The Tenacity of Zero-One Laws

Joel H. Spencer
Courant Institute
New York University
251 Mercer Street
New York, NY 10012
spencer@cs.nyu.edu

Katherine St. John
Dept of Math & Computer Science
Graduate Center & Lehman College
City University of New York
Bronx, NY 10468
stjohn@lehman.cuny.edu

Submitted: 9 March 2000
Revised: 6 August 2000

1 Abstract

The Ehrenfeucht-Fraisse game is a two-person game of perfect information which is connected to the Zero-One Laws of first order logic. We give bounds for roughly how quickly the Zero-One Laws converge for random bit strings and random circular bit sequences. We measure the tenaciousness of the second player ("Duplicator") in playing the Ehrenfeucht-Fraisse game, by bounding the numbers of moves Duplicator can play and win with probability $1-\epsilon$. We show that for random bit strings and random circular sequences of length n generated with a low probability ($p \ll n^{-1}$), the number of moves, $T_\epsilon(n)$, is $\Theta(\log_2 n)$. For random bit strings and circular sequences with isolated ones ($n^{-1} \ll p \ll n^{-1/2}$), $T_\epsilon(n) = O(\min(\log_2 np, -\log_2 np^2))$. For $n^{-1/2} \ll p$ and $(1-p) \ll n^{-1/2}$, we show that $T_\epsilon(n) = O(\log^* n)$ for random circular sequences, where $\log^* n$ has the usual definition– the least number of times you iteratively apply the logarithm to get a value less than one.

2 Introduction

In this paper, we examine the Ehrenfeucht-Fraisse game, and the length of such games over random structures for which a Zero-One Law holds. Assume $\mathcal{M}_n$ is a random structure on n elements. For example, it could be the binary tree with n leaves under uniform probability, the random graph of Erdős and Renyi [4], or the random bit string on n bits. In a general setting (a random structure defined for all n) and fixing positive ϵ, we define the **tenacity function**, $T_\epsilon(n)$, equal to the maximal k so that if $n_1, n_2 \geq n$, then Duplicator wins this k-move Ehrenfeucht-Fraisse game played on independent structures of size n_1 and n_2 with probability at least $1-\epsilon$. A Zero-One Law holds if for every first order sentence, ϕ, and random structure of size n, $\mathcal{M}_n$:

$$\lim_{n\to\infty} \Pr[\mathcal{M}_n \models \phi] = 0 \text{ or } 1$$

where $\mathcal{M} \models \phi$ is an abbreviation for "$\mathcal{M}$ has the property ϕ." In terms of the Ehrenfeucht-Fraisse game, a Zero-One Law implies that, for every k, Duplicator will eventually win, and $T_\epsilon(n) \to \infty$ (this is not obvious, see [6] for more details). But how quickly does the tenacity function grow?

The answer varies for different random structures. The choice of ϵ has little effect in the overall answer, and we focus on the growth rates in terms of the size of the structures. While this paper concentrates on random bit strings and random circular orders, we could also look at the growth rates for other random structures such as the random graph of Erdős and Renyi [4] and random binary trees.

2.1 The Ehrenfeucht-Fraisse Game

In the Ehrenfeucht-Fraisse Game, the players alternate placing pebbles on one of two structures that serve as the game boards. The number of rounds that are played correspond to the complexity of first order sentences considered. Given two structures, $\mathcal{M}_1$ and $\mathcal{M}_2$, $\mathcal{M}_1$ and $\mathcal{M}_2$ are indistinguishable by first order sentences with quantifier rank at most k (written $\mathcal{M}_1 \equiv_k \mathcal{M}_2$) if and only if the second player has a winning strategy for every k-move Ehrenfeucht-Fraisse game played on $\mathcal{M}_1$ and $\mathcal{M}_2$. We define the game below:

Definition 1 *The* **k-move Ehrenfeucht-Fraisse game (EF game)** *on $\mathcal{M}_1$ and $\mathcal{M}_2$ is a two-person game of perfect information. For the game, we have:*

- **Players:** *There are two players:*
  - *Player I, often called* Spoiler, *who tries to ruin any correspondence between the structures.*
  - *Player II, often called* Duplicator, *who tries to duplicate Spoiler's last move.*
- **Equipment:** *We have k pairs of pebbles and the two structures $\mathcal{M}_1$ and $\mathcal{M}_2$ as game boards.*
- **Moves:** *The players take turns moving. At the ith move, the Spoiler chooses a structure and places his ith pebble on an element in that structure. Duplicator then places her ith pebble on an element in the other structure.*
- **Winning:** *If after any of Duplicator's moves, the substructures induced by the pebbles are not isomorphic, then Spoiler wins. After both players have played k moves, if Spoiler has not won, then Duplicator wins.*

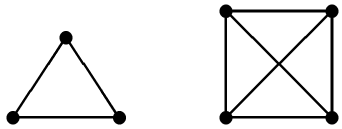

Figure 1: K_3 and K_4

For example, let K_3 and K_4 be the complete graphs on 3 and 4 vertices, respectively (see Figure 1). Then $K_3 \not\equiv_4 K_4$, since Spoiler can place each pebble on a different vertex in K_4, and Duplicator has no response in K_3 since there are only 3 vertices. However, there is a winning strategy for Duplicator to win the 3-move game (i.e. $K_3 \equiv_3 K_4$). To win, Duplicator needs to maintain an isomorphism between the substructures induced by the pebbles, after every move. But, in this case, since every pair of distinct vertices are connected, only equality matters in the placement of pebbles (that is, if the ith and jth pebble are on the same vertex in the first structure, then they need to be in second structure for Duplicator to win). So, on these structures, Duplicator has a winning strategy if and only if the number of moves is less than the number of vertices in either structure.

