On Lempel-Ziv Complexity of Sequences

Ali Doğanaksoy[1,2,4] and Faruk Göloğlu[2,3]

[1] Department of Mathematics, Middle East Technical University
Ankara, Turkey
aldoks@metu.edu.tr
[2] Institute of Applied Mathematics, Middle East Technical University
Ankara, Turkey
[3] Dept. of Computer Technology and Information Systems
Bilkent University, Ankara, Turkey
gologlu@bilkent.edu.tr
[4] TUBITAK-UEKAE, Gebze, Turkey

Abstract. We derive recurrences for counting the number $a(n, r)$ of sequences of length n with Lempel-Ziv complexity r, which has important applications, for instance testing randomness of binary sequences. We also give algorithms to compute these recurrences. We employed these algorithms to compute $a(n, r)$ and expected value, EP_n, of number of patterns of a sequence of length n, for relatively large n. We offer a randomness test based on the algorithms to be used for testing randomness of binary sequences. We give outputs of the algorithms for some n. We also provide results of the proposed test applied to the outputs of contestant stream ciphers of ECRYPT's eSTREAM.

Keywords: Lempel-Ziv complexity, randomness, χ^2-statistics.

1 Introduction

There are several complexity measures to test the randomness of a sequence. Linear complexity, for example, is one of these measures. Lempel-Ziv complexity of a sequence was defined by Lempel and Ziv in 1976 [1]. This measure counts the number of different patterns in a sequence when scanned from left to right. For instance Lempel-Ziv complexity of $s = 101001010010111110$ is 8, because when scanned from left to right, different patterns observed in s are 1|0|10|01|010|0101|11|110|.

Lempel-Ziv complexity is the basis of LZ77 compression algorithm [2]. It is also an important measure used in cryptography. For instance, it was used to test the randomness of the output of a symmetric cipher [3]. One expects a 'random' sequence of length n has a close Lempel-Ziv complexity to the expected value of Lempel-Ziv complexity of a sequence of length n. However, the expected value of Lempel-Ziv complexity for arbitrary n is unknown. For limiting behaviour of this value, the reader is referred to Jacquet and Szpankowski [4] and Kirschenhofer et. al. [6]. Some cryptographic applications of Lempel-Ziv complexity are given in [5].

G. Gong et al. (Eds.): SETA 2006, LNCS 4086, pp. 180–189, 2006.

Some sequences end with a pattern that was observed before (one simplest example is: $s = 0|0$), which we call *open*; and remaining sequences (i.e., that end without same pattern appearing twice) are called *closed*.

In this paper we derive a recurrence for $a(n, r)$, the number of sequences of length n with Lempel-Ziv complexity r; and a recurrence for $c(n, r)$, the number of closed sequences of length n with Lempel-Ziv complexity r. By using these recurrences and with the help of a computer, we compute $a(n, r)$ for as large n as possible.

A test based on Lempel-Ziv complexity was used in the NIST test suite, to test the randomness of sequences. However the test had some weaknesses. First of all, the test could only be applied to data of a specified length: 10^6 bits. Moreover, the test used empirical data generated by SHA-1 (under randomness assumptions) for estimating the expected value of Lempel-Ziv complexity of sequences of length 10^6 bits. Apparently, the data generated by SHA-1 led to not-so-good an estimate, hence, for instance, first 10^6 bits of the binary expansion of e failed the randomness test. Using asymptotic formulae for an estimate will not work either, since the sequences, as we will see in the forthcoming sections, are distributed tightly around the mean. Recently, apparently because of the spelt out reasons, Lempel-Ziv test had been excluded from the NIST test suite. Inclusion of a Lempel-Ziv complexity based randomness test in a statistical test suite is important concerning completeness. In the last section, we offer a new and stronger variant of this test, which employs the results we found and present in this paper. The data we use are neither empirical nor derived from asymptotic formulae, but are exact results; thanks to the recurrences (1),(2), hence avoid the errors present in the previous test.

2 Preliminaries

Lempel-Ziv complexity was first defined in [1]. We include the definitions here. For the sequel, juxtaposition denotes concatenation of strings.

