automata.tex

Entropy and Expected Acceptance Counts for Finite Automata

Nicholas Pippenger*
(nicholas@cs.ubc.ca)

Department of Computer Science
The University of British Columbia
Vancouver, British Columbia V6T 1Z4
CANADA

Abstract: If a sequence of independent unbiased random bits is fed into a finite automaton, it is straightforward to calculate the expected number of acceptances among the first n prefixes of the sequence. This paper deals with the situation in which the random bits are neither independent nor unbiased, but are nearly so. We show that, under suitable assumptions concerning the automaton, if the the difference between the entropy of the first n bits and n converges to a constant exponentially fast, then the change in the expected number of acceptances also converges to a constant exponentially fast. We illustrate this result with a variety of examples in which numbers following the reciprocal distribution, which governs the significands of floating-point numbers, are recoded in the execution of various multiplication algorithms.

* The work reported here was supported by an NSERC Research Grant and a Canada Research Chair.

1. Introduction and Examples

Consider a finite automaton M with binary input alphabet $\{0,1\}$. Let $M = (Q, \Sigma, \iota, \Delta, F)$, where Q is the set of states, $\Sigma = \{0,1\}$ is the input alphabet, $\iota \in Q$ is the initial state, $\Delta : Q \times \Sigma \to Q$ is the transition function, and $F \subseteq Q$ is the set of final (or accepting) states. We shall extend Δ to a function $\Delta : Q \times \Sigma^* \to Q$ in the usual way. Throughout most of this paper we shall consider only automata that are strongly connnected and aperiodic. An automaton M is *strongly connected* if, for every pair of states $\alpha, \beta \in Q$, there exists an input string $\xi \in \{0,1\}^*$ such that $\Delta(\alpha, \xi) = \beta$. A strongly connected automaton M is *aperiodic* if there there exists an integer $k \geq 1$ such that, for every pair of states $\alpha, \beta \in Q$, there exists an input string $\xi \in \{0,1\}^k$ of length k such that $\Delta(\alpha, \xi) = \beta$.

If a sequence $X_1, X_2, \ldots$ of independent unbiased random bits is fed into M, the sequence of states that M passes through forms a stationary Markov chain. Let $q = |Q|$ be the number of states of M, and let P denote the $q \times q$ transition matrix of the Markov chain, so that $P_{\alpha,\beta}$ is the probability that feeding a random bit into M when it is in state β takes it into state α. If M is strongly connected and aperiodic, this Markov chain is ergodic. This implies that P has a unique stationary distribution ϕ satisfying

$$\phi_\alpha \geq 0,$$

$$\sum_{\beta \in Q} \phi_\beta = 1$$

and

$$\phi_\alpha = \sum_{\beta \in Q} P_{\alpha,\beta} \phi_\beta,$$

for all $\alpha \in Q$. Furthermore, if P_l denotes the l-th power of P, so that $P^l_{\alpha,\beta}$ is the probability that feeding l random bits into M when it is in state β takes it into state α, then every column of P^l converges to ϕ exponentially fast as $l \to \infty$. This means that there exists a constant $c < 1$ such that, for all $\alpha, \beta \in Q$, we have

$$P^l_{\alpha,\beta} = \phi_\alpha + O(c^l). \tag{1.1}$$

As further consequence, for all $\alpha, \beta, \gamma \in Q$, we have

$$P^l_{\alpha,\beta} = P^l_{\alpha,\gamma} + O(c^l). \tag{1.2}$$

Let p_n be the probability that M accepts a uniformly distributed random input string of length n. Then from (1.1) we obtain

$$\begin{aligned} p_n &= \sum_{\beta \in F} P^n_{\beta,\iota} \\ &= \varrho + O(c^n), \end{aligned} \tag{1.3}$$

where

$$\varrho = \sum_{\beta \in F} \phi_\beta .$$

Let

$$e_n = \sum_{1 \le k \le n} p_k$$

be the expected number of accepted non-empty prefixes when n independent unbiased bits are fed into M. Then (1.3) implies

$$e_n = \varrho n + C + O(c^n)$$

for some constant C.

For example, let M_0 be the minimal automaton accepting the language $(0+1)^*1$ comprising all strings that end with a 1. Let M_1 be the minimal automaton accepting the language $1 + (0+1)^*(01+10)$ comprising all string that either consist of a single 1 or consist of two or more symbols, of which the last two are different. For both M_0 and M_1, we have

$$p_n = \frac{1}{2}$$

and

$$e_n = \frac{n}{2}.$$

Our goal in this paper is to study what happens when the distribution of the random input to M is not uniform, but is in some sense close to uniform. Our motivating example is the case in which the input bits $X_1, X_2, \ldots$ are the successive bits in the binary expansion of a real number $X = \sum_{n \ge 1} X_n \, 2^{-n}$ that is distributed on the interval $[1/2, 1)$ according to the reciprocal distribution (also known as the logarithmic distribution):

$$\Pr[x \le X \le y] = \frac{1}{\log 2} \int_x^y \frac{dX}{X} = \log_2 \left(\frac{y}{x} \right) .$$

Hamming [H] has argued that the reciprocal distribution is the appropriate one to use for the significand (also known as the mantissa) of a random floating-point number. In particular, Hamming shows among other things that the product of a large number of independent indentically distributed numbers (with mild assumptions concerning their distribution) has a significand with the reciprocal distribution. (This phenomenon is related to the logarithmic distribution of leading digits observed by Newcomb [N].)

First consider feeding the bits of a reciprocally distributed significand into the finite automaton M_0 accepting the language $(0+1)^*1$, so that e_n is simply the expected number of 1s among the first n bits of the significand. A simple calculation shows that $p_1 = 1$ and, for $n \geq 2$,

$$\begin{aligned} p_n &= \sum_{2^{n-2}+1 \leq k \leq 2^{n-1}} \frac{1}{\log 2} \int_{(2k-1)/2^n}^{(2k)/2^n} \frac{dX}{X} \\ &= \log_2 \prod_{2^{n-2}+1 \leq k \leq 2^{n-1}} \left(\frac{2k}{2k-1} \right), \end{aligned}$$

so that

$$\begin{aligned} e_n &= \log_2 \prod_{1 \leq k \leq 2^{n-1}} \left(\frac{2k}{2k-1} \right) \\ &= \frac{1}{2} \log_2 \prod_{1 \leq k \leq 2^{n-1}} \left(\frac{2k}{2k-1} \right)^2 \\ &= \frac{n}{2} + \frac{1}{2} \log_2 \prod_{1 \leq k \leq 2^{n-1}} \left(\frac{2k}{2k-1} \cdot \frac{2k}{2k+1} \right), \end{aligned}$$

where we have used $\frac{1}{2}\log_2(2^n+1) = \frac{n}{2} + O(1/2^n)$. Using Wallis's [W1] formula

$$\frac{\pi}{2} = \prod_{1 \leq k < \infty} \left(\frac{2k}{2k-1} \cdot \frac{2k}{2k+1} \right),$$

together with the estimate

$$\frac{k}{k-1} \cdot \frac{k}{k+1} = 1 + O\left(\frac{1}{k^2} \right),$$

which yields

$$\prod_{l \leq k < \infty} \left(\frac{2k}{2k-1} \cdot \frac{2k}{2k+1} \right) = 1 + \left(\frac{1}{l} \right),$$

we obtain

$$e_n = \frac{n}{2} + \log_2 \left(\frac{\pi}{2} \right) + O\left(\frac{1}{2^n} \right).$$

This result differs from the result $n/2$ for uniform input by a constant, $\frac{1}{2}\log_2\frac{\pi}{2}$, and an exponentially small error term. Taking differences gives the estimate

$$\begin{aligned} p_n &= e_n - e_{n-1} \\ &= \frac{1}{2} + O\left(\frac{1}{2^n}\right) \end{aligned} \tag{1.4}$$

for the probability that $X_n = 1$. (Note that the first bit of a significand is always 1, which raises the expected number of acceptances by 1/2. Each remaining bit, however, is more likely to be 0 than 1, so the limit of the number of extra acceptances, $\frac{1}{2}\log_2\frac{\pi}{2} = 0.3257\ldots$, is less than 1/2.)

