monadic second-order sentence. Then there is a positive integer K depending on σ such that for any probability function $p(n)$ that satisfies one of the conditions listed in Theorem 1.1, there is a positive integer a such that for all natural numbers $b < a$,*

$$\lim_{n\to\infty} \mathrm{pr}(w \models \sigma, an+b)$$

exists.

That is, the sequence of natural numbers splits up into disjoint arithmetic subsequences, and the probability of σ converges on each subsequence. An immediate consequence of Theorem 1.2 is the following:

Corollary 1.3 *Let σ be a monadic second-order sentence. Then there is a positive integer K depending on σ such that for any probability function $p(n)$ that satisfies one of the conditions listed in Theorem 1.1, the sequence $\langle \mathrm{pr}(w \models \sigma, n)\rangle$ is Cesaro-summable, i.e.*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \mathrm{pr}(w \models \sigma, n)$$

exists.

Special cases of Theorem 1.1 have been proven in earlier publications. The case when p is constant was sketched in Lynch [18], although the key idea in the proof is due to A. Ehrenfeucht (personal communication). Note that this is an instance of Case (i) (when $p = 0$) or Case (iii) (when $p > 0$). Dolan [5] proved convergence (in fact 0-1 laws) for Case (i) and $n^{-1} \ll p(n) \ll n^{-1/2}$ (an instance of Case (ii)). Thus our Theorem 1.1 fills in the gaps between p that approach 0 very rapidly and p that are positive constants. Recently, using a different approach, Shelah and Spencer [22] have proven convergence laws for our first-order language when $p(n)$ satisfies Case (i) or (ii), or $n^{-\epsilon} \ll p(n)$ for any $\epsilon > 0$ (a subcase of Case (iii)). They have also obtained 0-1 laws for a closely related first-order language and the same cases of $p(n)$. To our knowledge, Theorem 1.2 is completely new.

We give some examples illustrating how the asymptotic probability of a sentence can be more complex than the simple convergence law for constant p and first-order sentences. The first example shows that the asymptotic probability of a first-order sentence can depend on the growth rate of p, and demonstrates the threshold phenomenon typical of many graph theoretic properties. Let σ be

$$\exists x \exists y (x < y \wedge \forall z (x \le z \wedge z \le y \Rightarrow z = x \vee z = y) \wedge Z(x) \wedge Z(y)).$$

That is, $w \models \sigma$ if and only if w has the substring 00. Then

$$\begin{aligned}
\lim_{n\to\infty} \mathrm{pr}(w \models \sigma, n) &= 0 \text{ if } p(n) \ll n^{-1/2},\\
\lim_{n\to\infty} \mathrm{pr}(w \models \sigma, n) &= e^{-c^2} \text{ if } p(n) \sim cn^{-1/2} \text{ for some constant } c, \text{ and}\\
\lim_{n\to\infty} \mathrm{pr}(w \models \sigma, n) &= 1 \text{ if } p(n) \gg n^{-1/2}.
\end{aligned}$$

Convergence on arithmetic subsequences is demonstrated by the monadic second-order sentence τ that holds if and only if the length of a word is even. This sentence is similar to our previous example of a sentence that holds if and only if the number of ones is even. Clearly $\mathrm{pr}(\tau, 2n) = 1$ while $\mathrm{pr}(\tau, 2n+1) = 0$.

The rest of the paper is organized as follows. In the next section, we describe a combinatorial game, a generalization of the well-known Ehrenfeucht game, that plays a central role in our proofs. We then prove the weak convergence law (Theorem 1.2) for monadic second-order sentences. The proof also uses a generalization of Markov chains known as a Markov chain with variable transitions. Next we prove the convergence law (Theorem 1.1) for first-order sentences. We conclude with a summary of related results and problems for future study.

2 Ehrenfeucht Games

The only structural feature of sentences that is used in our proofs is their *depth*. This is the maximum level of quantifier nesting. A more formal definition is as follows. A formula with no quantifiers has depth 0. If α_1 and α_2 are formulas with depths d_1 and d_2 respectively, then the depth of $\neg\alpha_1$ is d_1, the depth of $\alpha_1 \vee \alpha_2$, $\alpha_1 \wedge \alpha_2$, $\alpha_1 \Rightarrow \alpha_2$, and $\alpha_1 \Leftrightarrow \alpha_2$ is $\max(d_1, d_2)$, and the depth of $\forall x\alpha_1$, $\exists x\alpha_1$, $\forall X\alpha_1$, and $\exists X\alpha_1$ is $d_1 + 1$.