Let $p = p_1 p_2 \cdots p_k$ and $s = s_1 s_2 \cdots s_k \cdots s_n$ be binary strings. p is a *prefix* of s if $p_i = s_i$ for $1 \leq i \leq k$. If $k < n$, then p is said to be a *proper prefix* of s.

Let again $s = s_1 s_2 \cdots s_n$ be a binary string of length n. $\sigma_1|\cdots|\sigma_r$ is called the *Lempel-Ziv partition* of s, if

- for $1 \leq i < r$, σ_i is different from σ_j for $0 \leq j < i$, satisfying
- $s = \sigma_1\sigma_2\cdots\sigma_r$, and
- for $1 \leq i \leq r$, every proper prefix of σ_i is equal to σ_j for some $0 \leq j < i$.

where σ_i are binary strings (*patterns*) and σ_0 is defined to be the empty string.

Lempel-Ziv complexity of s is then defined to be the number of patterns, r, in the Lempel-Ziv partition of s.

Note that σ_r may or may not satisfy $\sigma_r = \sigma_i$ for some $1 \leq i < r$. If $\sigma_r = \sigma_i$ for some $1 \leq i < r$, then we call s an *open* sequence. s is called *closed* otherwise.

Lempel-Ziv partition of:

- an open sequence s is denoted by $s = \sigma_1|\cdots|\sigma_r$,
- a closed sequence s is denoted by $s = \sigma_1|\cdots|\sigma_r|$.

Succint background for statistical tests (especially for randomness) can be found in [3].

3 The Recurrences

Let $A(n,r)$ denote the set of binary strings of length n with Lempel-Ziv complexity r. For any $s = s_1 \cdots s_n \in A(n,r)$ and $s_{n+1} \in \{0,1\}$, it is evident that $ss_{n+1} \in A(n+1,r) \cup A(n+1,r+1)$. In fact $s0 \in A(n+1,r) \iff s1 \in A(n+1,r)$. We define

$$C(n,r) = \{s \in A(n,r) : s0 \in A(n+1,r+1)\} .$$

Note that $C(n,r)$ is the set of *closed* sequences. One has

$$a(n,r) = 2c(n-1,r-1) + 2\left[a(n-1,r) - c(n-1,r)\right] , \tag{1}$$

where $a(n,r) = |A(n,r)|$ and $c(n,r) = |C(n,r)|$.

Given $s = s_1 \cdots s_n \in C(n,r)$, let $\sigma_1|\ldots|\sigma_r|$ be the Lempel-Ziv partition of s. We define the mapping $\delta^0_{n,r} : C(n,r) \to C(n+r+1, r+1)$ by setting $\delta^0_{n,r}(s) = 00\sigma_1 0\sigma_2 \cdots 0\sigma_r$ for $s = \sigma_1 \cdots \sigma_r \in C(n,r)$. $\delta^1_{n,r}$ is defined in a similar way. Let $C_0(n,r) = \mathrm{Im}(\delta^0_{n-r,r-1})$, $C_1(n,r) = \mathrm{Im}(\delta^1_{n-r,r-1})$, and $C_*(n,r) = C_0(n,r) \cup C_1(n,r)$. It follows that $c_*(n,r) = c_0(n,r) + c_1(n,r) = 2c(n-r, r-1)$, where $c_*(n,r) = |C_*(n,r)|$, $c_0(n,r) = |C_0(n,r)|$, $c_1(n,r) = |C_1(n,r)|$, and $\mathrm{Im}(f)$ denotes the image of the map f.

Any $s = \sigma_1|\cdots|\sigma_r| \in C(n,r) \setminus C_*(n,r)$ has a unique substring $\alpha = \alpha_1|\cdots|\alpha_p| \in C_0(a,p)$, and a unique substring $\beta = \beta_1|\cdots|\beta_q| \in C_1(b,q)$ such that $a+b = n$ and $p+q = r$.

