Information Processing Letters 26 (1987/88) 23–28
North-Holland

15 September 1987

ON THE LENGTH OF WORD CHAINS

Jean BERSTEL * and Srečko BRLEK

Department of Mathematics and Computer Science, University of Québec at Montréal, C.P. 8888, Montréal, Québec, Canada H3C 3P8

Communicated by L. Boasson
Received December 1986
Revised February 1987

Word chains are an extension of addition chains to words. We show that over a q-letter alphabet, any long enough word admits a word chain of length at most $(1+\varepsilon)n/\log_q n$, for a fixed arbitrary $\varepsilon > 0$; there exist words with no chain shorter than $n/\log_{q-1} n$. Several examples are given. Finally, we show that words with few factors have short chains.

Keywords: Word, addition chain

1. Introduction

Recently, Diwan [1] suggested to generalize the well-known notion of addition chains (see, e.g., [2]) to words, in the following way. A sequence of words

$$w_1, \ldots, w_r$$

is a *word chain* if for each w_i there are indices $j, k < i$ with $w_i = w_j w_k$. (By convention, w_j is a letter of the underlying alphabet if $j \leqslant 0$.) Clearly, addition chains are exactly word chains over a one-letter alphabet. The word chain is said to compute a word w if w belongs to the chain. The *chain length* of w is the smallest length of a word chain computing w.

Besides the idea of generalizing addition chains, the interest of this notion comes from the intuitively clear fact that it takes into account regularities in words, much better than for instance finite automata can do. This claim will be supported by some of the subsequent examples.

It is well known that the length of a shortest addition chain for some integer n is basically log n. The aim of this article is to show that this is no longer true for word chains. Our main result (Propositions 3.1 and 3.2) states that a word of length n over a q-letter alphabet can be computed in $n/\log_q n$ steps, and that there are words achieving this bound, up to a constant factor. Several examples will be given of families of words which can be computed in shorter length. The most striking example is perhaps the set of overlap-free words over a two-letter alphabet. Each such word can be computed in logarithmic length. Finally, we consider words having only few factors, i.e., having only polynomially many factors of given length. For these words, we show that there is a clear improvement on the chain length (Proposition 5.1). As a special case, we obtain that words with a linear number of factors admit word chains of length $\theta(\sqrt{n})$, where n is the length of the word.

2. Definitions and notation

Let A be a q-letter alphabet. A word chain over A is a sequence

$$c(w_{1-q}, \ldots, w_0, w_1, \ldots, w_r) \qquad (1)$$

* On sabbatical leave from LITP and University of Paris VI.

of words such that $A = \{w_{1-q}, \ldots, w_0\}$, and for each i ($1 \leqslant i \leqslant r$) there exist j, k with $1 - q \leqslant j, k < i$ such that

$$w_i = w_j w_k. \tag{2}$$

The *length* of the word chain c is the integer r and is denoted by $|c|$. The word chain c is said to *compute* a word w if $w = w_i$ for some $i \in \{1 - q, \ldots, r\}$. The *chain length* of a word w is the integer

$$\ell(w) = \min\{|c| : c \text{ computes } w\}.$$

It is easily seen that, in chain (1), $|w_i| \leqslant 2^i$ for $0 \leqslant i \leqslant r$. Thus, $\ell(w) \geqslant \log |w|$ for any nonempty word w. We shall see below that, for alphabets with more than one letter, more precise bounds can be given.

3. Results

3.1. Proposition. *Let* A *be a* q*-letter alphabet. For an arbitrary* $\varepsilon > 0$ *there is a constant* n_0 *such that, for any word* $w \in A^*$ *of length* $n \geqslant n_0$, *there exists a word chain computing* w *of length* $\leqslant (1 + \varepsilon)n/\log_q n$.

Proof. Let w be a word of length n, and consider an integer p, with $1 \leqslant p \leqslant n$. The word w factorizes into

$$w = w_1 w_2 \ldots w_{\lceil n/p \rceil},$$

where $|w_i| = p$ for $i = 1, 2, \ldots, \lceil n/p \rceil - 1$, and $|w_{\lceil n/p \rceil}| \leqslant p$. In order to compute w, it suffices to compute the whole set of q^p words of length p over A. Each such word can clearly be computed sequentially in $p - 1$ steps; thus, all w_i's (even the last) can be computed in $(p - 1)q^p$ steps. Then, $\lceil n/p \rceil - 1 \leqslant \lfloor n/p \rfloor$ steps suffice to get the word w. Thus,