Next consider feeding the bits of a reciprocally distributed significand into the finite automaton M_1 accepting the language $1 + (0+1)^*(01+10)$. A simple calculation shows that $p_1 = \log_2\frac{2}{1}$, $p_2 = \log_2\frac{3}{2}$ and, for $n \geq 3$,

$$\begin{aligned} p_n &= \sum_{2^{n-3}+1 \leq k \leq 2^{n-2}} \frac{1}{\log 2} \int_{(4k-3)/2^n}^{(4k-1)/2^n} \frac{dX}{X} \\ &= \log_2 \prod_{2^{n-3}+1 \leq k \leq 2^{n-2}} \left(\frac{4k-1}{4k-3}\right), \end{aligned}$$

so that

$$\begin{aligned} e_n &= \log_2 \prod_{1 \leq k \leq 2^{n-2}} \left(\frac{4k-1}{4k-3}\right) \\ &= \frac{1}{2}\log_2 \prod_{1 \leq k \leq 2^{n-2}} \left(\frac{4k-1}{4k-3}\right)^2 \\ &= \frac{n}{2} + \frac{1}{2}\log_2 \prod_{1 \leq k \leq 2^{n-2}} \left(\frac{4k-1}{4k-3} \cdot \frac{4k-1}{4k+1}\right) + O\left(\frac{1}{2^n}\right) \\ &= \frac{n}{2} + \frac{1}{2}\log_2 \prod_{1 \leq k < \infty} \left(\frac{4k-1}{4k-3} \cdot \frac{4k-1}{4k+1}\right) + O\left(\frac{1}{2^n}\right). \end{aligned} \tag{1.5}$$

We shall use the formula

$$\prod_{k \geq 1} \frac{(k+\alpha_1)\cdots(k+\alpha_t)}{(k+\beta_1)\cdots(k+\beta_t)} = \frac{\Gamma(1+\beta_1)\cdots\Gamma(1+\beta_t)}{\Gamma(1+\alpha_1)\cdots\Gamma(1+\alpha_t)}, \tag{1.6}$$

where $\Gamma(\cdots)$ denotes Euler's Gamma-function (see Whittaker and Watson [W2], Section $12 \cdot 13$), and $\alpha_1 + \cdots + \alpha_t = \beta_1 + \cdots + \beta_t$. (Note that Wallis's formula is the special case of (1.6) in which $t = 2$, $\alpha_1 = \alpha_2 = 0$, $\beta_1 = -1/2$ and $\beta_2 = 1/2$, since

$$\prod_{k\geq 1} \frac{2k}{2k-1} \cdot \frac{2k}{2k+1} = \prod_{k\geq 1} \frac{k}{k-\frac{1}{2}} \cdot \frac{k}{k+\frac{1}{2}},$$

and $\Gamma(1) = 1$, $\Gamma(1/2) = \pi^{1/2}$ and $\Gamma(3/2) = \pi^{1/2}/2$.) Applying (1.6) to (1.5), we have

$$\begin{aligned} e_n &= \frac{n}{2} + \frac{1}{2}\log_2 \prod_{1\leq k<\infty} \left(\frac{k-\frac{1}{4}}{k-\frac{3}{4}} \cdot \frac{k-\frac{1}{4}}{k+\frac{1}{4}} \right) + O\left(\frac{1}{2^n}\right) \\ &= \frac{n}{2} + \frac{1}{2}\log_2 \left(\frac{\Gamma(\frac{1}{4})\,\Gamma(\frac{5}{4})}{\Gamma(\frac{3}{4})^2} \right) + O\left(\frac{1}{2^n}\right) \\ &= \frac{n}{2} + \frac{1}{2}\log_2 \left(\frac{\Gamma(\frac{1}{4})^4}{8\pi^2} \right) + O\left(\frac{1}{2^n}\right), \end{aligned}$$

where we have used $\Gamma(5/4) = \Gamma(1/4)/4$ and $\Gamma(3/4) = \pi/\sin(\pi/4)\Gamma(1/4) = 2^{1/2}\pi/\Gamma(1/4)$ (see Whittaker and Watson [W2], Sections $12 \cdot 12$ and $12 \cdot 14$). This result again differs from the result $n/2$ for uniform input by a constant, in this case $\frac{1}{2}\log_2\left(\Gamma(\frac{1}{4})^4/8\pi^2\right)$, and an exponentially small error term. (Note that the first bit of a significand is always 1, which raises the expected number of acceptances by 1/2. The second bit is more likely to be 0 than 1, and thus more likely to be different from than the same as the first bit, which further raises the expected number of acceptances by $\log_2(3/2) - (1/2) = 0.08496\ldots$. Each remaining bit is also more likely to be 0 than 1, and thus more likely to be the same as than different from the previous bit, but by rapidly diminishing amounts, so the limit of the number of extra acceptances, $\frac{1}{2}\log_2\left(\Gamma(\frac{1}{4})^4/8\pi^2\right) = 0.5649\ldots$, is greater than 1/2.)

The acceptances of the automaton M_0 correspond to the 1s in its input, and reflect the additions performed by the standard shift-and-add algorithm for multiplication. The acceptances of M_1 reflect the additions and subtractions performed by a multiplication algorithm that recodes the multiplier in a manner suggested by Booth [B1]. (This recoding differs from what has become known as "Booth recoding", which we shall deal with later.) This recoding replaces a substring of the form $0^k 1^l$ in the input by the string $0^{k-1}10^{l-1}\overline{1}$, where $\overline{1}$ calls for a subtraction rather than an addition. When applying this algorithm to a finite input string, it is necessary to append a 0 to the end of the string; this "flushes out" the automaton, returning it to its initial state, and producing one further acceptance if the last bit of the input string is 1. For uniformly distributed input, the probability of this further accceptance is exactly 1/2, and we have seen in (1.4) that for reciprocally distributed input it is $1/2 + O(1/2^n)$. Thus this recoding actually increases the number of operations required for a random multiplier. Its advantage, however, lies in eliminating almost all of the operations corresponding to leading 1s when the multiplier is a small negative number represented in 2s-complement form. (We should note that when we speak of the number of additions and subtractions, we are assuming that the partial product is initialized to zero, and that the adder is used for each non-zero digit in the recoded multiplier. An alternative is to use the first non-zero digit to initialize the partial product, and not to use the adder until a second non-zero digit is encountered. The effect of this improvement is to reduce the number of uses of the adder by exactly one, except in those cases in which all n bits of the multiplier are 0s; this event has probability $1/2^n$ for the uniform distribution, and probability 0 for the reciprocal distribution.)