Let k be fixed. We define an equivalence relation $\equiv$ on the set of words. For any words v and w, $v \equiv w$ if and only if they agree on all monadic second-order sentences of depth at most k, i.e. for any such sentence σ, $v \models \sigma$ if and only if $w \models \sigma$. Given an equivalence class $\mathcal{C}$ of $\equiv$, we will put $\mathcal{C} \models \sigma$ if $w \models \sigma$ for some $w \in \mathcal{C}$.

There is a game-theoretic characterization of $\equiv$ due to Ehrenfeucht [7] that is particularly useful in proofs about expressibility. It was originally formulated for first-order languages, but it extends without difficulty to second-order languages. We will use two versions of the game that are tailored to the languages we are considering.

Definition 2.1 *Let v and w be two words.*

(i) The monadic second-order Ehrenfeucht game *of k rounds on v and w is the following game of perfect information. In each round $i = 1, \ldots, k$ Player I chooses either an index or a set of indices in one of the words, and Player II responds with the same kind of choice in the other word. That is, if Player I chooses an index in $\{1, \ldots, |v|\}$ (or $\{1, \ldots, |w|\}$), then Player II chooses an index in $\{1, \ldots, |w|\}$ (or $\{1, \ldots, |v|\}$), and similarly if Player I chooses a set of indices. Let C_i be the choice made (by either player) in $\{1, \ldots, |v|\}$, and D_i be the choice made (by the other player) in $\{1, \ldots, |w|\}$. Player II wins if $C_1, \ldots, C_k$ and $D_1, \ldots, D_k$ induce the same substructures in v and w. That is, for $1 \le i, j \le k$,*

- *If C_i and C_j are indices, then $C_i \le C_j$ if and only if $D_i \le D_j$.*
- *If C_i is an index, then $v \models Z(C_i)$ if and only if $w \models Z(D_i)$.*
- *If C_i is an index and C_j is a set of indices, then $C_i \in C_j$ if and only if $D_i \in D_j$.*

(ii) The first-order Ehrenfeucht game *is the same, except that the choices made must always be indices.*

Note that if Player I can win the first-order game, then he can win the monadic second-order game *a fortiori*, or equivalently, if Player II can win the monadic second-order game, then he can win the first-order game. The fundamental fact about these games is the following theorem.

Theorem 2.2 (Ehrenfeucht [7]) *(i) Two words v and w agree on all monadic second-order sentences of depth at most k if and only if Player II can win the monadic second-order Ehrenfeucht game of k rounds.*

(ii) Two words v and w agree on all first-order sentences of depth at most k if and only if Player II can win the first-order Ehrenfeucht game of k rounds.

As mentioned above, the Ehrenfeucht game and Theorem 2.2 were originally stated only for first-order sentences. On the other hand, our versions are more restrictive because the game and Theorem apply to all types of models and first or second-order sentences. The proof of this theorem is not difficult. Interested readers may gain some insight into it by proving that Player II can win the Ehrenfeucht games of two rounds on 00011 and 000011 while Player I can win the games of three rounds. Thus, by the Theorem, there must be a first-order sentence of depth three that distinguishes the two words. One such example is

$$\begin{aligned} &\exists x[\ \exists y(y < x \wedge \forall z(y \leq z) \wedge \forall z(y \leq z \wedge z \leq x \Rightarrow z = x \vee z = y)) \\ &\quad \wedge \forall y(x < y \wedge Z(y) \Rightarrow \forall z(x \leq z \wedge z \leq y \Rightarrow z = x \vee z = y))] \end{aligned}$$

This sentence is true for v but false for w.

For the remainder of this section and all of the next, Ehrenfeucht game shall mean monadic second-order game. In Section 4, we will work with the first-order game. As before, k is fixed. Two easy consequences of Theorem 2.2 will play important roles in our proofs.