For any pair (p,q) of positive integers, we consider the subset $\Xi^{p,q}$ of the symmetric group S^{p+q} given by:

$$\Xi^{p,q} = \left\{\sigma \in S^{p+q} : i < j \le p \text{ or } p+1 \le i < j \Rightarrow \sigma(i) < \sigma(j)\right\} .$$

For $\alpha = \alpha_1|\cdots|\alpha_p| \in C_0(a,p)$, $\beta = \beta_1|\cdots|\beta_q| \in C_1(b,q)$ and $\pi \in \Xi^{p,q}$, $\pi(\alpha,\beta)$ stands for $\pi(\alpha_1,\ldots,\alpha_p,\beta_1,\ldots,\beta_q)$.

Any triple (π,α,β), where $\pi \in \Xi^{p,q}, \alpha \in C_0(a,p), \beta \in C_1(b,q)$, corresponds to a unique string in $C(n,r) \setminus C_*(n,r)$, namely to $\pi(\alpha,\beta)$. Conversely given any $\sigma \in C(n,r) \setminus C_*(n,r)$, there exist a unique triple (π,α,β), such that $\pi(\alpha,\beta) = \sigma$.

Given a, b, p and q, the number of all possible triples (π,α,β) with $\pi \in \Xi^{p,q}, \alpha \in C_0(a,p), \beta \in C_1(b,q)$ is

$$\binom{p+q}{p} c_0(a,p) c_1(b,q) .$$

It follows that

$$\begin{aligned} c(n,r) - c_*(n,r) &= \sum_{a+b=n} \sum_{\substack{p+q=r \\ p,q\ge 1}} \binom{p+q}{p} c_0(a,p) c_1(b,q) \\ &= \sum_{a+b=n} \sum_{\substack{p+q=r \\ p,q\ge 1}} \binom{p+q}{p} c(a-p, p-1) c(b-q, q-1) \end{aligned}$$

$$\Rightarrow c(n,r) = 2c(n-r,r-1)+ \sum_{a+b=n}\sum_{\substack{p+q=r\\ p,q\geq 1}} \binom{p+q}{p} c(a-p,p-1)c(b-q,q-1)$$

$$= 2c(n-r,r-1)+ \sum_{0\leq a\leq n}\sum_{1\leq p<r} \underbrace{\binom{r}{p} c(a-p,p-1)c(n-a-r+p,r-p-1)}_{\tau(n,r,a,p)} \tag{2}$$

We can give upper and lower bounds for r, since not all r are possible given any n. Indeed, observing $s = 0|00|000|\cdots$ has minimum complexity, and

$$s = 0|1|00|01|10|11|000|001|010|011|\cdots$$

has maximum complexity among all sequences of length n, we limit r by:

$$\left\lceil \frac{-1+\sqrt{1+8n}}{2} \right\rceil \leq r \leq \left\lceil \frac{2^{t+2}+n-2t-4}{t+1} \right\rceil \tag{3}$$

where $t = \mathsf{max}\left\{i \in \mathbb{N} : (i-1)2^{i+1}+2 \leq n\right\}$. Note here that r is bounded by $r < k\frac{n}{\log n}$, for some $k \in \mathbb{N}$. Indeed, $t < \log n$ for all $n \geq 2$. Also

$$\left\lceil \frac{2^{t+2}+n-2t-4}{t+1} \right\rceil = \left\lceil \frac{2^{t+2}-2(t+2)}{t+1} + \frac{n}{t+1} \right\rceil$$

increases when t increases, hence

$$r \leq \left\lceil \frac{2^{t+2}+n-2t-4}{t+1} \right\rceil \leq \left\lceil \frac{4\cdot 2^{\log n}+n-2\log n-4}{\log n+1} \right\rceil < 5\frac{n}{\log n}\,.$$