At this point we could give many other examples of feeding sequences of bits with various distributions into various finite automata. For most of the examples that arise in analysis of arithmetic algorithms with natural input distributions, the results are similar to those presented above: the expected number of acceptances for the given input distribution differs from that for the uniform input distribution by a constant and an exponentially small error term. Our goal in the next section will be to determine reasonably general conditions under which this type of result holds.

2. The Main Theorems

Our main result will relate the asymptotic behaviour of the acceptance counts to that of the entropy of the input sequence. Following Shannon [S], we define the *entropy* of a random variable Ξ to be

$$H(\Xi) = -\sum_{\xi} \Pr[\Xi = \xi] \log_2 \Pr[\Xi = \xi].$$

Let $\Xi_n = X_1 \cdots X_n$ denote the first n bits of the input sequence $X_1, X_2, \ldots$. Define

$$h_n = H(\Xi_n)$$

to be the entropy of these n bits.

Theorem 2.1: Suppose the entropy of $X_1, X_2, \ldots$ satisfies

$$h_n = n - A + O(a^n)$$

for some constant $A \geq 0$ and some $1/2 < a < 1$. If the input $X_1, X_2, \ldots$ is fed into a strongly-connected and aperiodic finite automaton, then

$$p_n = \varrho + O(b^n)$$

for some constant $b < 1$, and thus

$$e_n = \varrho n + B + O(b^n)$$

for some constant B.

The proof will require a lemma.

Lemma 2.2: Let U be a random variable taking values in $\{0,1\}^l$. Let $1/2 < a < 1$ and let $a^l < \varepsilon < 1$. Suppose that there is a set $T \subseteq \{0,1\}^l$ such that either

$$\Pr[U \in T] \geq \varepsilon$$

or

$$\frac{|T|}{2^l} \geq \varepsilon,$$

and that either

$$\Pr[U = v] \geq \frac{1+\varepsilon}{2^l}$$

for every $v \in T$, or

$$\Pr[U = v] \leq \frac{1}{(1+\varepsilon)\, 2^l}$$

for every $v \in T$. Then

$$H(U) \leq l - \Omega(\varepsilon^3)$$

for sufficiently large l.

Proof: We shall deal with the case in which

$$\Pr[U \in T] \geq \varepsilon$$

and

$$\Pr[U = v] \geq \frac{1+\varepsilon}{2^l};$$

the remaining three cases are similar.

We note that

$$\frac{2^l}{1+\varepsilon/2} - \frac{2^l}{1+\varepsilon} = \Omega(\varepsilon\, 2^l) = \Omega\big((2a)^l\big).$$

Since $(2a)^l \to \infty$ as $l \to \infty$, the interval $[2^l/(1+\varepsilon/2), 2^l/(1+\varepsilon)]$ contains an integer for all sufficiently large l. Thus we can find ε' in the range $\varepsilon/2 \le \varepsilon' \le \varepsilon$ such that $2^l/(1+\varepsilon')$ (and therefore also $\varepsilon'\, 2^l/(1+\varepsilon')$) is an integer,

$$\Pr[U \in T] \ge \varepsilon'$$

and

$$\Pr[U = v] \ge \frac{1+\varepsilon'}{2^l}.$$

It will now suffice to show that

$$H(U) \le l - \Omega\big((\varepsilon')^3\big).$$

Under these conditions, it is easy to see that the distribution that maximizes the entropy equally divides probability ε' among $\varepsilon'\, 2^l/(1+\varepsilon')$ values in $\{0,1\}^l$ (giving each of them probability $(1+\varepsilon')/2^l$), and equally divides the remaining probability $1-\varepsilon'$ among the remaining $2^l/(1+\varepsilon')$ values in $\{0,1\}^l$ (giving each of them probability $\big(1-(\varepsilon')^2\big)/2^l$). The entropy of this distribution is

$$\begin{aligned} H(U) &= -\varepsilon' \log_2 \frac{1+\varepsilon'}{2^l} - (1-\varepsilon') \log_2 \frac{1-(\varepsilon')^2}{2^l} \\ &= l - \log_2(1+\varepsilon') - (1-\varepsilon') \log_2(1-\varepsilon'). \end{aligned}$$

Using the expansion $\log x = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 + O(x^4)$, we obtain $H(U) = l - \frac{1}{2}(\varepsilon')^3 + O\big((\varepsilon')^4\big)$. This completes the proof of the lemma. □

Proof of Theorem 2.1: Let $\Xi_n = YZ$, where $|Y| = m = \lfloor n/2 \rfloor$ and $|Z| = l = \lceil n/2 \rceil$. A simple calculation shows that

$$H(\Xi_n) = H(Y) + H(Z \mid Y),$$

where the *conditional entropy* of Z with respect to Y is given by

$$H(Z \mid Y) = -\sum_{\eta} \Pr[Y = \eta] \sum_{\zeta} \Pr[Z = \zeta \mid Y = \eta] \log_2 \Pr[Z = \zeta \mid Y = \eta].$$

Thus

$$\begin{aligned} H(Z \mid Y) &= h_n - h_m \\ &= l + O(a^{n/2}), \end{aligned}$$

by the hypothesis of the theorem. We can write this as

$$\sum_{\eta} \Pr[Y = \eta]\big(l - H(Z \mid Y = \eta)\big) = O(a^{n/2}), \tag{2.1}$$

where the *entropy* of Z *conditioned on the event* $Y = \eta$ is given by

$$H(Z \mid Y = \eta) = -\sum_{\zeta} \Pr[Z = \zeta \mid Y = \eta] \log_2 \Pr[Z = \zeta \mid Y = \eta].$$

Since Z takes on at most 2^l different values, $H(Z \mid Y = \eta) \leq l$. Thus the quantity in parentheses in (2.1) is non-negative, which implies that there exists a set V such that

$$\Pr[Y \notin V] = O(a^{n/4}) \tag{2.2}$$

and, for every $\eta \in V$, we have

$$H(Z \mid Y = \eta) = l + O(a^{n/4}).$$

Using Lemma 2.2, we have that there exists a set W_η such that

$$\Pr[Z \notin W_\eta \mid Y = \eta] = O(a^{n/12}), \tag{2.3}$$

$$\frac{\big|\{0,1\}^l \setminus W_\eta\big|}{2^l} = O(a^{n/12}), \tag{2.4}$$

and, for every $\zeta \in W_\eta$, we have

$$\Pr[Z = \zeta \mid Y = \eta] = \frac{1}{2^l}\big(1 + O(a^{n/12})\big). \tag{2.5}$$

We have

$$p_n = \sum_{\beta \in F} \sum_{\substack{|\xi| = n \\ \Delta(\iota, \xi) = \beta}} \Pr[\Xi_n = \xi].$$

From this we obtain

$$p_n = \sum_{\alpha \in Q} \sum_{\substack{|\eta| = m \\ \Delta(\iota, \eta) = \alpha}} \Pr[Y = \eta] \sum_{\beta \in F} \sum_{\substack{|\zeta| = l \\ \Delta(\alpha, \zeta) = \beta}} \Pr[Z = \zeta \mid Y = \eta].$$

Taking account of (2.2), (2.3) and (2.5), we have

$$p_n = \sum_{\alpha\in Q} \sum_{\substack{\eta \in V \\ \Delta(\iota,\eta)=\alpha}} \Pr[Y=\eta] \sum_{\beta\in F} \sum_{\substack{\zeta \in W_\eta \\ \Delta(\alpha,\zeta)=\beta}} \frac{1}{2^l} \;+\; O(a^{n/12}).$$