2.2 Random Bit Strings and Random Circular Sequences

We concentrate on bit strings, that is ordered strings of zeros and ones, and circular orders of bit sequences. Let n be a positive integer, and $0 \leq p(n) \leq 1$. We will write $U(i)$ if the ith bit is "on" in the string, and $\neg U(i)$ if the ith bit is "off." U is an unary predicate– there are 2^n possible choices for such a predicate over n elements. The random bit string $U_{n,p}$ is a probability space over predicates U on $[n] = \{1, \ldots, n\}$ with the probabilities determined by $\Pr[U(x)] = p(n)$, for $1 \leq x \leq n$, and the events $U(x)$ are mutually independent over $1 \leq x \leq n$. Similarly, the random circular bit sequence $C_{n,p}$ is a probability space over predicates U on $[n] = \{1, \ldots, n\}$ with the probabilities determined by $\Pr[U(x)] = p(n)$, for $1 \leq x \leq n$, and the events $U(x)$ are mutually independent over $1 \leq x \leq n$. A bit string is a linearly ordered sequence of zeros and ones, while a circular order roughly is a bit string with the ends "glued" together. The bit strings have an underlying order to their bits, and we will write $i \leq j$ if the ith bit comes before the jth bit. For the circular ordering, there is no beginning or end, we can only talk about the order of two elements with respect to a third. We will write $cc(i, j, k)$ if j is between i and k (see Figure 2), when moving in a counter-clockwise direction.

We form statements about the bit strings and circular orders using $=$ (equality), $\leq$ (linear order for bit strings) or cc (counterclockwise order for circular sequences), U (an unary predicate, indicating if a bit is one or zero), the binary connectives $\vee$ (disjunction) and $\wedge$ (conjunction), $\neg$ (negation), and the first order quantifiers $\exists$ (existential quantification) and $\forall$ (universal quantification). "First-order" refers to the range of the quantifiers–

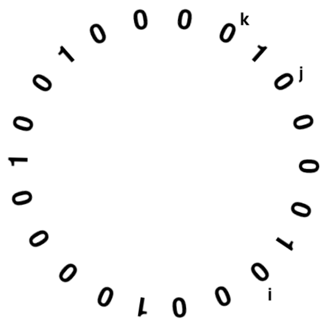

Figure 2: A circular bit sequence with $cc(i, j, k)$

we only allow quantification over variables, not sets of variables. For example, let ϕ be the first order sentence for bit strings:

$$(\exists x)(\forall y)(x \leq y)$$

ϕ expresses the property that there is a least element. The x and y are assumed to range over elements of the universe, or underlying set of the structure. The sentence

$$(\exists x)[U(x) \wedge (\forall y)(x \leq y)]$$

expresses the property that the bit string starts with a one. No equivalent statement can be made in the circular language as there is no notion of least element. We can, however, express the property that there exists two consecutive ones by the first order sentence:

$$(\exists x)(\exists y)[x \neq y \wedge U(x) \wedge U(y) \wedge \neg(\exists z)(cc(x, z, y))]$$

While many things can be expressed using first order sentences, many cannot. For example, there is no first order sentence that captures the property that a structure's underlying set has an even number of elements (see [3], p. 21). That is, there is no first order sentence ϕ such that for every model $\mathcal{M}$, $\mathcal{M} \models \phi \iff m$ is even.

By the work of Dolan [2], the random bit string satisfies the Zero-One law for $p(n) \ll n^{-1}$ and $n^{-1} \ll p(n) \ll n^{-1/2}$. Dolan also showed that the Zero-One Law does not hold for bit strings with $n^{-1/k} \ll p(n) \ll n^{-1/(k+1)}$, $k > 1$. In [5], Shelah and Spencer showed that for $p(n) \ll n^{-1}$ or $n^{-1/k} \ll p(n) \ll n^{-1/(k+1)}, k > 1$, a Zero-One Law holds for circular sequences. (The same result holds for $1 - p(n) \ll n^{-1}$ or $n^{-1/k} \ll 1 - p(n) \ll n^{-1/(k+1)}, k > 1$.)

3 Very, Very Sparse

When the probability generating the random bit strings and circular sequences is small, $p \ll n^{-1}$, then, almost surely, ones do not occur, and a Zero-One Law holds (see [7]). We will focus on bit strings in this section, but the results hold also for circular sequences.