Lemma 2.3 *There are only finitely many equivalence classes of* $\equiv$.

Proof. Let v and w be two words, $i \leq k$, $C_1, \ldots, C_i$ and $D_1, \ldots, D_i$ be the choices made in an Ehrenfeucht game of i rounds That Player II has won. Define an Ehrenfeucht game of $k - i$ rounds numbered $i + 1, \ldots, k$ played on $\langle v, C_1, \ldots, C_i \rangle$ and $\langle w, D_1, \ldots, D_i \rangle$. In each round $j = i + 1, \ldots, k$, the players choose C_j in v and D_j in w. The rules are the same as before. We put $\langle v, C_1, \ldots, C_i \rangle \underset{i}{\equiv} \langle w, D_1, \ldots, D_i \rangle$ if Player II can win this game. By Theorem 2.2, $\underset{0}{\equiv}$ agrees with $\equiv$. Thus we must show that there are only finitely many equivalence classes of $\underset{0}{\equiv}$. In fact, we will show by decreasing induction on i that there are only finitely many equivalence classes of $\underset{i}{\equiv}$. This is true for $i = k$ because there are only finitely many isomorphism types on $C_1, \ldots, C_k$.

Assuming the result for $i + 1 \leq k$, we show it for i. Given any choices $C_1, \ldots, C_i$ in v, the $\underset{i}{\equiv}$ class of $\langle v, C_1, \ldots, C_i \rangle$ is determined by the set of $\underset{i+1}{\equiv}$ classes of $\langle v, C_1, \ldots, C_{i+1} \rangle$ where C_{i+1} varies over all choices on v. If there are m equivalence classes of $\underset{i+1}{\equiv}$, then 2^m is an upper bound on the number of equivalence classes of $\underset{i}{\equiv}$. □

The second lemma shows that $\equiv$ is preserved by concatenation.

Lemma 2.4 *For any words v and w and $s \in \{0, 1\}$, if $v \equiv w$ then $vs \equiv ws$.*

Proof. By Theorem 2.2, we need only show that the Ehrenfeucht game of k rounds on vs and ws is a win for Player II. Since $v \equiv w$, Player II can win the game on v and w, and this strategy gives a winning strategy for Player II on vs and ws. For $1 \leq i \leq k$ let C_i and D_i be the choices made in the game on vs and ws, and let

$$C_i' = \begin{cases} C_i & \text{if } \ C_i \in \{1, \ldots, |v|\} \\ \emptyset & \text{if } \ C_i = |v| + 1 \\ C_i \cap \{1, \ldots, |v|\} & \text{if } \ C_i \subseteq \{1, \ldots, |v|\} \end{cases}$$

and similarly for D'_i. Using the notation of the previous lemma, Player II should choose so that:

- He can win the $(k-i)$-round game on $\langle v, C'_1, \ldots, C'_i \rangle$ and $\langle w, D'_1, \ldots, D'_i \rangle$.
- $C_i = |v| + 1$ if and only if $D_i = |w| + 1$.
- $|v| + 1 \in C_i$ if and only if $|w| + 1 \in D_i$.

That this can be done follows by an easy induction on i. After k rounds these conditions imply that Player II has won the game on vs and ws. □
Although not needed for our proofs, a more general proposition with essentially the same proof is: for any words v, w, v', and w', if $v \equiv w$ and $v' \equiv w'$ then $vv' \equiv ww'$.

3 Proof of the Weak Convergence Law

Since $\{w : w \models \sigma\}$ is the disjoint union of those equivalence classes $\mathcal{C} \models \sigma$, and there are finitely many of them (by Lemma 2.3), for every n,

$$\mathrm{pr}(w \models \sigma, n) = \sum_{\mathcal{C} \models \sigma} \mathrm{pr}(w \in \mathcal{C}, n).$$

Thus our proof has been reduced to the following lemma.

Lemma 3.1 *There is a positive integer K depending on k such that for any probability function $p(n)$ that satisfies one of the conditions listed in Theorem 1.1 and any equivalence class $\mathcal{C}$ of $\equiv$, there is a positive integer a such that for all natural numbers $b < a$,*

$$\lim_{n \to \infty} \mathrm{pr}(w \in \mathcal{C}, an + b)$$

exists.