$$\ell(w) \leqslant (p - 1)q^p + \lfloor n/p \rfloor.$$

Consequently,

$$\ell(w) \leqslant f(n),$$

where

$$f(n) = \min_{1 \leqslant p \leqslant n} \{pq^p + n/p\}.$$

Now consider the real function

$$g(x) = xq^x + n/x, \quad x \in [1, n].$$

For $n \geqslant q^2$ it has one extremum in [1, n] which is a minimum since $g''(x)$ is positive. The argument satisfies the functional equation $g'(x) = 0$, that is,

$$q^x(1 + x \log q) - n/x^2 = 0$$

or, equivalently,

$$xq^x = \frac{n}{x(1 + x \log q)}. \tag{3}$$

Substituting (3) into g(x) and denoting the argument of the minimum by x(n) we have

$$g(x(n)) = \frac{n}{x(n)}\left(1 + \frac{1}{1 + x(n) \log q}\right).$$

Let $m = \lfloor x(n) \rfloor$; then,

$$f(n) = \min\{g(m), g(m + 1)\}.$$

Then it follows that

$$g(x(n)) \leqslant f(n) \leqslant g(x(n) + 1). \tag{4}$$

We shall achieve the result in two steps; first, we show that f(n) and g(n) verify

$$\lim_{n \to \infty} f(n)/g(x(n)) = 1,$$

and, second, we compute

$$\lim_{n \to \infty} g(x(n)).$$

Rewrite (3) as

$$n = x^2 q^x (1 + x \log q)$$

and observe that for $k > q$ and x sufficiently large the following inequality holds:

$$n = x^2 q^x (1 + x \log q) < k^x.$$

Therefore, the inverse functions are related by $x(n) > \log_k n$. As a first consequence, since $x(n) \to \infty$ as $n \to \infty$,

$$\lim_{n \to \infty} \frac{g(x(n))}{g(x(n) + 1)} = \lim_{n \to \infty} \frac{x(n) + 1}{x(n)} = 1.$$

Combining this result with (4), it is easy to see

that

$$1 \leqslant \lim_{n\to\infty} \frac{f(n)}{g(x(n))} \leqslant \lim_{n\to\infty} \frac{g(x(n)+1)}{g(x(n))} = 1$$

and the first claim clearly follows.

As a second consequence, g(x(n)) satisfies

$$g(x(n)) = \frac{n}{x(n)}\left(1 + \frac{1}{1 + x(n)\log q}\right) \leqslant \frac{n}{\log_k n}\left(1 + \frac{1}{1 + \log_k n \log q}\right).$$

Therefore, g(x(n)) is $\theta(n/\log_k n)$. Finally, remark that k can be computed in the following way:

$$\lim_{n\to\infty} \frac{g(x(n))}{n/\log n} = \lim_{n\to\infty} \frac{\log n}{x(n)}\left(1 + \frac{1}{1 + x(n)\log q}\right) = \lim_{n\to\infty} \frac{\log n}{x(n)}.$$

Then, substituting

$$n = x^2 q^x (1 + x \log q)$$

in this last expression, we have

$$\log(x^2 q^x (1 + x \log q)) = 2 \log x + x \log q + \log(1 + x \log q)$$

and this limit is

$$\lim_{x\to\infty} \frac{2 \log x + x \log q + \log(1 + x \log q)}{x} = \log q.$$

Thus,

$$\lim_{n\to\infty} g(x(n))/(n/\log_q n) = 1,$$

and the proof is complete. □

3.2. Proposition. *Let* A *be a* q*-letter alphabet, with* $q \geqslant 3$.

There exist words $w \in A^*$ *such that* $\ell(w) \geqslant n/\log_{q-1} n$, *where* $n = |w|$.

Proof. We consider a special symbol \$ in A, and set $B = A - \{\$\}$. Fix an integer $N > 1$ and consider the word

$$w = \$u_1\$u_2\$ \ldots \$u_{(q-1)^N}\$,$$

where $u_1, \ldots, u_{(q-1)^N}$ is some enumeration of the set of words of length N over B. Let

$$c = (a, b, \ldots, \$, w_1, \ldots, w_r)$$

be a word chain of minimal length computing w. A \$-*step* in c is an index i such that in this i*th* step $w_i = w_j w_k$, and both w_j and w_k have at least one occurrence of the letter \$. We show that c has at least $(q-1)^N$ \$-steps.