Since the random bit string for these probabilities is just a string of zeros, the game on any structures reduces to counting the number of bits. For $n_1, n_2 \geq 2^k - 1$, Duplicator wins the k-move game on U_{n_1} and U_{n_2}. We omit the proof of this since it can be found in many places, including [3]. Instead, we give an example to illustrate why this fact is true. Consider the Ehrenfeucht-Fraisse game on U_7 and U_8:

$$\begin{array}{cccccccccccccccccccc}
&&&&U_7&&&&&&&&&&U_8&&&&& \\
[& 0 & 0 & 0 & 0 & 0 & 0 & 0 &] & \quad & [& 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &]
\end{array}$$

Let's play a 3-move game on these bit strings. Spoiler moves first. His goal is to illustrate the difference between the bit strings, which, in this case, is just their differing lengths. Assume Spoiler places his first pebble (call it s_1) on the middle bit of U_7:

$$\begin{array}{cccccccccccccccccccc}
&&&&U_7&&&&&&&&&&U_8&&&&& \\
[& 0 & 0 & 0 & 0 & 0 & 0 & 0 &] & \quad & [& 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &] \\
&&&&\uparrow&&&&&&&&&&&&&&& \\
&&&&s_1&&&&&&&&&&&&&&&
\end{array}$$

Duplicator tries to duplicate Spoiler's last move on the other bit string, U_8. Let's assume that she chooses to place her pebble (d_1) on an element close to middle bit, say the fourth bit:

$$\begin{array}{cccccccccccccccccccc}
&&&&U_7&&&&&&&&&&U_8&&&&& \\
[& 0 & 0 & 0 & 0 & 0 & 0 & 0 &] & \quad & [& 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &] \\
&&&&\uparrow&&&&&&&&&&\uparrow&&&&& \\
&&&&s_1&&&&&&&&&&d_1&&&&&
\end{array}$$

Spoiler can choose to play his move on either bit string. Assume he responds by playing his second pebble, s_2, the seventh bit on U_8:

$$\begin{array}{cccccccccccccccccccc}
&&&&U_7&&&&&&&&&&U_8&&&&& \\
[& 0 & 0 & 0 & 0 & 0 & 0 & 0 &] & \quad & [& 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &] \\
&&&&\uparrow&&&&&&&&&&\uparrow&&&\uparrow&& \\
&&&&s_1&&&&&&&&&&d_1&&&s_2&&
\end{array}$$

Duplicator must respond with her move, d_2, on an element of U_7. Recall that if at the end of Duplicator's move, the substructures, induced by the pebbles are not isomorphic, then Spoiler wins. Since the $d_1 < s_2$ in U_8, the choice of d_2 must be greater than s_1 in U_7, to maintain the correspondence of the structures and keep Duplicator in the game. This gives Duplicator the choice of the fifth, sixth, and seventh bits in U_7. Assume she chooses to place d_2 on the sixth bit:

$$\begin{array}{cccccccccccccccccccc}
&&&&U_7&&&&&&&&&&U_8&&&&& \\
[& 0 & 0 & 0 & 0 & 0 & 0 & 0 &] & \quad & [& 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &] \\
&&&&\uparrow&&\uparrow&&&&&&&&\uparrow&&&\uparrow&& \\
&&&&s_1&&d_2&&&&&&&&d_1&&&s_2&&
\end{array}$$

We have that the first pebble is strictly less than the second in both structures, so Spoiler has not won yet. If we're playing the 3-move game, then this would be Spoiler's last move. For Spoiler to win, he needs to place his pebble such that Duplicator cannot duplicate the order of pebbles in the other structure. So, let's assume he puts his pebble, s_3 between the pebbles in U_7:

$$
\begin{array}{ccccccccc}
 & & & & U_7 & & & & \\
[& 0 & 0 & 0 & 0 & 0 & 0 & 0 &] \\
 & & & & \uparrow & \uparrow & \uparrow & & \\
 & & & & s_1 & s_3 & d_2 & &
\end{array}
\qquad
\begin{array}{cccccccccc}
 & & & & U_8 & & & & & \\
[& 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &] \\
 & & & & \uparrow & & & \uparrow & & \\
 & & & & d_1 & & & s_2 & &
\end{array}
$$

To keep up the correspondence, Duplicator must play between the pebbles on U_8:

$$
\begin{array}{ccccccccc}
 & & & & U_7 & & & & \\
[& 0 & 0 & 0 & 0 & 0 & 0 & 0 &] \\
 & & & & \uparrow & \uparrow & \uparrow & & \\
 & & & & s_1 & s_3 & d_2 & &
\end{array}
\qquad
\begin{array}{cccccccccc}
 & & & & U_8 & & & & & \\
[& 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &] \\
 & & & & \uparrow & \uparrow & & \uparrow & & \\
 & & & & d_1 & d_3 & & s_2 & &
\end{array}
$$

The substructures induced by the pebbles are isomorphic. So, Duplicator wins this 3-move game. In general, Duplicator's strategy at the ith round of a k-move game is if Spoiler plays within $2^{(k-i)}$ of a placed pebble, then she plays her pebble with that exact distance to the corresponding pebble in her structure. If Spoiler plays farther than $2^{(k-i)}$ from any pebble, then Duplicator plays "far" (greater than $2^{(k-i)}$ at the ith round) from any pebble (but preserving the order).