Proof. First, let us consider constant p. We construct a finite Markov chain M. Its states are the equivalence classes of $\equiv$. For any two states $\mathcal{C}_1$ and $\mathcal{C}_2$ and $w \in \mathcal{C}_1$, if $w0 \in \mathcal{C}_2$ then there is a transition from $\mathcal{C}_1$ to $\mathcal{C}_2$ with probability p, and if $w1 \in \mathcal{C}_2$ then there is such a transition with probability $1-p$. By Lemma 2.4, The transitions of M are well defined, i.e. they do not depend on the choice of $w \in \mathcal{C}_1$. Note also that if $k \geq 2$, it cannot be that both $w0$ and $w1 \in \mathcal{C}_2$.

M is simply another way of describing the construction of a random word. Starting in the state which is the class containing the word of length 0, we successively choose symbols to add to the growing word. When we stop, the word is in the equivalence class in which the chain has ended.

Now when $p(n)$ depends on n, the transitions of M are no longer constant, and we do not have a Markov chain in the usual sense. This is known as a *Markov chain with variable transitions.* More generally, let $M(0)$ be a matrix of nonnegative real numbers whose row sums are 1, and E be a matrix of real numbers whose row sums are 0. Then for any real variable ε,

$$M(\varepsilon) = M(0) + \varepsilon E$$

is an instance of such a chain. The following theorem is a generalization of fundamental results about finite Markov chains (see for example Feller [11] or Kemeny and Snell [17]).

Theorem 3.2 *Let $M(\varepsilon)$ be a Markov chain with variable transitions. Then there is a positive integer K such that for any probability function $p(n)$ that satisfies one of the conditions listed in Theorem 1.1, there is a stochastic matrix D and a positive integer a such that*

$$\lim_{n\to\infty} M(p(n))^{an} = D.$$

Further, D and a are the same for all functions p in any of the classes of types (i) - (iii).

This theorem was proven by researchers in perturbation theory (see Kato [16]), using analytic and generating function techniques. A graph theoretic proof and algorithm for computing the limiting matrix D may be found in Lynch [20].

To finish the proof, for any $b < a$, $\lim_{n\to\infty} M(p(n))^b$ exists. Therefore $\lim_{n\to\infty} M(p(n))^{an+b}$ exists, and the Lemma follows. □

4 Proof of the Convergence Law

Now we restrict attention to the first-order Ehrenfeucht game, and $v \equiv w$ shall mean v and w agree on all first-order sentences of depth at most k. Let $N(\varepsilon)$ be the chain analogous to $M(\varepsilon)$, but whose states are the first-order equivalence classes. Lemmas 2.3, 2.4, and 3.1 still apply, and the proof of Theorem 1.1 is finished if we show that $a = 1$ in Lemma 3.1. To do this, it suffices to show that from every state $\mathcal{C}$, there is a state $\mathcal{D}$ such that with positive probability, $\mathcal{D}$ can be reached from $\mathcal{C}$ in some fixed number of steps, and $\mathcal{D}$ can reach itself in one step. Then every ergodic set contains a state that can be returned to in one step, and the chain $N(\varepsilon)$ is aperiodic, i.e. $a = 1$. One more lemma will help prove this.

Lemma 4.1 *There is a positive integer m (depending on k) such that for any word w, $w1^m \equiv w1^{m+1}$.*

Proof. We can take $m = 2^k - 1$. Let $C_0 = D_0 = |w|$, $C_{-1} = |w| + m + 1$, and $D_{-1} = |w| + m + 2$. We will show by induction on $i = 0, \ldots, k$ that Player II can play so that the following three conditions hold for $\langle w1^m, C_{-1}, \ldots, C_i\rangle$, $\langle w1^{m+1}, D_{-1}, \ldots, D_i\rangle$, and all $h, j \in \{-1, \ldots, i\}$.

(i) $C_h \leq C_j$ if and only if $D_h \leq D_j$.

(ii) If $C_h \leq |w|$ then $C_h = D_h$.