For this, we associate to each word u_s ($1 \leqslant s \leqslant (q-1)^N$) its *rank* ρ_s defined as follows. The rank of u_s is the smallest index i such that $\$u_s\$$ is a factor of w_i. This definition makes sense because each u_s is a factor of $w_r = w$. Observe that ρ_s is a \$-step.

It suffices to show that distinct words u_s have distinct ranks. Assume the contrary. Then $\rho_s = \rho_t$ for some s, t with $s \neq t$. Set $i = \rho_s = \rho_t$. Then

$$w_i = w_j w_k$$

for some $j, k < i$; the words $\$u_s\$$ and $\$u_t\$$ both appear in w_i, but none of them is a factor of w_j and w_k. Because \$ plays the role of a marker, it follows that $u_s = u_t$, whence $s = t$.

This proves that there are at least $(q-1)^N$ \$-steps in the chain c. Consequently,

$$\ell(w) \geqslant (q-1)^N.$$

Setting

$$n = |w| = (q-1)^N(N+1) + 1,$$

it follows that

$$\ell(w) \geqslant n/\log_{q-1} n,$$

since, for sufficiently large n,

$$\frac{n}{\log_{q-1} n} \leqslant \frac{(q-1)^N(N+1)+1}{N+2} \leqslant (q-1)^N. \quad \square$$

Putting both results together, we obtain, for

$\varepsilon > 0$ and infinitely many integers n, the bounds

$$\frac{n}{\log_{q-1} n} \leqslant \ell(w) \leqslant \frac{(1+\varepsilon)n}{\log_q n} \quad (|w| = n).$$

3.3. Remark. A convenient representation of a word chain is the sequence of pairs of indices: the *ith* step

$w_i = w_j w_k$

is represented by (j, k). Thus, the binary notation of a word chain of length r requires roughly r log r space. The bounds given above just mean that there is no data compression by word chains.

4. Examples

In this section, we give some examples of families of words which are 'easy' to compute in the sense of word chains.

4.1. Example (*D0L systems*). Consider an alphabet A, and a morphism

$h: A^* \to A^*$.

Given a word $u \in A^*$ and an integer n, the *nth* iterate $h^n(u)$ can be computed by a word chain of length less than

$n \cdot \|h\| + |u|$,

where

$$\|h\| = \sum_{a \in A} |h(a)|.$$

We indeed proceed by computing the set of words

$\{h^n(a) : a \in A\}$

inductively as follows. The set

$\{h(a) : a \in A\}$

is computed in less than $\|h\|$ steps. Next, since

$h^n(a) = h(h^{n-1}(a))$,

each $h^n(a)$ can be computed from the set $\{h^{n-1}(a) : a \in A\}$ in at most $|h(a)|$ steps. Thus,

$\{h^n(a) : a \in A\}$

is computed from the alphabet in $\|h\| \cdot n$ steps, as claimed.

Observe that the length of $h^n(u)$ usually grows exponentially with n (see [6] for a systematic discussion). Thus, we have a logarithmic bound.

4.2. Example. Each word w_n of the form

$w_n = baba^2ba^3b \ldots ba^nb$

has chain length $\theta(n)$. It is easily seen that there is a word chain of length $2n-1$. Conversely, an adaptation of the proof of Proposition 3.2 shows that at least n steps are required. Thus,

$$\ell(w_n) = \theta\left(\sqrt{|w_n|}\right).$$

4.3. Example (*overlap-free words*). Consider the alphabet $A = \{0, 1\}$. A word is *overlap-free* if it has no factor of the form xuxux, with x, u words, and x nonempty (for more details, see [3]). Overlap-free words over a two-letter alphabet have completely been characterized by Restivo and Salemi [5] as follows.

Define sequences of words by

$$a_0 = 0 \qquad b_0 = 1$$
$$a_{n+1} = a_n b_n b_n a_n, \qquad b_{n+1} = b_n a_n a_n b_n,$$

and sequences of sets by

$$H_n = \{1, a_n, a_n a_n, a_n a_n b_n, a_n a_n b_n a_n, a_n b_n, a_n b_n a_n, a_n b_n a_n b_n\},$$
$$G_n = H_n \cup \overline{H}_n,$$
$$D_n = \tilde{G}_n,$$

where $\overline{H}_n$ is obtained from H_n by exchanging a's and b's, and $\tilde{G}_n$ is the set of reversals of G_n. Then, every overlap-free word w may (uniquely) be written as

$$w = g_0 g_1 \cdots g_{k-1} u d_{k-1} d_{k-2} \cdots d_0,$$

where $g_i \in G_i$, $d_i \in D_i$, and u is the product of at least 2 and at most 11 words in $\{a_k, b_k\}$.