If the game was to go on for just one more, Spoiler could win. To win, Spoiler plays his fourth pebble, s_4, between d_3 and s_2 on U_8:

$$
\begin{array}{ccccccccc}
 & & & & U_7 & & & & \\
[& 0 & 0 & 0 & 0 & 0 & 0 & 0 &] \\
 & & & & \uparrow & \uparrow & \uparrow & & \\
 & & & & s_1 & s_3 & d_2 & &
\end{array}
\qquad
\begin{array}{cccccccccc}
 & & & & U_8 & & & & & \\
[& 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &] \\
 & & & & \uparrow & \uparrow & \uparrow & \uparrow & & \\
 & & & & d_1 & d_3 & s_4 & s_2 & &
\end{array}
$$

To maintain the correspondence, Duplicator must play on a distinct element between s_3 and d_2 in U_7. Since there is no such element, Spoiler wins this 4-move game. Further analysis shows that Spoiler can always win the 4-move game played on U_7 and U_8, and we write: $U_7 \not\equiv_4 U_8$. In general, Spoiler's strategy is to show the difference in length by splitting the smallest distance between any two placed pebbles or endpoints. If one of the structures has length less than $2^k - 1$, then Spoiler can show the difference within k moves.

Duplicator can win any k-move game on structures of size larger than 2^k. So, $T_\epsilon(n) \geq \log_2 n$. We also have that if the structures differ in size and one is smaller than $2^k - 1$, then Spoiler has a winning strategy. This gives that Duplicator loses on such structures, and $T_\epsilon(n) = O(\log_2 n)$. Thus, $T_\epsilon(n) = \Theta(\log_2 n)$.

4 Very sparse

As in the section above, we will focus on random bit strings, but a Zero-One Law holds for both bit strings and circular sequences, as do the results of this section.

For $n^{-1} \ll p \ll n^{-1/2}$, we shall see that the resulting structures almost surely have arbitrarily many ones, and the ones that do occur are isolated from one another by arbitrarily many zeros. The players' strategies for Ehrenfeucht-Fraisse game become much more interesting. As in Section 3, Spoiler can win the k-move game if the lengths of the structures are different and at least one has length less than $2^k - 1$. So, we have an upper bound, $T_\epsilon(n) = O(\log_2 n)$. We can improve this bound by examining the conditions which give a winning strategy for Duplicator.

Before giving sufficient conditions for Duplicator, let's look at a game on two structures, U and V:

$$\begin{array}{cc} U & V \\ [000100100001000100] & [000010000010001000] \end{array}$$

Spoiler's goal is illustrate the differences between the two structures. Duplicator's strategy is two-fold, depending on whether Spoiler plays on a one or zero. If he played on a one, then Duplicator must also play a one. If the moves are restricted to the elements that are ones, then we can view the game as occurring on an ordered structure of identical elements, as in Section 3. The strategies from Section 3 serve both players well, and if the number of ones in both structures is greater than 2^k, Duplicator wins the k-move game played only on the ones.

If Spoiler plays on a zero, Duplicator must play on a zero, with the same relative position as Spoiler's move. For example, assume Spoiler placed his pebble, s_1, two elements before the second one:

$$\begin{array}{ccccc} U & & & V & \\ [000100100001000100] & \quad & [00001000 & 0 & 010001000] \\ & & & \uparrow & \\ & & & s_1 & \end{array}$$

Let's view Spoiler's move as actually three moves: the placement of s_1, plus the placement of two more "shadow" pebbles, S_1 and S_2 on ones nearest to Spoiler's move:

$$\begin{array}{ccccccccc} U & & & & & V & & & \\ [000100100001000100] & \quad & [0000 & 1 & 000 & 0 & 0 & 1 & 0001000] \\ & & & \uparrow & & \uparrow & & \uparrow & \\ & & & S_1 & & s_1 & & S_2 & \end{array}$$

These extra shadow pebbles are used only by Duplicator to determine her next move and are not a part of the actual game. For Duplicator's next move, we first, using the strategy for playing on the ones from Section 3, place the "shadow" pebbles, D_1 and D_2:

$$\begin{array}{ccccc} & & & U & \\ [000 & 1 & 00 & 1 & 00001000100] \\ & \uparrow & & \uparrow & \\ & D_1 & & D_2 & \end{array} \qquad \begin{array}{ccccccc} & & & V & & & \\ [0000 & 1 & 000 & 0 & 0 & 1 & 0001000] \\ & \uparrow & & \uparrow & & \uparrow & \\ & S_1 & & s_1 & & S_2 & \end{array}$$

The placement of the "real" pebble now reduces to a game on the strings of zeros between the shadow pebbles: [00] in U and [00000] in V. If these subintervals have length at least 2^i, where i is the number of moves left in the game, than Duplicator has a winning strategy on this substructure.

To summarize the discussion above: Duplicator has a winning strategy for the k-move game if:

1. The lengths of both structures are greater than 2^k.
2. The number of ones in each structure is greater than 2^k.
3. The distance between any two ones in each structure is greater than 2^k. Also, the distance between any one and an endpoint is greater than 2^k.

In [7], these conditions for Duplicator to win the k-move game were shown to be necessary (following from the definition of "k-types").