4 Algorithms and Their Complexities

(1) implies computing $c(n,r)$ for all $k \leq n$, and knowing $a(1,1) = 2$, is enough to compute $a(n,r)$ for any $n \geq 2$. Therefore we use (2) to compute $c(n,r)$, the result of which is used by another algorithm to compute $a(n,r)$. However, it is inefficient to compute larger values (e.g., computing $a(2000,r)$ for all r takes two hours on a standard PC with our implementation). We use the recurrence (2) in the following algorithm.

```
COMPUTE-C(N,R)(N)
1  c(1,1) ← 2
2  for n ← 2 to N
3     do for r ← r_l(n) to r_u(n)
4           do c(n,r) ← 2c(n−r,r−1)
5              for a ← 0 to n
6                 do for p ← 1 to r
7                       do c(n,r) ← c(n,r) + τ(n,r,a,p)
```

After we compute all $c(n,r)$ for $n < N$, we use the following algorithm which is based on the recurrence (1) to compute $a(n,r)$.

COMPUTE-A(N,R)$(N, c(i,j))$
1 $a(1,1) \leftarrow 2$
2 **for** $n \leftarrow 2$ **to** N
3 **do for** $r \leftarrow r_l(n)$ **to** $r_u(n)$
4 **do** $a(n,r) \leftarrow 2c(n-1,r-1) + 2\,[a(n-1,r) - c(n-1,r)]$

In the algorithms, r_l and r_u are computed by the inequalities (3).

We have the following observations for the complexity of the algorithms. For any (n,r) pair, $c(n,r) < 2^n$, hence an (at most) n-bit integer. Since $r \leq k \cdot n/\log n$, and complexity of multiplication of two n bit integers is $\mathcal{O}(n \log n)$ we have :

Proposition 1. *Complexity of the algorithm:*

- COMPUTE-C(N,R) *is* $\mathcal{O}(n^5/\log n)$*, and*
- COMPUTE-A(N,R) *is* $\mathcal{O}(n^2/\log n)$ *(after computing* $c(n,r)$*).*

5 Computing $a(n,r)$ for Large n

Tables 1 and 2 in Appendix A display the results for $n = 100$ and $n = 250$. Note that without using the recurrences (1) and (2), time complexity to find these results is $\mathcal{O}(n2^{n-1})$, impractical for today's computers for $n = 100$ or $n = 250$.

Expected values EP_n of number of patterns of a sequence of length n, for $n = 100$ and $n = 250$ are $EP_{100} = 29.04319$ and $P_{250} = 57.93485$.

Table 4 in Appendix C displays the EP_n values for some $n \leq 1000$.

6 An Application: A Randomness Test for Binary Sequences

We design a randomness test for binary sequences which employs the algorithms as follows.

Given a sequence of length n bits. First divide the sequence into $M = \lfloor \frac{n}{k} \rfloor$ non-overlapping blocks of length k bits, omitting if necessary last few bits. Apply Lempel-Ziv partitioning procedure to each of these M blocks to get the number of Lempel-Ziv partitions π_i for $1 \leq i \leq M$. From now on we choose $k = 1024$. Set:

$$
\begin{aligned}
r_1 &= |\{i \ : \ \pi_i \leq 174,\ 1 \leq i \leq M\}|,\\
r_2 &= |\{i \ : \ \pi_i = 175,\ 1 \leq i \leq M\}|,\\
r_3 &= |\{i \ : \ \pi_i = 176,\ 1 \leq i \leq M\}|,\\
r_4 &= |\{i \ : \ \pi_i = 177,\ 1 \leq i \leq M\}|,\\
r_5 &= |\{i \ : \ \pi_i \geq 178,\ 1 \leq i \leq M\}|.
\end{aligned}
$$

We obviously have $\sum_{i=1}^{5} r_i = M$. The numbers 174 through 178 are chosen to align $EP_{1024} = 176.09949$ to the center.