Taking account of (2.4), we have

$$p_n = \sum_{\alpha\in Q} \sum_{\substack{\eta \in V \\ \Delta(\iota,\eta)=\alpha}} \Pr[Y=\eta] \sum_{\beta\in F} \sum_{\substack{|\zeta| = l \\ \Delta(\alpha,\zeta)=\beta}} \frac{1}{2^l} \;+\; O(a^{n/12}).$$

The innermost sum is the probability $P^l_{\beta,\alpha}$ that a uniformly distributed sequence of l bits takes M from state α to state β. Using (1.2), this implies

$$p_n = \sum_{\alpha\in Q} \sum_{\substack{\eta \in V \\ \Delta(\iota,\eta)=\alpha}} \Pr[Y=\eta] \sum_{\beta\in F} P^l_{\beta,\iota} \;+\; O(c^{n/2}) \;+\; O(a^{n/12}).$$

The innermost sum is now the probability that M accepts a uniformly distributed sequence of l bits. Using (1.3), we obtain

$$p_n = \varrho \sum_{\alpha\in Q} \sum_{\substack{\eta \in V \\ \Delta(\iota,\eta)=\alpha}} \Pr[Y=\eta] \;+\; O(c^{n/2}) \;+\; O(a^{n/12}).$$

Again taking account of (2.2), we have

$$p_n = \varrho \sum_{\alpha\in Q} \sum_{\substack{|\eta| = m \\ \Delta(\iota,\eta)=\alpha}} \Pr[Y=\eta] \;+\; O(c^{n/2}) \;+\; O(a^{n/12}).$$

The double sum is 1, since every sequence η must take the initial state ι to some state α. This completes the proof of the theorem. $\Box$

Let us now consider what probability distributions meet the condition of Theorem 2.1.

Theorem 2.3: Suppose that $X_1, X_2, \ldots$ are the successive bits in the binary expansion of a real number $X = \sum_{n\geq 1} X_n \, 2^{-n}$ that is distributed on the interval $[0,1)$ according to the density function f, which satisfies the following conditions: there exist $j \geq 0$ and breakpoints $0 = a_0 < a_1 < \cdots < a_j < a_{j+1} = 1$ such that, for each $0 \le i \le j$, f satisfies the Lipschitz condition $|f(x) - f(y)| = O(|x-y|)$ for $a_i < x < y < a_{i+1}$. Then

$$h_n = n - A + O\left(\frac{n}{2^n}\right),$$

where

$$A = \int_0^1 f(X) \log_2 f(X)\, dX.$$

Proof: We shall deal with the case with $j = 0$ breakpoints; the general case follows merely by elaborating the notation. We have

$$h_n = - \sum_{0 \le k < 2^n} \left(\int_{k/2^n}^{(k+1)/2^n} f(X)\, dX \right) \log_2 \left(\int_{k/2^n}^{(k+1)/2^n} f(X)\, dX \right). \tag{2.6}$$

From the Lipschitz condition we have

$$\int_{k/2^n}^{(k+1)/2^n} f(X)\, dX = \frac{1}{2^n} f\left(\frac{k}{2^n}\right) \left(1 + O\left(\frac{1}{2^n}\right)\right).$$

Substituting this into (2.6), and using the fact that the Lipschitz condition ensures that f, and therefore also

$$g(X) = f(X) \log_2 f(X)$$

is bounded, yields

$$h_n = n - \frac{1}{2^n} \sum_{0 \le k < 2^n} f\left(\frac{k}{2^n}\right) \log_2 f\left(\frac{k}{2^n}\right) + O\left(\frac{n}{2^n}\right).$$

Estimating the sum by an integral, using the fact that the Lipschitz condition ensures that f, and therefore also g, has bounded total variation, we obtain

$$h_n = n - \int_0^1 f(X) \log_2 f(X)\, dX + O\left(\frac{n}{2^n}\right).$$

This completes the proof of the theorem. $\Box$

As an example, consider the reciprocal distribution, which corresponds to

$$f(X) = \begin{cases} 0, & \text{for } 0 \le X < 1/2; \\ \dfrac{1}{X \log 2}, & \text{for } 1/2 \le X < 1. \end{cases}$$

For this distribution we obtain $A = \frac{1}{2} + \log_2 \log 2e = 1.0287\ldots$ bits. Since the first bit is always 1, it alone accounts for 1 bit of entropy loss; and this first bit can be omitted from the representation of the significand as a "hidden" bit. The remaining $n-1$ bits thus have just $-\frac{1}{2} + \log_2 \log_2 e = 0.0287\ldots$ bits less entropy than $n-1$ uniformly distributed bits.

3. Further Examples

We have seen in Theorem 2.1 that if the input to a strongly connected and aperiodic finite automaton has entropy that differs from that of a unifomly distributed input by a constant and an exponentially small error term, then the expected acceptance count also differs from that for a uniformly distributed input by a constant and an exponentially small error term. Our motivating example has been the reciprocal distribution, which governs the significands of floating-point numbers. As Theorem 2.3 shows, however, many other natually arising distributions satisfy this condition. We may take, for further examples, the stationary distributions governing the partial remainders when a dividend (with mild assumptions concerning its distribution) is divided by a positive integer using the original S-R-T division algorithm (see Freiman [F1]). These distributions are piecewise constant, with breakpoints at dyadic rational numbers; as a result, they have, for all sufficiently large n,

$$h_n = n - A,$$

with no error term! For division by 3, 5 and 7, we have

$$f_3(X) = \begin{cases} 4/3, & \text{if } 0 \le X < 1/2, \\ 2/3, & \text{if } 1/2 \le X < 1, \end{cases}$$

$$A_3 = \frac{5}{3} - \log_2 3 = 0.0817\ldots,$$

$$f_5(X) = \begin{cases} 8/5, & \text{if } 0 \le X < 1/4, \\ 16/15, & \text{if } 1/4 \le X < 1/2, \\ 4/5, & \text{if } 1/2 \le X < 3/4, \\ 8/15, & \text{if } 3/4 \le X < 1, \end{cases}$$

$$A_5 = \frac{46}{15} - \frac{2}{3}\log_2 3 - \log_2 5 = 0.1107\ldots$$

and

$$f_7(X) = \begin{cases} 8/7, & \text{if } 0 \le X < 3/4, \\ 4/7, & \text{if } 3/4 \le X < 1, \end{cases}$$

$$A_7 = \frac{20}{7} - \log_2 7 = 0.04978\ldots,$$

respectively.

Theorem 2.1 has a corollary that covers the case in which the automaton M is strongly connected but periodic. In this case we can find a positive integer d such that the automaton with input alphabet $\{0,1\}^d$ that accepts d input bits and produces d output bits at a time has d connected components, each of which is strongly connected and aperiodic. The smallest such d is called the *period* of M. Analysis similar to that in the proof of Theorem 2.1 then applies to the original automaton for each equivalence class of n modulo d. For each such class $0 \le c \le d-1$, there will be an acceptance rate ϱ_c, and the overall accceptance rate ϱ will be the average of these:

$$\varrho = \frac{\varrho_0 + \cdots \varrho_{d-1}}{d}.$$

The conclusion is then that there are constants $B_0, \ldots, B_{d-1}$ such that, for any $0 \le c \le d-1$,

$$p_n = \varrho_c + O(b^n)$$

and

$$e_n = \varrho n + B_c + O(b^n),$$

as $n \to \infty$ through integers congruent to c modulo d.