Given n, for what k will these conditions hold $1-\epsilon$ of the time? We have already addressed the first condition above. We would like the second and third condition to hold at least $1-\epsilon$ amount of the time. For the second condition to hold, we need that the number of ones is greater than 2^k at least $1-\epsilon$ of the time. Let be the random variable X count the number of ones that occur. The expectation, $\mu = E[X]$, is np, and the variance, $\sigma^2 = \mathrm{Var}[X] = np(1-p)$. As $\sigma = o(\mu)$, by Chebyshev's Inequality, $X > \mu/2$, almost surely (see, for example, [1]). So, the second condition, that there is at least 2^k ones, is satisfied if $k \leq \log_2(\mu/2) = \log_2(np) - O(1)$.

To guarantee the third condition, that the ones occur far apart (at distances greater than 2^k), we let D be the random variable for the number of pairs of ones within distance 2^k. The expectation, $E[D] \leq n2^kp^2$. When $k = -\log_2(np^2) - O(1)$, $E[D] < \epsilon$, so the third condition holds with probability at least $1-\epsilon$.

This gives $T_\epsilon(n) = O(\min(\log_2(np), -\log_2(np^2)))$. To illustrate why the minimum cannot be removed from the above equation, we focus on the probability functions of the form, $p = n^{-\alpha}$ for $\frac{1}{2} < \alpha < 1$. Figure 3 contains the graph of the $y = \log_2(np) = \log_2(n^{1-\alpha})$ and $y = -\log_2(np^2) = \log_2(n^{2\alpha-1})$. For $\alpha > \frac{2}{3}$, the minimum is attained by $\log_2(np)$, and for $\alpha < \frac{2}{3}$, $-\log_2(np^2)$ provides the better bound.

5 $p \gg n^{-1/2}$

Throughout this section we assume $p \gg n^{-1/2}$ and $1-p \gg n^{-1/2}$. In this realm, consecutive pairs 11 (and their counterpart 00) occur arbitrarily often. Our result will be the upper bound $T_\epsilon = O(\log^* n)$. We look, therefore, only at Spoiler strategies for the game played on circular sequences C_n, C_m.

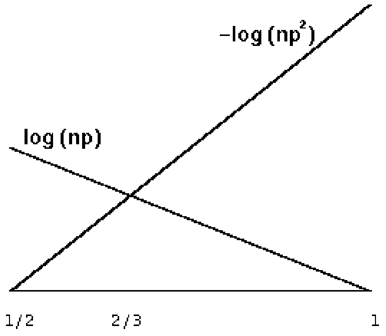

Figure 3: The functions, $\log_2(np)$ and $-\log_2(np^2)$ for $p = n^{-\alpha}$ and $\frac{1}{2} < \alpha < 1$.

Spoiler begins (see Figure 4) by playing $s_1, s_2 \in C_m$ to which Duplicator responds by playing some $d_1, d_2 \in C_n$ Thereafter, all Spoiler's plays in C_m will be in $[s_1, s_2]$ and all his plays in C_n will be in $[d_1, d_2]$. (Formally, $[s_1, s_2]$ is s_1, s_2 and those x with $cc(s_1, x, s_2)$.) As Duplicator must preserve the ordering, cc, she must also play in these intervals. Thus, it will suffice to show that Spoiler can win the EF game on these intervals. To simplify the arguments, we will encode the bit strings into strings of As and Bs, with 11 encoding to A and 101 to B (see Section 5.1.1). We show that it suffices for Spoiler to win a game on the encoded strings (see Theorem 1).

We show in Section 5.2 that all patterns w of $\{A, B\}$ of fixed length occur almost surely in the encoding of some interval of $C_{n,p}$. In Section 5.1.2 we show that there are $w \in \{A, B\}^*$ that can be distinguished from any $w' \in \{A, B\}^*$ of length at most $Tower(k)$ in approximately $5k$ moves (the function $Tower(k)$ is defined inductively as $Tower(0) = 1$ and $Tower(k+1) = 2^{Tower(k)}$ for $k > 0$). This leads us to the Main Result (Theorem 5 in Section 5.3) that $T_\epsilon(n) = O(\log^* n)$. For very large m, the random C_m with high probability has a subinterval that encodes to the pattern w above. Spoiler picks that subinterval in C_m. For all structures of size n or smaller, Spoiler will win because the second structure is just not large enough to contain a pattern w' indistinguishable from w.

5.1 Logic

In this section, we define the encoding function, ENC, and show that if Spoiler has a winning strategy for the the k-move game on the encoded structures, then he has a winning strategy for the $(k+2)$-move game on the original structures.

The rest of this section focuses on showing that there are $Tower(k)$ different structures that can be distinguished by the $(5k-8)$-move game. From this, we find a sequence w_0 which can be distinguished from all sequences w of length at most $Tower(k)$ in only $5k-3$ moves.