Define the random variable X to be the number of partitions of a random sequence of fixed length k bits. Employing the algorithm described in Section 4, we obtain the following probabilities for $k = 1024$.

$$\begin{aligned} p_1 &= Pr(X \leq 174) = 0.05262, \\ p_2 &= Pr(X = 175) = 0.19987, \\ p_3 &= Pr(X = 176) = 0.39720, \\ p_4 &= Pr(X = 177) = 0.29107, \\ p_5 &= Pr(X \geq 178) = 0.05924. \end{aligned}$$

Then apply the χ^2-statistic to the observed data:

$$X(obs) = \sum_{i=1}^{5} \frac{(r_i - Mp_i)^2}{Mp_i}$$

to get the χ^2 random variable $X(obs)$ with degree of freedom 4. Then, the P-value of the test is:

$$\frac{\int_{X(obs)}^{\infty} e^{-u/2} u \; du}{\Gamma(2)\; 2^2} = \frac{1}{2}(X(obs) + 2)e^{-X(obs)/2}\,.$$

A condition that can be safely used with χ^2-approximation is:

$$M \cdot \mathsf{min}\{p_i : 1 \leq i \leq 5\} = M \cdot 0.05262 \geq 5\,.$$

Hence, if k is chosen to be 1024, then n should satisfy $n \geq 100000$ approximately. Note that the test can be applied for any k with respective p_i 's and 'bins' are aligned around EP_k and of course provided that computation of $a(k, l)$ is feasible.

If the P-value of the observed data is less than some threshold (e.g., 0.01), one can conclude that the given sequence is not random. The test applied to the outputs of stream ciphers contesting in ECRYPT's eSTREAM can be found in Appendix B.

7 Conclusion and Future Work

We give two recurrences for the number of sequences of length n with Lempel-Ziv complexity r. We also give the the algorithms and the output of the computer programs that we run to calculate $a(n, r)$ for relatively large values.

We also offer a randomness test that can be applied to the output of ciphers.

The recurrence (2) is quite hard to simplify, but can be used to improve the limiting behaviour of the expected value of $a(n, r)$.

Acknowledgments

The authors would like to thank to anonymous refeeres for their comments, which improved the presentation of the paper. The authors also would like to thank Meltem Sönmez Turan and Çağdaş Çalık for making the outputs of eSTREAM contestant stream ciphers available.

References

1. Lempel, A., Ziv, J.: On the complexity of finite sequences. IEEE Transactions on Information Theory **IT-22** (1976) 75–81
2. Ziv, J., Lempel, A.: A universal algorithm for sequential data compression. IEEE Transactions on Information Theory **IT-23** (1977) 337–343
3. Soto, J.: Statistical testing of random number generators. In: Proceedings of the 22nd National Information Systems Security Conference, Crystal City, Virginia (1999)
4. Jacquet, P., Szpankowski, W.: Asymptotic behavior of the Lempel-Ziv parsing scheme and digital search trees. Theoretical Computer Science **144** (1995)
5. Mund, S.: Ziv-Lempel complexity for periodic sequences and its cryptographic application. In: Advances in cryptology – EUROCRYPT 91 (Brighton, 1991). Volume 547 of Lecture Notes in Comput. Sci. Springer, Berlin (1991) 114–126
6. Kirschenhofer, P., Prodinger, H., and Szpankowski, W.: Digital Search Trees Again Revisited: The Internal Path Length Perspective, SIAM Journal on Computing **23** (1994) 598–616

A Tables for $n = 100$ and $n = 250$

Table 1. $a(100, r)$ and their probabilities

r	$\mathbf{a(100, r)}$	$\mathbf{a(100, r)/2^{100}}$
14	122880	0.000000000000000
15	96129024	0.000000000000000
16	1754408140	0.000000000000000
17	169010698649	0.000000000000000
18	12282745099264	0.000000000000000
19	726896570696704	0.000000000000006
20	35864704163873996	0.000000000000283
21	1555171539525474304	0.000000000012268
22	629504083451115732992	0.000000000496591
23	23657061862581861351424	0.000000018662131
24	796717339700675605430272	0.000000628499162
25	25016712354109852183691264	0.000019734706353
26	701956405285233154502688768	0.000553745965299
27	14929637765344244033503887360	0.011777407562191
28	190072463603886098540862111744	0.149940735696149
29	785071700104053917078962307072	0.619312372007483
30	276807820976750678936983175168	0.218362868227975
31	41183640732091617843871744	0.000032488164108