In order to construct a word chain for w, we observe that each pair (a_{n+1}, b_{n+1}) can be computed in four steps from (a_n, b_n), and that, given a_n, b_n, a word in G_n, D_n, is computed in at most two steps. Thus, w can be computed by a word

chain of length at most $2k + 2k + 4k + 2k + 10 = 10(k + 1)$. Next, for $n \geqslant 0$,

$$|a_n| = 4^n, \qquad |b_n| = 4^n,$$

$$1 \leqslant |g_n| \leqslant 4^{n+1} \quad (g_n \in G_n),$$

which easily implies that

$$2k + 2 \times 4^k \leqslant |w| < 14 \times 4^k.$$

This shows that the chain length of w is $\theta(\log |w|)$.

5. Words with few factors

As already mentioned, word chains take into account the structure of the factors of the word they compute. This will now be demonstrated by the observation that words with few factors have short chains.

To be more precise, we need some notation. Given a word w, we denote by $\mathscr{F}_w(h)$ the set of factors of length h of w, and we denote by $\varphi_w(h)$ the size of $\mathscr{F}_w(h)$:

$$\varphi_w(h) = \text{Card } \mathscr{F}_w(h) \quad (h \geqslant 1).$$

We omit the subscript when no confusion can arise.

5.1. Proposition. *Let* w *be a word of length* n, *and assume that there are constants* $C \geqslant 1$, $p \in \mathbb{N}$, $p \geqslant 1$ *such that*

$$\varphi_w(h) \leqslant Ch^p \quad (1 \leqslant h \leqslant \lceil n^{1/(p+1)} \rceil).$$

Then,

$$\ell(w) < 6Cn^{p/(p+1)}.$$

For $p = 1$ we get the following special case.

5.2. Corollary. *Let* w *be a word of length* n, *and assume that* $\varphi_w(h) = \theta(h)$, *for* $h = 1, 2, \ldots, n$, *i.e, there is a linear number of factors of each length. Then*,

$$\ell(w) = O(\sqrt{n}).$$

There is still a gap between the upper bound given by Proposition 5.1 and some of the lower bounds derived before. Thus, the word

$$w_n = baba^2ba^3b \ldots ba^nb$$

of length $\theta(n^2)$ has a word chain of length $\theta(\sqrt{|w_n|})$. Since $\varphi_{w_n}(h) = \theta(h^2)$, our proposition just gives the length $O(|w_n|^{2/3})$.

Proof of Proposition 5.1. The proof is a refinement of the construction in Proposition 3.1, and uses an improvement of the length of computation due to the size of $\varphi(h)$.

First, we introduce a straightforward extension of the notion of word chain. We say that a word chain computes a (finite) set S of words if each element of S appears in the chain.

Now let w be a word of length n, and consider a fixed integer m with $1 \leqslant m < n$ whose value will be determined later. Next, consider any addition chain for m,

$$t = (t_0, t_1, \ldots, t_r),$$

with $t_0 = 1$ and $t_r = m$. This chains is used to 'program' the construction of a word chain for $\mathscr{F}(m)$ $(= \mathscr{F}_w(m))$ as follows. For each t_i, we consider the set $\mathscr{F}(t_i)$.

If $t_i = t_j + t_k$ (j, k < i), then clearly

$$\mathscr{F}(t_i) \subset \mathscr{F}(t_j)\mathscr{F}(t_k)$$

(where $\mathscr{F}(t_j)\mathscr{F}(t_k) = \{uv \mid u \in \mathscr{F}(t_j) \text{ and } v \in \mathscr{F}(t_k)\}$). Consequently, the set $\mathscr{F}(t_i)$ can be computed from the sets $\mathscr{F}(t_j)$ and $\mathscr{F}(t_k)$ in $\varphi(t_i)$ chain steps, just by picking the convenient elements in the previously constructed word chain. Since $\mathscr{F}(t_0)$ is the alphabet, it follows that there is a word chain computing $\mathscr{F}(m)$ of length

$$\varphi(t) = \varphi(t_1) + \cdots + \varphi(t_r),$$

where $t = (t_0, t_1, \ldots, t_r)$ is any addition chain for m.