Examples of strongly connected but periodic automata are those whose acceptances correspond to the additions and subtractions performed by multiplication algorithms that recode pairs or triplets of bits at each step (see MacSorley [M]). (These algorithms have become know as "Booth recoding" algorithms; they were first published by MacSorley, though Booth [B2] attributes the triplet version to K. D. Tocher!)

Recoding of pairs (see MacSorley [M], "Uniform Shifts of Two") operates as follows. The bits of the multiplier are examined in pairs, starting from the binary point and working to the right. Each pair is recoded so as to call for a single addition or subtraction, *unless* both bits of the pair are the same as the right bit of the preceding pair, in which case no operation is required. (When examining the first pair, this preceding bit is taken to be 0.) The operations called for by the recoding thus correspond to the acceptances of an automaton M_2 recognizing the language

$$L_2 = \big((0+1)^2 - 00\big) + \big((0+1)^2\big)^*(0+1)\big((0+1)^3 - (000+111)\big).$$

This automaton has period 2. This language contains only strings of even length, and it is easy to see that for uniform input we have

$$p_n = \begin{cases} 0, & \text{if } n \text{ odd;} \\ 3/4, & \text{if } n \text{ even.} \end{cases}$$

Thus we have

$$e_n = \frac{3n}{8} + C_c$$

for n congruent to c modulo 2, where $C_0 = 0$ and $C_1 = -3/8$. For simplicity, we shall confine our attention to the case of n even, say $n = 2m$. When applying this algorithm to a finite input string, it is necessary to append two 0s to the end of the string. This will produce one further acceptance if the last bit of the input string is 1. For uniformly distributed input, the probability of this further accceptance is exactly 1/2, and we have seen in (1.4) that for reciprocally distributed input it is $1/2 + O(1/2^n)$.

Let us now consider this algorithm with reciprocally distributed input. It will be convenient to work with the complement

$$L_2' = \left((0+1)^2\right)^* - L = 00 + \left((0+1)^2\right)^*(0+1)(000+111)$$

of L_2 with respect to the set of strings of even length. The strings of L_2' correspond to cases in which the recoding of a pair does *not* call for an addition or subtraction. Let q_m denote the probability that an automaton M_2' recognizing L_2' accepts a recriprically distributed input string of length $n = 2m$. Then we have

$$e_{2m} = m - f_m, \tag{3.1}$$

where

$$f_m = \sum_{1 \le l \le m} q_l.$$

We have $q_1 = 0$. (Since the first bit is always 1, the first two bits cannot both be 0s.) For $l \ge 2$ we have

$$\begin{aligned} q_l &= \sum_{0 \le j \le 4^{l-2}-1} \frac{1}{\log 2} \left(\int_{1/2+(8j)/4^l}^{1/2+(8j+1)/4^l} \frac{dX}{X} + \int_{1/2+(8j+7)/4^l}^{1/2+(8j+8)/4^l} \frac{dX}{X} \right) \\ &= \sum_{4^{l-2} \le k \le 2\cdot 4^{l-1}-1} \log_2 \left(\frac{8k+1}{8k} \cdot \frac{8k+8}{8k+7} \right). \end{aligned}$$

Summing for $1 \le l \le m$, we obtain

$$\begin{aligned} f_m &= \sum_{\substack{1 \le k \le 4^{m-1}-1 \\ \lfloor \log_2 k \rfloor \text{ even}}} \log_2 \left(\frac{8k+1}{8k} \cdot \frac{8k+8}{8k+7} \right) \\ &= \frac{1}{2} \sum_{1 \le k \le 4^{m-1}-1} \log_2 \left(\frac{8k+1}{8k} \cdot \frac{8k+8}{8k+7} \right) \\ &\quad + \frac{1}{2} \sum_{1 \le k \le 4^{m-1}-1} (-1)^{\lfloor \log_2 k \rfloor} \log_2 \left(\frac{8k+1}{8k} \cdot \frac{8k+8}{8k+7} \right). \end{aligned} \tag{3.2}$$

The first sum can be evaluated using the formula (1.6); the result is

$$\begin{aligned}
&\frac{1}{2} \sum_{1 \le k \le 4^{m-1}-1} \log_2 \left(\frac{8k+1}{8k} \cdot \frac{8k+8}{8k+7} \right) \\
&\qquad = \frac{m}{4} + \frac{1}{2} \log_2 \left(\frac{7\,\Gamma(\frac{7}{8})}{2^{1/2}\,\Gamma(\frac{1}{8})} \right) + O\left(\frac{1}{4^m} \right) \\
&\qquad = \frac{m}{4} + \frac{1}{2} \log_2 \left(\frac{7\,\pi}{2^{1/2}\, \sin\frac{\pi}{8}\,\Gamma(\frac{1}{8})^2} \right) + O\left(\frac{1}{4^m} \right), \qquad (3.3)
\end{aligned}$$

where we have used $\Gamma(15/16) = 7\,\Gamma(7/16)/16$ and $\Gamma(7/8) = \pi/\bigl(\sin(\pi/8)\,\Gamma(1/8)\bigr)$ (see Whittaker and Watson [W2], Sections $12 \cdot 12$ and $12 \cdot 14$). For the second sum we must exercise care, since the sum of $T_k = \log_2\bigl((8k+1)(8k+8)/(8k)(8k+7)\bigr)$ over $1 \le k \le N$ oscillates without converging as $N \to \infty$. With the upper limit of $4^{m-1} - 1$, however, we can associate each positive term T_k (with $\lfloor \log_2 k \rfloor$ even) with the two negative terms T_{2k} and T_{2k+1} (for which $\lfloor \log_2 k \rfloor$ is odd). The result is

$$\begin{aligned}
&\frac{1}{2} \sum_{1 \le k \le 4^{m-1}-1} (-1)^{\lfloor \log_2 k \rfloor} \log_2 \left(\frac{8k+1}{8k} \cdot \frac{8k+8}{8k+7} \right) \\
&\qquad = \frac{1}{2} \sum_{\substack{1 \le k \le 4^{m-1}-1 \\ \lfloor \log_2 k \rfloor \text{ even}}} \log_2 \left(\frac{16k+2}{16k+1} \cdot \frac{16k+7}{16k+9} \cdot \frac{16k+15}{16k+14} \right).
\end{aligned}$$

Since

$$\log_2 \left(\frac{16k+2}{16k+1} \cdot \frac{16k+7}{16k+9} \cdot \frac{16k+15}{16k+14} \right) = O\left(\frac{1}{k^2} \right),$$

this last sum is absolutely convergent, and we obtain

$$\begin{aligned}
&\frac{1}{2} \sum_{1 \le k \le 4^{m-1}-1} (-1)^{\lfloor \log_2 k \rfloor} \log_2 \left(\frac{8k+1}{8k} \cdot \frac{8k+8}{8k+7} \right) \\
&\qquad = \frac{1}{2} \sum_{\substack{1 \le k < \infty \\ \lfloor \log_2 k \rfloor \text{ even}}} \log_2 \left(\frac{16k+2}{16k+1} \cdot \frac{16k+7}{16k+9} \cdot \frac{16k+15}{16k+14} \right) + O\left(\frac{1}{4^m} \right). \qquad (3.4)
\end{aligned}$$

Substituting (3.4) and (3.3) into (3.2), and the result into (3.1), yields

$$e_n = \frac{3n}{8} + B_0 + O\left(\frac{1}{2^n} \right),$$

for n even, where

$$B_0 = -\frac{1}{2}\log_2\left(\frac{7\,\pi}{2^{1/2}\,\sin\frac{\pi}{8}\,\Gamma(\frac{1}{8})^2}\right) - \frac{1}{2}\sum_{\substack{1\le k<\infty \\ \lfloor\log_2 k\rfloor \text{ even}}} \log_2\left(\frac{16k+2}{16k+1}\cdot\frac{16k+7}{16k+9}\cdot\frac{16k+15}{16k+14}\right).$$

Numerical computation gives $B_0 = 0.2359\ldots$. As was mentioned before, it is necessary to add a further $1/2 + O(1/2^n)$ to this result to obtain the expected number of operations when recoding a string of even length n (The first bit of a significand is always 1, so the first two bits cannot be 00, which raises the expected number of operations by 1/4. The bias of the remaining bits decreases the expected number of operations for the remaining pairs, but by rapidly diminishing amounts, so the limit of the expected number of extra operations is slightly less than 1/4.)