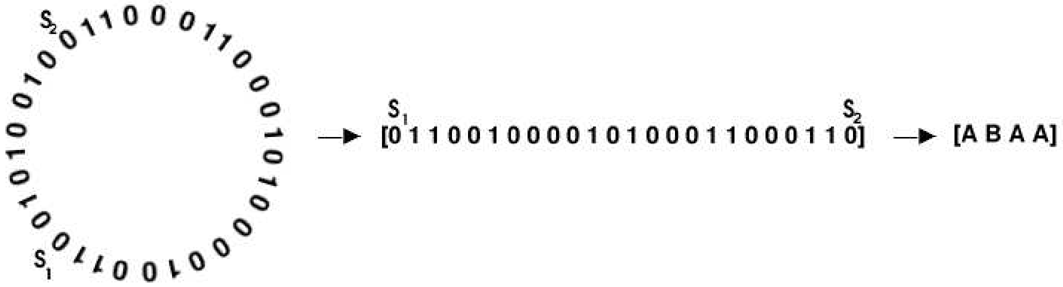

Figure 4: Spoiler first chooses an interval from the circular sequence. The string is then encoded into a string of A's and B's.

5.1.1 Encoding

To get our results, it will suffice to focus on just two pairs of ones: $A =$"11" and $B =$"101", and ignore all other occurrences of pairs. We show first that given two intervals, w_1 and w_2, if Spoiler wins the k-move game played on encoded sequences, $ENC(w_1)$ and $ENC(w_2)$, then Spoiler will win the $(k+2)$-move game played on the original sequences, w_1 and w_2.

We define an encoding function $ENC : \{0,1\}^* \to \{A,B\}^*$ as follows. Each 1 followed by 1 is encoded as A, each 1 followed by 01 is encoded as B, all other symbols are ignored.

Theorem 1 *: Let $w_1, w_2 \in \{0,1\}^*$. If $ENC(w_1) \not\equiv_k ENC(w_2)$ then $w_1 \not\equiv_{k+2} w_2$.*

Proof: If $ENC(w_1) \not\equiv_k ENC(w_2)$, then Spoiler has a winning strategy on the k-move game played on the encoded structures of strings of $\{A,B\}^*$. Let S_i be Spoiler's *ith* move, following his winning strategy, on the encoded structures, $ENC(w_1)$ and $ENC(w_2)$. Translate Spoiler's ith move, S_i, to a move on the original structure, s_i, by playing the first element of the corresponding pair. For example, if Spoiler moved on the third element of $ABABAAB$ in the encoded structure, then his move, s_i, on the original structure would be the first one that occurs in the second pair of "11." Duplicator responds with her move, d_i, on the other original structure. Her move must be on a one for the game to continue. Moreover, if Spoiler's move on the encoded structures was an A, then Duplicator must have played the first one of a pair "11", and if Spoiler played a B, the Duplicator must play in the original structure on the first one of "101." If not, in an additional two moves, Spoiler can distinguish "11" or "101" from the string Duplicator.

Thus, with an additional two moves in the original structures, we can distinguish the strings, and Spoiler wins the $(k+2)$-move game on the original structures. □

5.1.2 Equivalence Classes

The k-move Ehrenfeucht-Fraisse game induces equivalence classes on the structures. We will say that two structures $\mathcal{M}_1$ and $\mathcal{M}_2$ are **equivalent** (with respect to k), if Duplicator has a winning strategy for the k-move game, $\mathcal{M}_1 \equiv_k \mathcal{M}_2$. A well-known, but not obvious fact, is that the number of equivalence classes with respect to the k-move game is finite (see [3] for details).

Theorem 2 *The number of* $\equiv_{(5k-8)}$*-classes is greater than* $Tower(k)$ *for* $k \geq 2$*.*

Proof: We proceed by induction on k. For the base case $k = 2$ we observe that the bit strings $00, 01, 10, 11$ are mutually inequivalent with respect to $\equiv_2$, hence there are at least $Tower(2) = 4$ $\equiv_2$-classes.

For the inductive step, assume there are more than $Tower(k)$ equivalence classes for the $(5k-8)$-move game, and show that there are more than $Tower(k+1)$ equivalence classes for the $(5(k+1)-8)$-move game. We begin by expanding the alphabet to include a marker, M, to simplify the arguments.

By inductive hypothesis, there exist more than $Tower(k)$ equivalence classes for the $(5k-8)$-move game. Let $E = \{e_1, e_2, \ldots, e_T\}$ be representatives of these $T > Tower(k)$ classes. For any subset, $A = \{a_1, a_2, \ldots, a_m\} \subseteq E$, let w_A be the string:

$$Ma_1Ma_2M\ldots Ma_mM$$

where M is a marker. This string belongs to an extended alphabet of $\{0, 1, M\}$.

We claim that for any two subsets $A = \{a_1, a_2, \ldots, a_m\}, B = \{b_1, b_2, \ldots, b_l\} \subseteq E$, if $A \neq B$, then $w_A \not\equiv_{(5k-6)} w_B$. We need to show Spoiler wins the $(5k-6)$-move game. If $A \neq B$, then, without loss of generality, there exists $a' \in A$ such that $a' \notin B$. Assume Spoiler plays the markers on either side of $a' \in A$:

$$\begin{array}{ccccccc} & U & & & & & V \\ [Ma_1Ma_2M\ldots & M & a' & M & \ldots a_mM] & \quad & [Mb_1Mb_2M\ldots Mb'M\ldots b_mM] \\ & \uparrow & & \uparrow & & & \\ & S_1 & & S_2 & & & \end{array}$$

Duplicator must respond by playing markers on w_B that surround some element of B, say b'. By choice, $a' \notin B$, and $a' \neq b'$. Since a' and b' are distinct representatives of the $\equiv_{(5k-8)}$ equivalence classes, this gives that $a' \not\equiv_{(5k-8)} b'$. Spoiler has a winning strategy for the $(5k-6)$-move game on the intervals w_A and w_B.