(iii) If $0 \leq C_j - C_h < 2^{k-i}$ then $C_j - C_h = D_j - D_h$, and if $0 \leq D_j - D_h < 2^{k-i}$ then $C_j - C_h = D_j - D_h$.

These conditions obviously hold for $i = 0$.

Assuming (i) - (iii) hold for i, we show how Player II can choose so that they hold for $i + 1$. If Player I chooses C_{i+1} (or D_{i+1}) $\leq |w|$, then Player II simply chooses so that $C_{i+1} = D_{i+1}$. Clearly (i) and (ii) hold. Also, (iii) holds because if $0 \leq C_j - C_{i+1} < 2^{k-i-1}$ and $C_j > |w|$ then $0 \leq C_j - C_0 < 2^{k-i-1} < 2^{k-i}$, so $C_j - C_0 = D_j - D_0$, implying $C_j - C_{i+1} = D_j - D_{i+1}$.

If Player I chooses $C_{i+1} > |w|$ (choosing $D_{i+1} > |w|$ is symmetric), let $C_h \leq C_{i+1} \leq C_j$ be the chosen indices closest to C_{i+1}. By (i) there is no chosen index between D_h and D_j. If $C_j - C_h < 2^{k-i}$ then $C_j - C_h = D_j - D_j$, and (i) - (iii) hold if Player II chooses $D_{i+1} = D_h + C_{i+1} - C_h$.

If $C_j - C_h \geq 2^{k-i}$ then $D_j - D_h \geq 2^{k-i}$, and there are three cases: $C_{i+1} - C_h < 2^{k-i-1}$, $C_j - C_{i+1} < 2^{k-i-1}$, and neither of the previous two inequalities holds. In the first case, Player II chooses $D_{i+1} = D_h + C_{i+1} - C_h$. Then $C_j - C_{i+1} = C_j - C_h - (C_{i+1} - C_h) > 2^{k-i} - 2^{k-i-1} = 2^{k-i-1}$, and similarly for $D_j - D_{i+1}$, and so (i) - (iii) hold. An analogous argument applies to the second case. In the third case, Player II chooses $D_{i+1} = D_h + 2^{k-i-1}$. Then $D_j - D_{i+1} \geq 2^{k-i} - 2^{k-i-1} = 2^{k-i-1}$. This completes the induction step.

Since Player II can choose so that (i) - (iii) hold for $i = k$, he can win the game. □

To finish the proof of Theorem 1.1, take any state $\mathcal{C}$ and $w \in \mathcal{C}$. By the last lemma, there is a state $\mathcal{D}$ such that $w1^m \in \mathcal{D}$ and $w1^{m+1} \in \mathcal{D}$. Since $p(n) < 1$ for sufficiently large n, with positive probability, $\mathcal{D}$ can be reached from $\mathcal{C}$ in m steps and can return to itself in one step.

5 Related Results and Problems

If binary second-order variables are allowed, Theorem 1.2 fails. A consequence of some theorems in complexity theory (see Lynch [19]) is that any set S of natural numbers that is accepted in nondeterministic time n^2 when encoded in unary notation is definable by a binary second-order sentence. This sentence will be true for all words whose length is in S and false otherwise. An example is the set of primes. Whether there is some even weaker convergence law is unknown.

In a different direction, other papers have studied random relations of degree greater than one on an ordered set of indices. A word may be regarded as a unary relation on a linearly ordered set of indices. A random binary relation on the same set of indices is an ordered random graph. Again, there is no convergence law when the edge probability is constant (Compton, Henson, and Shelah [4]) or variable except in very restricted cases (Dolan and Lynch [6]).

These negative results would seem to imply that the convergence laws of this paper cannot be extended to more powerful languages, in particular ones that capture important complexity classes. However, it is possible that the techniques of this article would be useful in the analysis of specific sentences. Potential applications are to the average case complexity and reliability of algorithms.

Given an algorithm or program, using the methods of descriptive complexity, a sentence can be constructed such that the probability of the sentence equals the probability that the algorithm will halt within a certain number of steps or will enter certain states. The average complexity of the algorithm can be computed from the halting probability, and the reliability is given by the probability that the algorithm does not enter any undesirable states.

sentences, techniques related to those of this paper may be applicable to many sentences that arise in practice. This does not seem an unreasonable hope since a similar philosophy underlies much of the work in automated theorem proving. There, the general problem is unsolvable, and researchers concentrate on finding methods that work on broad classes of sentences.