Recoding of triplets (see MacSorley [M], "Uniform Shifts of Three") operates as follows. The bits of the multiplier are examined in groups of three, starting from the binary point and working to the right. Each triplet is recoded so as to call for a single addition or subtraction (of either the muliplicand or the triple of the multiplicand, which is assumed to have been prepared in advance), *unless* all three bits of the triplet are the same as the rightmost bit of the preceding triplet, in which case no operation is required. (When examining the first triplet, this preceding bit is taken to be 0.) The operations called for by the recoding thus correspond to the acceptances of an automaton M_3 recognizing the language

$$L_3 = \big((0+1)^3 - 000\big) + \big((0+1)^3\big)^*(0+1)^2\big((0+1)^4 - (0000+1111)\big).$$

This automaton has period 3. This language contains only strings of length divisible by three, and it is easy to see that for uniform input we have

$$p_n = \begin{cases} 0, & \text{if } n \equiv 1, 2 \pmod 3; \\ 7/8, & \text{if } n \equiv 0 \pmod 3). \end{cases}$$

Thus we have

$$e_n = \frac{7n}{24} + C_c$$

for n congruent to c modulo 3, where $C_0 = 0$, $C_1 = -7/24$ and $C_2 = -7/12$. For simplicity, we shall consider only the case in which n is a multiple of 3. When applying this algorithm to a finite input string, it is necessary to append three 0s to the end of the string. This will produce one further acceptance if the last bit of the input string is 1. For uniformly distributed input, the probability of this further accceptance is exactly 1/2, and we have seen in (1.4) that for reciprocally distributed input it is $1/2 + O(1/2^n)$.

The analysis of this algorithm with reciprocally distributed input is analogous to that for recoding pairs, and we shall only present the final result. We have

$$e_n = \frac{7n}{24} + B_0 + O\left(\frac{1}{2^n}\right),$$

for n congruent to 0 modulo 3, where

$$B_0 = -\frac{1}{3}\log_2\left(\frac{15\cdot 31\,\pi}{17\cdot 2^{5/4}\,\sin\frac{\pi}{16}\,\Gamma(\frac{1}{16})^2}\right) - \frac{1}{3}\sum_{\substack{2\le k<\infty\\ \lfloor\log_2 k\rfloor\equiv 1\,(\mathrm{mod} 3)}} \log_2 R_k$$

and

$$R_k = \frac{(64k+4)^2}{(64k+60)^2}\cdot\frac{64k+15}{64k+49}\cdot\frac{64k+30}{64k+34}\cdot\frac{64k+31}{64k+33}\cdot\frac{64k+47}{64k+17}\cdot\frac{64k+62}{64k+2}\cdot\frac{64k+63}{64k+1}.$$

Numerical computation gives $B_0 = 0.1227\ldots$. As was mentioned before, it is necessary to add a further $1/2 + O(1/2^n)$ to this result to obtain the number of operations when recoding a string of length n, a multiple of 3. (Just as the expected number of extra operations for recoding pairs is slightly less than 1/4, the number for recoding triplets is slightly less than 1/8.)

Finally, we should mention that though Theorem 2.1 deals with finite automata processing in successive bits of a real number from left to right (that is, from most significant to least significant), it is also possible to apply it to multiplication algorithms that recode the multiplier from right to left. An example of this is canonical recoding. The first published descriptions of canonical recoding were given by Lehman [L1, L2], Tocher [T] and Reitwiesner [R], who all developed it indpendently during the 1950s. It was also discussed in the review by MacSorley [M] (in the section "Multiplication Using Variable Length Shift").

Canonical recoding is performed by an automaton that processes the input from right to left. It cannot be produced (with any bounded delay) by a finite automaton processing the bits from left to right. It can be produced in this direction, however, by a technique called "on-the-fly" conversion (see Frougny [F]); this technique employs a sequential machine that has a register, capable of holding a string, for each state of the finite automaton. It is also possible to produce in this direction a non-canonical recoding that the same number of non-zero digits as the canonical recoding (see MacSorley [M]). Thus the number of non-zero digits in the canonical recoding (though not the canonical recoding itself) can be determined by counting the acceptances of a finite automaton that processes the bits from left to right. This automaton, M_4, has three states, 0, 1/2 and 1, corresponding to the regular expressions

$$
\begin{aligned}
L_0 &= (10)^* + 0(10)^* + (0+1)^*00(10)^*, \\
L_{1/2} &= 0(10)^*1 + (10)^*1 + (0+1)^*11(01)^*0 + (0+1)^*00(10)^*1
\end{aligned}
$$

and

$$L_1 = (0+1)^*11(01)^*.$$

The state 0 is the initial state, and the state 1/2 is the unique accepting state. The automaton is strongly connected and aperiodic, and for the uniform distribution we have

$$p_n = \frac{2^n + (-1)^{n-1}}{3 \cdot 2^n}$$

and thus

$$e_n = \frac{n}{3} + C + O\left(\frac{1}{2^n}\right),$$

where $C = 1/9$. As usual, when applying this algorithm to a finite string we must flush out the automaton, this time by adding two 0s at the end of the string. One further non-zero digit will be produces if the automaton is in state 1 at that time. The probability of this event is $1/3 + +O(1/2^n)$ for uniformly distributed input. This condition is more complicated that that of the last bit of the input being 1; instead of appealing to (1.4), we shall have to show below that this probability is $1/3 + O(n^2/2^{n/3})$ for reciprocally distributed input.

To analyze this algorithm with reciprocally distributed input, we shall estimate the expected number of visits to the state 1/2 as the automaton processes the first n bits of the signifcand. It will be convenient to classify these according to the terms in a regular expression $L_{1/2}$. Since the first bit of a significand is always 1, the expected number of visits for the first term vanishes.