The above argument is for the extended alphabet of $\{0, 1, M\}$. Instead of the extra symbol M, we could have been encoded the three letter alphabet by sequence of zeros and ones. For example, $f : \{0, 1, M\} \to \{0, 1\}^*$ such that $f(0) =$"1", $f(1) =$ "10," and $f(M) =$ "100." If we now play the game on these strings in $\{0, 1\}$, we follow the strategy above with the following addition: if Spoiler plays i in the structures on the three letter alphabet, then Spoiler now plays the the first element of the encoded substring, $f(i)$. In the worst case (where Duplicator plays on a different encoded symbol on the $(5k-6)$th move), Spoiler needs three more moves to distinguish the encoded symbols. Thus, Spoiler needs $5k - 3 = 5(k + 1) - 8$ moves to distinguish any two subsets. So, the number of $\equiv_{(5(k+1)-8)}$ equivalence classes is more than $Tower(k + 1)$. □

Corollary 1 *There exists a sequence,* $w_0 \in \{0, 1\}^*$ *such that for all sequences,* w*, if* $w \equiv_{5k-3} w_0$*, then* $|w| > Tower(k)$*.*

Proof: Let $E = \{e_1, e_2, \ldots, e_T\}$ be representatives of the equivalence classes for the $(5k-8)$-move game. We first expand the alphabet to $\{0, 1, M\}$ and let $w_0^* = [Me_1Me_2M \ldots Me_TM]$ the concatenation of all the $\equiv_{(5k-8)}$ representatives. Then let $w_0 = f(w_0^*) \in \{0,1\}^*$ with the encoding f given in Theorem 2.

For any $w \equiv_{(5k-3)} w_0$, w must contain a representative of each $\equiv_{(5k-8)}$ equivalence class. Otherwise, Spoiler could distinguish w and w_0 in $(5k-3)$ moves using the strategy outlined in the proof of Theorem 2. The length of w is larger than the number of representatives of $\equiv_{(5k-8)}$ classes (since it contains all of them) which is greater than $Tower(k)$, by Theorem 2. □

5.2 Probability

We show that all patterns of $\{A, B\}$ of fixed length occur with positive probability in the encoding of $C_{n,p}$. We begin by showing that any string of $\{A, B\}$ occurs with positive probability.

Theorem 3 *Let $p = p(m) \sim m^{-1/2}$. Let $\vec{v} = X_1 \cdots X_s \in \{A, B\}^*$. Then the probability that $ENC(U_{m,p}) = \vec{v}$ is at least $\frac{1}{e^2 s!}(1 - o(1))$. The asymptotics are for fixed $\vec{v}$ as $m \to \infty$.*

Proof: For any $1 \le u_1 < \ldots < u_s \le m - 4$ with all $u_{i+1} - u_i \ge 5$, consider the event $E[\vec{v}, \vec{u}]$ that

- When $X_i = A$, the U sequence starting at u_i goes 11000.
- When $X_i = B$, the U sequence starting at u_i goes 10100.
- There are no strings 11 nor 101 in the U string other than these s.

Then $E[\vec{v}, \vec{u}]$ has probability at least $[p^2(1-p)^3]^s(1-p^2)^{2m}$. The first term is the probability of those particular $5s$ entries. For the second term we have less than $2m$ potential pairs of ones adjacent or one apart and each has probability $1-p^2$ of not being there. From the FKG inequality (the easy part of Janson's Inequality, see [1]) the probability that none are there is at least $1-p^2$ to the power which is the number of those events. The events $E[\vec{v}, \vec{u}]$ are disjoint over all $\vec{u}$ and the number of $\vec{u}$ is $\sim m^s/s!$. Hence the event that some $E[\vec{v}, \vec{u}]$ holds is at least asymptotically

$$\frac{m^s}{s!}[p^2(1-p)^3](1-p^2)^{2m} \sim \frac{1}{e^2 s!}$$

□

Theorem 4 *Let $p = p(n)$ satisfy $n^{-1/2} \ll p(n) \le 1/2$. Let $\vec{v} = X_1 \cdots X_s \in \{A, B\}^*$ be fixed. Then almost surely $C_{n,p}$ has a substring that encodes to $\vec{v}$.*

Proof: First, assume $p(n) = o(1)$. Set $m = \lfloor p^{-2} \rfloor$ so that $m \to \infty$ and $m = o(n)$. Split $C_{n,p}$ into $\lfloor n/m \rfloor$ disjoint strings of length m. Each encodes to $\vec{v}$ with probability $\sim \frac{1}{e^2 s!}$. They are independent so the probability that none encodes to $\vec{v}$ is at most $(1 - \frac{1}{e^2 s!})^{\lfloor n/m \rfloor}$ which goes to zero. Finally, if $p(n)$ is bounded from below by a positive constant then almost surely $C_{n,p}$ contains any particular substring and hence one that encodes to $\vec{v}$. □