We conclude with two simple extensions of our languages that capture complexity classes. They may be good candidates for the kind of analysis that was described. One is obtained from our first-order language by adding another binary relation symbol B with a fixed interpretation. For any indices x and y, $B(x,y)$ means that bit x in the binary representation of y is 1. This language corresponds to uniform AC^0. It is a proper subclass of AC^0, but it is still of significant interest and has been studied in numerous papers (again see Immerman [15]). The other language is an extension of our monadic second-order language. We add the ternary relation symbol A such that $A(x,y,z)$ means $x+y=z$. As alluded to in the Introduction, this language captures the class of properties recognizable in nondeterministic linear time, which includes many properties of practical importance.

References

[1] M. Ajtai, Σ_1^1 - formulae on finite structures, *Ann. Applied and Pure Logic* **24** (1983), 1-48.

[2] J. R. Büchi, Weak second-order arithmetic and finite automata, *Z. Math. Logik Grundlagen Math.* **6** (1960), 66-92.

[3] K. J. Compton, 0-1 laws in logic and combinatorics, *Proc. 1987 NATO Adv. Study Inst. on Algorithms and Order*, I. Rival, ed., Reidel, Dordrecht (1988).

[4] K. J. Compton, C.W. Henson, and S. Shelah, Nonconvergence, undecidability, and intractability in asymptotic problems, *Ann. Pure and Applied Logic* **36** (1987), 207-224.

[5] P. Dolan, A zero-one law for a random subset, *Random Struct. Alg.* **2** (1991), 317-326.

[6] P. Dolan and J. F. Lynch, The logic of ordered random structures, to appear in *Random Struct. Alg.*

[7] A. Ehrenfeucht, An application of games to the completeness problem for formalized theories, *Fund. Math.* **49** (1961), 129-141.

[8] C. C. Elgot, Decision problems of finite-automata design and related arithmetics, *Trans. AMS* **98** (1961), 21-51.

[9] R. Fagin, Generalized first-order spectra and polynomial-time recognizable sets, in *Complexity of Computation*, R. Karp, ed., SIAM-AMS Proc. **7**, Am. Math. Soc., New York, 1974, 43-73.

[10] ____________, Probabilities on finite models, *J. Symbolic Logic* **41** (1976), 50-58.

[11] W. Feller, *An Introduction to Probability Theory and its Application*, 3rd ed., Wiley, New York (1967).

[12] M. Furst, J. Saxe, and M. Sipser, Parity, circuits, and the polynomial-time hierarchy, *Math. Systems Theory* **17** (1984), 13-27.

[13] H. Gaifman, Concerning measures in first-order calculi, *Israel J. Math.* **2** (1964), 1-18.

[14] Y. V. Glebskiĭ, D. I. Kogan, M. I. Liogon'kiĭ, and V. A. Talanov, Range and degree of realizability of formulas in the restricted predicate calculus, *Kibernetika (Kiev)* **2** (1969), 17-28; English translation, *Cybernetics* **5** (1972), 142-154.

[15] N. Immerman, Expressibility as a complexity measure: results and directions, *Proc 2nd Structure in Complexity Theory Conf.* (1987), 194-202.

[16] T. Kato, *Perturbation Theory for Linear Operators*, Springer-Verlag, Berlin (1966).

[17] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*, Springer-Verlag, New York-Heidelberg (1976).

[18] J. F. Lynch, Almost sure theories, *Ann. Math. Logic* **18** (1980), 91-135.

[19] ____________, The quantifier structure of sentences that characterize nondeterministic time complexity, *Computational Complexity* **2** (1992), 40-66.

[20] ____________, Threshold functions for Markov chains: a graph theoretic approach, submitted to *Combinatorica.*

[21] E. Mendelson, *Introduction to Mathematical Logic*, Wadsworth and Brooks/Cole, Monterey, California, 1987.

[22] S. Shelah and J. Spencer, Random sparse unary predicates, submitted to *Random Struct. Alg.*

(Received 4/8/92; revised 3/11/92)