Let us consider the second term, $(10)^*1$. For $t \ge 1$, the probability that the first $2t-1$ bits of the significand are $(10)^{t-1}1$ is

$$\frac{1}{\log 2} \int_{(4^t-1)/6\cdot 4^{t-1}}^{(4^t+2)/6\cdot 4^{t-1}} \frac{dX}{X} = \log_2\left(\frac{4^t+2}{4^t-1}\right)$$

if $2t - 1 \le n$. Thus the expected number of visits to state 1/2 attributable to this term is

$$\sum_{1 \le t \le (n+1)/2} \log_2\left(\frac{4^t+2}{4^t-1}\right) = \sum_{1 \le t < \infty} \log_2\left(\frac{4^t+2}{4^t-1}\right) + O\left(\frac{1}{2^n}\right), \tag{3.5}$$

where we have used the estimate

$$\log_2\left(\frac{4^t+2}{4^t-1}\right)=O\left(\frac{1}{4^t}\right).$$

Next let us consider the third term, $(0+1)^*11(01)^*0$. For $t\ge 1$, the probability that the first $2t+1$ bits of the significand are $11(01)^{t-1}0$ is

$$\frac{1}{\log 2}\int_{(5\cdot 4^t-2)/6\cdot 4^t}^{(5\cdot 4^t+1)/6\cdot 4^t}\frac{dX}{X}=\log_2\left(\frac{5\cdot 4^t+1}{5\cdot 4^t-2}\right)$$

if $2t+1\le n$. For $s\ge 1$ and $t\ge 1$, the probability that the first $s+2t+1$ bits of the significand match the expression $(0+1)^s 11(01)^{t-1}0$ is

$$\sum_{2^{s-1}+1\le j\le 2^s}\frac{1}{\log 2}\int_{\left((6j-1)4^t-2\right)/6\cdot 2^s\cdot 4^t}^{\left((6j-1)4^t+1\right)/6\cdot 2^s\cdot 4^t}\frac{dX}{X}=\sum_{2^{s-1}+1\le j\le 2^s}\log_2\left(\frac{(6j-1)4^t+1}{(6j-1)4^t-2}\right)$$

if $s+2t+1\le n$. Thus, for $t\ge 1$, the expected number of visits to state $1/2$ attributable to the expression $(0+1)^*11(01)^{t-1}0$ is

$$J_{n,t}=\sum_{1\le j\le 2^{n-2t-1}}\log_2\left(\frac{(6j-1)4^t+1}{(6j-1)4^t-2}\right).$$

We have

$$\begin{aligned}
J_{n,t}&=\log_2\prod_{1\le j\le 2^{n-2t-1}}\frac{(6j-1)4^t+1}{(6j-1)4^t-2}\\
&=\frac{1}{2\cdot 4^t}\log_2\prod_{1\le j\le 2^{n-2t-1}}\left(\frac{(6j-1)4^t+1}{(6j-1)4^t-2}\right)^{2\cdot 4^t}\\
&=\frac{1}{2\cdot 4^t}\log_2\left(\frac{(6\cdot 2^{n-2t-1}-1)4^t-2}{5\cdot 4^t-2}\right)\times\\
&\qquad\prod_{1\le j\le 2^{n-2t-1}}\frac{\left((6j-1)4^t+1\right)^{2\cdot 4^t}}{\left((6j-1)4^t-2\right)^{2\cdot 4^t-1}\left((6j+5)4^t-2\right)}\\
&=\frac{n-2t-1}{2\cdot 4^t}+\frac{1}{2\cdot 4^t}\log_2\left(\frac{6\cdot 4^t}{5\cdot 4^t-2}\right)\times\\
&\qquad\prod_{1\le j\le 2^{n-2t-1}}\frac{\left((6j-1)4^t+1\right)^{2\cdot 4^t}}{\left((6j-1)4^t-2\right)^{2\cdot 4^t-1}\left((6j+5)4^t-2\right)}+O\left(\frac{1}{2^n}\right)\\
&=\frac{n-2t-1}{2\cdot 4^t}+\frac{1}{2\cdot 4^t}\log_2\left(\frac{6\cdot 4^t}{5\cdot 4^t-2}\right)\times\\
&\qquad\prod_{1\le j<\infty}\frac{\left((6j-1)4^t+1\right)^{2\cdot 4^t}}{\left((6j-1)4^t-2\right)^{2\cdot 4^t-1}\left((6j+5)4^t-2\right)}+O\left(\frac{1}{2^{n-4t}}\right),
\end{aligned}\tag{3.6}$$

where we have used the estimates

$$\log_2\big((6\cdot 2^{n-2t-1}-1)4^t-2\big) = (n-2t-1)+\log_2(6\cdot 4^t)+O\left(\frac{1}{2^{n-2t}}\right)$$

and

$$\frac{\big((6j-1)4^t+1\big)^{2\cdot 4^t}}{\big((6j-1)4^t-2\big)^{2\cdot 4^t-1}\big((6j+5)4^t-2\big)} = 1+O\left(\frac{4^{2t}}{j^2}\right),$$

so that

$$\prod_{2^{n-2t-1}+1\le j<\infty}\frac{\big((6j-1)4^t+1\big)^{2\cdot 4^t}}{\big((6j-1)4^t-2\big)^{2\cdot 4^t-1}\big((6j+5)4^t-2\big)} = 1+O\left(\frac{4^{2t}}{2^n}\right).$$

We can evaluate the infinite product in (3.6) by using (1.6). This gives

$$\begin{aligned}\prod_{1\le j<\infty}\frac{\big((6j-1)4^t+1\big)^{2\cdot 4^t}}{\big((6j-1)4^t-2\big)^{2\cdot 4^t-1}\big((6j+5)4^t-2\big)} &= \frac{\Gamma\left(\frac{5\cdot 4^t-2}{6\cdot 4^t}\right)^{2\cdot 4^t-1}\Gamma\left(\frac{11\cdot 4^t-2}{6\cdot 4^t}\right)}{\Gamma\left(\frac{5\cdot 4^t+1}{6\cdot 4^t}\right)^{2\cdot 4^t}}\\ &= \left(\frac{5\cdot 4^t-2}{6\cdot 4^t}\right)\left(\frac{\Gamma\left(\frac{5\cdot 4^t-2}{6\cdot 4^t}\right)}{\Gamma\left(\frac{5\cdot 4^t+1}{6\cdot 4^t}\right)}\right)^{2\cdot 4^t},\end{aligned}$$

where we have used $\Gamma\big((11\cdot 4^t-2)/6\cdot 4^t\big) = (5\cdot 4^t-2)\,\Gamma\big((5\cdot 4^t-2)/6\cdot 4^t\big)/6\cdot 4^t$ (see Whittaker and Watson [W2], Section $12\cdot 12$). Substituting this result into (3.5) yields

$$J_{n,t} = \frac{n-2t-1}{2\cdot 4^t}+\log_2\left(\frac{\Gamma\left(\frac{5\cdot 4^t-2}{6\cdot 4^t}\right)}{\Gamma\left(\frac{5\cdot 4^t+1}{6\cdot 4^t}\right)}\right)+O\left(\frac{1}{2^{n-4t}}\right). \tag{3.7}$$