In order to get a good estimation of this number, we use the following *special* addition chains. If $m = 2^s$, then the special addition chain for m is simply

$$(1, 2, 2^2, \ldots, 2^s).$$

If, on the contrary, $m = 2^s + m'$ with $1 \leqslant m' < 2^s$, then let u be such that $2^u \leqslant m' < 2^{u+1}$. Of course,

$u < s$. Then the special chain for m is

$$t = (t_0, t_1, \ldots, t_{r'}, 2^{u+1}, \ldots, 2^s, m), \quad (5)$$

where $t' = (t_0, t_1, \ldots, t_{r'})$ is the special chain for m'. (As an example, the sequence (1, 2, 4, 8, 9, 16, 32, 41, 64, 105) is the special chain for 105.)

Now we use the hypothesis, namely

$$\varphi(h) \leqslant Ch^p,$$

and assuming that $m = \lceil n^{1/(p+1)} \rceil$ we claim that

$$\varphi(t) = \varphi(t_1) + \cdots + \varphi(t_r) < 3Cm^p - 1,$$

provided t is the special chain for m.

Indeed, consider first the case where

$$t = (1, 2, 2^2, \ldots, 2^s);$$

then, $m = 2^s$ and

$$\varphi(t) \leqslant C(2^p + 2^{2p} + \cdots + 2^{sp}) \leqslant 2C(2^{sp} - 1).$$

Next, we assume that the special chain for m is given by (5). By induction,

$$\varphi(t_1) + \cdots + \varphi(t_{r'}) \leqslant 3C(m'^p - 1).$$

Thus, setting $q = 2^p$,

$$\frac{1}{C}\varphi(t) \leqslant 3m'^p - 1 + \frac{q^{s+1} - q^{u+1}}{q-1} + m^p.$$

Since $m' < 2^{u+1}$, we have $m'^p < q^{u+1}$, whence

$$\frac{1}{C}\varphi(t) \leqslant 2m'^p - 1 + q^{u+1}\frac{q-2}{q-1} + 2q^s - q^s\frac{q-2}{q-1} + m^p.$$

Thus, since $u + 1 \leqslant s$, we get

$$\frac{1}{C}\varphi(t) \leqslant 2m'^p + 2q^s + m^p - 1 = 2(m'^p + (2^s)^p) + m^p - 1 \leqslant 3m^p - 1,$$

because $m' + 2^s = m$. Thus,

$$\varphi(t) \leqslant 3C(m^p - 1) < 3Cm^p - 1.$$

We are now able to describe the word chain for w. It is first composed of a word chain computing the set $\mathscr{F}_w(m)$ with $m = \lceil n^{1/(p+1)} \rceil$. Next, the word w being factorized into

$$w = w_1 w_2 \ldots w_{\lceil n/m \rceil}$$

with each w_i (except perhaps the last) of length m, $\lceil n/m \rceil - 1$ steps suffice to compute w. The length of this chain is therefore bounded by

$$\begin{aligned} &3Cm^p - 1 + \lceil n/m \rceil + m - 1 \\ &\leqslant 3Cm^p + \lceil n/m \rceil + m - 1 \\ &\leqslant 3C(\lceil n^{1/(p+1)} \rceil)^p + n^{p/(p+1)} + n^{1/(p+1)} \\ &< 6Cn^{p/(p+1)}. \quad \square \end{aligned}$$

Acknowledgment

The authors acknowledge the helpful comments of the anonymous referee(s).

References

[1] A.A. Diwan, A new combinational complexity measure for languages, Rept., Computer Science Group, Tata Institute, Bombay, India, 1986.

[2] D.E. Knuth, The Art of Computer Programming, Vol. 2: Seminumerical Algorithms (Addison-Wesley, Reading, MA, 1969).

[3] M. Lothaire, Combinatorics on Words (Addison-Wesley, Reading, MA, 1983).

[4] A. Restivo and S. Salemi, On weakly square free words, Bull. EATCS 21 (1983) 49–56.

[5] A. Restivo and S. Salemi, Overlap free words on two symbols, in: D. Perrin, ed., Automata on Infinite Words, Lecture Notes in Computer Science, Vol. 192 (Springer, Berlin, 1985) 198–206.

[6] G. Rozenberg and A. Salomaa, The Mathematical Theory of L Systems (Academic Press, New York, 1980).