5.3 Main Result

Theorem 5 *Assume $p(n) \gg n^{-1/2}$ and $1 - p(n) \gg n^{-1/2}$ and $\epsilon > 0$ is fixed. If $n \leq Tower(k)$, then $T_\epsilon(n) < 5k+1$. In particular, $T_\epsilon(n) = O(\log^* n)$.*

Proof: Fix k (and don't worry about n at this point). Let w_0 be the string from Corollary 1. So, w_0 has the property that if $w \equiv_{5k-3} w_0$, then $|w| > Tower(k)$. Let $\vec{v}$ be the encoding of w_0 into $\{A, B\}^*$ described in Section 5.1.1. Note that $\vec{v}$ is a fixed string. So, by Theorem 4, for sufficiently large m, C_m contains a substring encoding w_0 with probability at least $1 - \epsilon$. The choice of m depends on k and w_0, but not on n.

Now, to show $T_\epsilon(n) < 5k+1$ is the same as showing that for some $n_1, n_2 \geq n$, Spoiler wins the $(5k+1)$-move game on independent structures of size n_1 and n_2 with probability at least ϵ. Since we need to show this only for some n_1 and n_2, let $n_1 = m$ (from the application of Theorem 4 above), and let $n_2 = n$.

Consider the $(5k+1)$-move Ehrenfeucht-Fraisse game with such a C_m and *any* C_n. Spoiler first selects $s_1, s_2 \in C_m$ so that $[s_1, s_2]$ encode to w_0. Duplicator selects $d_1, d_2 \in C_n$ and $[d_1, d_2]$ encodes to some w. Since $w \subseteq C_n$, $|w| \leq n$. n is less than $Tower(k)$, so $w \not\equiv_{5k-3} w_0$ by the construction of w_0 in Corollary 1.

From Theorem 1, Spoiler distinguishes the bit strings on $[s_1, s_2]$ and $[d_1, d_2]$ in the $5k - 3 + 2$ remaining moves. This gives $C_m \not\equiv_{5k+1} C_n$ for every C_n. By Theorem 4, C_m has the desired substring w_0 with probability $1 - \epsilon$. So, $C_m \not\equiv_{5k+1} C_n$ with probability $1 - \epsilon > \epsilon$ (for $\epsilon < \frac{1}{2}$). This gives that $T_\epsilon(n) < 5k+1$. Since $n \leq Tower(k)$, this implies $T_\epsilon(n) = O(\log^* n)$. □

6 Conclusion and Future Work

We give bounds for roughly how quickly the Zero-One Laws converge for random bit strings and random circular bit sequences. We show that for random bit strings and random circular sequences of length n generated with a low probability ($p \ll n^{-1}$), $T_\epsilon(n) = \Theta(\log_2 n)$. For random bit strings and circular sequences with isolated ones ($n^{-1} \ll p \ll n^{-1/2}$), $T_\epsilon(n) = O(\min(\log_2 np, -\log_2 np^2))$. For $n^{-1/2} \ll p$ and $(1-p) \ll n^{-1/2}$, we show that $T_\epsilon(n) = O(\log^* n)$ for random circular sequences.

While we have restricted our attention to random bit strings the tenacity function can be defined whenever there is a Zero-One Law. Can we get similar bounds on the tenacity function for other random structures such as random graphs and random trees?

For the last case we considered, where $n^{-1/2} \ll p$ and $(1-p) \ll n^{-1/2}$, we achieved an upper bound for all such p and all $\epsilon > 0$. The question naturally arising is whether one can get matching lower bounds. One needs some caution, for some p, such as $n^{-1/3}$, the Zero-One Law itself fails so that $T_\epsilon(n)$ does not approach infinity. Further, one might be extremely close to a threshold function. If $p = n^{-1/3}g(n)$ where $g(n)$ approaches infinity extremely slowly, say the inverse Ackermann function, then certainly $T_\epsilon(n)$ would be of lower order. Still, it seems reasonable to conjecture, for example, that if $p = n^{-\alpha}$ and the Zero-One Law holds then $T_\epsilon(n)$ is of the order $\log^*(n)$.

References

[1] Noga Alon, Joel H. Spencer, and Paul Erdős. *The Probabilistic Method.* John Wiley and Sons, Inc, New York, 1992.

[2] Peter Dolan. A zero-one law for a random subset. *Random Structures and Algorithms*, 2:317–326, 1991.

[3] Heinz-Dieter Ebbinghaus and Jorg Flum. *Finite Model Theory.* Springer, Berlin, 1995.

[4] P. Erdős and A. Rényi. The evolution of random graphs. *Magyar Tud. Akad. Mat. Kut. Int. Kőzl*, 5:17–61, 1960.

[5] Saharon Shelah and Joel H. Spencer. Random sparse unary predicates. *Random Structures and Algorithms*, 5(3):375–394, 1994.

[6] Joel H. Spencer. Threshold spectra via the Ehrenfeucht game. *Discrete Applied Mathematics*, 30:235–252, 1991.

[7] Joel H. Spencer and Katherine St. John. Random unary predicates: Almost sure theories and countable models. *Random Structures and Algorithms*, 13(#3/4), 1998.