The error term in (3.7) makes it useful only when t is small. When t is large, we shall use the following much cruder estimate. For $s\ge 0$ and $t\ge 1$, the probability that the first $s+2t+1$ bits of the significand match the expression $(0+1)^s 11(01)^{t-1}0$ is a sum of at most 2^s terms, each of which is the integral of a bounded distribution function over an interval of length $1/2^{s+2t+1}$. Thus this probability is $O(1/2^{2t})$. Summing over the $n-2t$ possible values of s, we have that the expected number of visits to state $1/2$ attributable to the expression $(0+1)^* 11(01)^{t-1}0$ satisfies

$$J_{n,t} = O\left(\frac{n}{2^{2t}}\right). \tag{3.8}$$

Summing over the $\lfloor (n-1)/2 \rfloor$ possible values of t, we have that the expected number of visits to state 1/2 attributable to the third term, $(0+1)^*11(01)^*0$, is

$$\sum_{1 \le t \le (n-1)/2} J_{n,t} = \frac{n}{6} - \frac{11}{18} + \sum_{1 \le t < \infty} \log_2 \left(\frac{\Gamma\left(\frac{5 \cdot 4^t - 2}{6 \cdot 4^t}\right)}{\Gamma\left(\frac{5 \cdot 4^t + 1}{6 \cdot 4^t}\right)} \right) + O\left(\frac{n^2}{2^{n/3}}\right), \tag{3.9}$$

where we have used (3.7) for $1 \le t \le n/6$ and (3.8) for $n/6 < t \le (n-1)/2$, together with the estimate

$$\log_2 \left(\frac{\Gamma\left(\frac{5 \cdot 4^t - 2}{6 \cdot 4^t}\right)}{\Gamma\left(\frac{5 \cdot 4^t + 1}{6 \cdot 4^t}\right)} \right) = O\left(\frac{1}{4^t}\right),$$

so that

$$\sum_{1 \le t \le n/6} \log_2 \left(\frac{\Gamma\left(\frac{5 \cdot 4^t - 2}{6 \cdot 4^t}\right)}{\Gamma\left(\frac{5 \cdot 4^t + 1}{6 \cdot 4^t}\right)} \right) = \sum_{1 \le t < \infty} \log_2 \left(\frac{\Gamma\left(\frac{5 \cdot 4^t - 2}{6 \cdot 4^t}\right)}{\Gamma\left(\frac{5 \cdot 4^t + 1}{6 \cdot 4^t}\right)} \right) + O\left(\frac{1}{2^{n/3}}\right).$$

Finally let us consider the fourth term, $(0+1)^*00(10)^*1$. A similar derivation gives that the expected number of visits to state 1/2 attributable to the fourth term is

$$\frac{n}{6} - \frac{11}{18} + \sum_{1 \le t < \infty} \log_2 \left(\frac{\Gamma\left(\frac{7 \cdot 4^t - 1}{6 \cdot 4^t}\right)}{\Gamma\left(\frac{7 \cdot 4^t + 2}{6 \cdot 4^t}\right)} \right) + O\left(\frac{n^2}{2^{n/3}}\right), \tag{3.10}$$

Summing the contributions in (3.5), (3.9) and (3.10), we obtain an estimate for the total expected number of acceptances

$$e_n = \frac{n}{3} + B + O\left(\frac{n^2}{2^{n/3}}\right),$$

where

$$\begin{aligned} B = -\frac{11}{9} &+ \sum_{1 \le t < \infty} \log_2 \left(\frac{4^t + 2}{4^t - 1} \right) \\ &+ \sum_{1 \le t < \infty} \log_2 \left(\frac{\Gamma\left(\frac{5 \cdot 4^t - 2}{6 \cdot 4^t}\right)}{\Gamma\left(\frac{5 \cdot 4^t + 1}{6 \cdot 4^t}\right)} \right) \\ &+ \sum_{1 \le t < \infty} \log_2 \left(\frac{\Gamma\left(\frac{7 \cdot 4^t - 1}{6 \cdot 4^t}\right)}{\Gamma\left(\frac{7 \cdot 4^t + 2}{6 \cdot 4^t}\right)} \right). \end{aligned}$$

Numerical computation gives $B = 0.42875\ldots$. (The first bit of a significand is always 1, which adds 1/2 to the expected number of acceptances. But the second bit is never accepted with the reciprocal distribution, since it is always preceded by a 1, whereas it is accepted with probability 1/4 for the uniform distribution; this subtracts 1/4 from the expected number of extra acceptances. The remaining bits lead to differences with more complicated behavior, but with diminishing magnitudes, so the limit of the number of extra acceptances, $B - C = 0.31764\cdots$, is somewhat more that 1/4.)

We must now estimate the probability that the automaton is in state 1 after processing the first n bits of the significand. Yet another similar derivation gives that the expected number of visits to state 1 due to the first n bits of the significand is

$$g_n = \frac{n}{3} - \frac{8}{9} + \sum_{1 \le t < \infty} \log_2 \left(\frac{\Gamma\left(\frac{10\cdot 4^{t-1}-1}{3\cdot 4^t}\right)}{\Gamma\left(\frac{10\cdot 4^{t-1}+2}{3\cdot 4^t}\right)} \right) + O\left(\frac{n^2}{2^{n/3}}\right).$$

Thus the probability that the automaton visits state 1 after the last of these n bits is

$$g_n - g_{n-1} = \frac{1}{3} + O\left(\frac{n^2}{2^{n/3}}\right).$$

As was mentioned before, it is necessary add this contribution to e_n to obtain the number of operations when recoding a string of length n.

4. Conclusion

Theorem 2.3 shows that for a sufficiently well-behaved distribution of a real number X on the interval $[0,1)$, the difference between the entropy of the first n bits in the binary expansion of X and the entropy of n uniformly distributed random bits converges to a constant exponentially fast. This constant is easily determined from the distribution of X. Theorem 2.1 shows that this implies that, for a strongly connected and aperiodic finite automaton M, the difference between the expected acceptance count for the first n bits in the binary expansion of X and the expected acceptance count for n uniformly distributed random bits also converges to a constant exponentially fast. This theory does not predict this latter constant, which (as can be seen by the examples of M_0 and M_1 in Section 1) depends on M as well as the distribution of X. For many natually arising examples of automata and probability distributions arising in the analysis of arithmetic operations, however, these constants can be determined explicitly by detailed analysis of the particular situation.

5. References

[B1] A. D. Booth, "A Signed Binary Multiplication Technique", *Quart. J. Mech. Appl. Math.*, 4 (1951) 236–240.

[B2] A. D. Booth, "Review of 'A Proof of the Modified Booth's Algorithm for Multiplication' by Louis P. Rubinfeld", *Math. Rev.*, 53 #4610.

[F1] C. V. Freiman, "Statistical Analysis of Certain Binary Division Algorithms", *Proc. IRE*, 49 (1961) 91–103.

[F2] C. Frougny, "On-the-Fly Algorithms and Sequential Machines", *IEEE Trans. on Computers*, 49 (2000) 859–863.

[H] R. W. Hamming, "On the Distribution of Numbers", *Bell System Tech. J.*, 49 (1970) 1609–1625.

[L1] M. Lehman, "High-Speed Digital Multiplication", *IRE Trans. on Electronic Computers*, 6 (1957) 204–205.

[L2] M. Lehman, "Short-Cut Multiplication and Division in Automatic Binary Digital Computers", *Proc. IEE*, 105 B (1958) 496–504.

[M] O. L. MacSorley, "High-Speed Arithmetic in Binary Computers", *Proc. IRE*, 49 (1961) 67–91.

[N] S. Newcomb, "Note on the Frequency of Use of the Different Digits in Natural Numbers", *Amer. J. Math.*, 4 (1881) 39–40.

[R] G. W. Reitwiesner, "Binary Arithmetic", *Advances in Computers*, 1 (1960) 232–308.

[S] C. E. Shannon, "A Mathematical Theory of Communication", *Bell System Tech. J.*, 27 (1948) 379–423, 623–655.

[T] K. D. Tocher, "Techniques of Multiplication and Division for Automatic Binary Computers", *Quart. J. Mech. Appl. Math.*, 11 (1958) 364–384.

[W1] J. Wallis, *Arithmetica Infinitorum*, Oxford, 1656.

[W2] E. T. Whittaker and G. N. Watson, *A Course of Modern Analysis*, Cambridge, 1963.