Table 2. $a(250, r)$

r	**a(250, r)**
22	12582912
23	172462440448
24	207405092700160
25	100022234734919680
26	29027442465801502720
27	5970493862438356647936
28	947059437548499752058880
29	124084577391675511972954112
30	14104448150286646440414281728
31	1429659188269782925153552039936
32	131701470381268947695969402486784
33	11234825836624304676748166609502208
34	902346385748231250614173057894580224
35	68848617082812392433571189369104498688
36	5026197932887293151555523266808542920704
37	353853624800555379505246051484079264628736
38	24168383146155367527845519053853996700663808
39	1608511050085914176405326626207802044763340800
40	104662286330519094422345952269389211024618422272
41	667132695551176212061077931878250432089895102054 4
42	417414764650712462333990517379047167232803455631360
43	25695057332640828405359742259152343883668370141216768
44	1557264023287411624909081480426191697539573325326450688
45	92701547946190343870914352507209237010079274453119795200
46	5404237040271065934800659259750349232918183084398749941760
47	308137037472343306269203510492671163205021043338817023508480
48	17174851395953502738636183446022389293687597007698530091925504
49	9316588073045937726599706610686713191610793793114050983194787 84
50	484817394691686047793981960433627218697379264696567633022822645 76
51	2362999038927091779739893669395333742467402505543139386088962916352
52	1047133818505153148275854660632845980372593186499230049992423568834 56
53	4060841943120707511367011625729606171901794317261167326529027544449024
54	13017346107694736140125677771307099292259458732892898970619719435105 0752
55	3144432233774945014197088996141663864291615518236615683726382172345466880
56	4927006583765989332733885796141541660061885678773427109356001896974778368 0
57	3895848169638225294328109752580747676381229057685776424806198651837533388 80
58	9901525752113743883943868384152065811671713805381086663572492167310999552 00
59	3737097419734912211580818139477038446686170841647726401136662202548224000 00
60	3255419797444187375980631332217791151748569890952982076721561862144000000
61	8866665445157526078469236276839101764442755548908722585600000000000

B Test Results

Table 3. Results of LZ randomness test applied to eSTREAM contestants with parameters $M = 800, k = 1024$ and threshold < 0.01

eSTREAM stream cipher	**P − value**
ABC-v2	0.215686
ACHTERBAHN	0.856026
CryptMT	0.281958
DECIM	0.435354
DICING	0.391681
Dragon	0.784314
Edon80	0.958401
F-FCSR-8	0.559503
FUBUKI	0.805604
Frogbit	0.247524
Grain	0.092822
HC-256	0.189772
Hermes8	0.548511
LEX	0.192730
MAG	0.172511
MICKEY-128	0.951844
MICKEY	0.958706
Mir-1	0.624140
POMARANCH	0.864929
Phelix	0.422482
Polar-Bear	0.032209
ProVEST-4	0.902847
Py	0.518629
Rabbit	0.654306
SFINKS	0.327318
Salsa20	0.325591
TRIVIUM	0.624686
TSC-3	0.943600
WG	0.836510
Yamb	0.514665
ZK-Crypt	0.590525

C Table of Expected Values

Table 4. Expected values EP_n for some $n \leq 1000$

n	EP_n
968	168.285154708125871909
969	168.425325208575350399
970	168.565472359678417715
971	168.705595531148748041
972	168.845694563342602216
973	168.985769897582357720
974	169.125822268928967939
975	169.265852191319223625
976	169.405859611321451745
977	169.545843978226506736
978	169.685804672692994605
979	169.825741477831057654
980	169.965654751385158620
981	170.105545179628954902
982	170.245413297909854937
983	170.385259128160740312
984	170.525082193178254703
985	170.664881890971218212
986	170.804657952890038172
987	170.944410653561969008
988	171.084140625407537432
989	171.223848418188401220
990	171.363534125420147438
991	171.503197345751651805
992	171.642837501466368323
993	171.782454278665110691
994	171.922047870128312202
995	172.061618849379214303
996	172.201167772807546273
997	172.340694800677496380
998	172.480199609262570074
999	172.619681652361296729
1000	172.759140578329111086