Publ. I.R.M.A. Strasbourg, 1988, 348/S-17
Actes 17^{e} Séminaire Lotharingien, p. 129-139

ON WORDS CHAINS

BY

SREČKO BRLEK

Abstract.- Word chains are an extension of addition chains to words. 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 and arbitrary $\varepsilon>0$; moreover, there exist words with no chain shorter than $n/\log_{q-1}(n)$. We study word chains for the Thue-Morse **M** word with a representation by binary trees. A conjecture on the enumeration of shortest word chains computing **M** is proposed.

Résumé.- Le concept de chaîne de mots est une extension naturelle du concept de chaîne d'addition. Sur un alphabet à q lettres, tout mot assez long possède une chaîne de mots de longueur inférieure ou égale à $(1+\varepsilon)n/\log_q(n)$; de plus, il existe des mots dont les chaînes sont de longueur plus grande que $n/\log_{q-1}(n)$. Nous étudions les chaînes de mots qui calculent le mot de Thue-Morse **M** à l'aide d'une représentation par arbres binaires et nous proposons une conjecture sur le dénombrement des chaînes les plus courtes qui calculent **M**.

1. Introduction

Fast computation of powers of monomials is a very old problem, and addition chains have been introduced as a general frame for its study (cf Knuth [4]). In order to get a convenient complexity measure for languages, A.A. Diwan [3] defined the notion of word chain on the free monoid A^* over a finite alphabet A. This notion appears as a natural generalization of addition chains, and is defined as follows. A sequence of words

$$w_1, \ldots\ldots, w_r$$

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 $j \leq 0$). 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.

It is well known that the length of a shortest addition chain for some integer n is basically $\log_2(n)$. This is no longer true for word chains. A word of length n over a q-letter alphabet can be computed in $n/\log_q(n)$ steps, and words achieving this bound, up to a constant factor, exist (Berstel and Brlek [1]). Regularities in words play a major

role, since they can be used to improve the chain length. In Berstel and Brlek [1], it is shown that there is a clear improvement in some cases .

In section 5, binary trees are used as a representation of word chains, and we analyze, in section 6, the chain length of the well known Thue-Morse word.

2. Definitions and notation

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

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

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

$$w_i = w_j w_k \ . \qquad (2)$$

Clearly, addition chains are exactly word chains over a one-letter alphabet. The *length* of the word chain $\mathbf{c}$ is the integer r and is denoted by $|\mathbf{c}|$. The word chain $\mathbf{c}$ 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

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

Straightforward extensions are given for sets of words as follows. For every non empty set $S \subset A^*$, $\mathbf{c}$ compute S, if and only if

$$\forall s \in S, s \in \mathbf{c},$$

and the chain length $\ell(S)$ of S, is defined similarly.

Observe that in chain (1), $|w_i| \le 2^i$ for $0 \le i \le r$. Therefore, for any non empty word w, $\ell(w) \ge \log(|w|)$. On the other hand, it is clear that every non empty word w is computed from the alphabet in $|w|-1$ steps, by concatenation of one letter at each step. We shall see later that more precise bounds can be given. In particular, when a word has regularities, a better result is in general achieved as is shown in the following example.

Example 1. Let $w = v\,i\,v\,i\,a\,n\,e$. This word has a square prefix, and this property can be used to compute it as follows:

step 1. v i
step 2. (v i) (v i)
step 3. (v i v i) a
step 4. (v i v i a) n
step 5. (v i v i a n) e

which yields the following word chain

$$\mathbf{c} = (a, e, i, n, v, v\,i, v\,i\,v\,i, v\,i\,v\,i\,a, v\,i\,v\,i\,a\,n, v\,i\,v\,i\,a\,n\,e).$$

3. Main results

We recall without proof the following results.

Proposition 3.1. (Berstel and Brlek [1])
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* $n \geq n_0$, *there exists a word chain* c *computing* w *of length* $|c| \leq (1+\varepsilon)\, n/\log_q(n)$.

Proposition 3.2. (Berstel and Brlek [1])
Let A *be a* q*-letter alphabet, with* $q \geq 3$. *There exist words* $w \in A^*$ *such that* $\ell(w) \geq n/\log_{q-1}(n)$, *where* $n = |w|$.

Putting both results together, we obtain, for $\varepsilon > 0$ and infinitely many integers n, the bounds

$$\frac{n}{\log_{q-1}(n)} \leq \ell(w) \leq (1+\varepsilon)\frac{n}{\log_q(n)} \qquad (|w|=n)$$

Observe also that if a word chain is represented by the sequence of pairs of indices (the i-th step, namely $w_i = w_j w_k$, is represented by (j,k)), then there is no data compression by word chains, because the binary notation of a word chain of length r requires roughly r.log(r) space.

In the following examples, we give an estimation of the chain length of some families of words which are "easy" to compute in the sense of word chains. We refer the reader to [1] for detailed proofs.

Example 2. (DoL systems). Consider an alphabet A, and a morphism $h : A^* \to A^*$. Given a word $u \in A^*$, and an integer n, the n-th iterate $h^n(u)$ can be computed by a word chain of length bounded by

$$\ell(h^n(u)) \leq n \cdot \Big(\sum_{a \in A} |h(a)| \Big) + |u| - 1$$

Since the length of $h^n(u)$ usually grows exponentially with n , we have a logarithmic bound.

Example 3. Each word w_n of the form

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

has chain length $\theta(n)$. It is easily seen that there is a word chain of length 2n-1. Thus

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

Example 4. (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 any overlap-free word w, its chain length $\ell(w)$ is $\theta(\log(|w|))$.

4. Words with few factors

As already mentioned, word chains take into account the structure of the factors of the word they compute. It will be shown that words with few factors have short chains.

Given a word w, we denote by $\mathfrak{F}_w(h)$ the set of factors of length h of w, and we denote by $\psi_w(h)$ the size of $\mathfrak{F}_w(h)$:

$$\psi_w(h) = \text{Card } \mathfrak{F}_w(h) \qquad (h \geq 1)$$

We omit the subscript if no confusion is possible.

Proposition 4.1. (Berstel and Brlek [1])
Let w *be a word of length* n, *and assume that there are constants* $C \geq 1$, $p \in \mathbb{N}$, $p \geq 1$ *such that*

$$\psi_w(h) \leq C\, h^p \quad , \quad (1 \leq h \leq \lceil n^{1/(p+1)} \rceil)$$

Then

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

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

Corollary. *Let* w *be a word of length* n, *and assume that* $\psi_w(h) = \theta(h)$, *for* $h=1,\ldots,n$, *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 this proposition and the lower bounds derived in particular cases as we shall see in the next example.

Example 5. Let $A=\{a,b\}$ and φ be the morphism $\varphi: A^* \to A^*$ given by $\varphi(a)= ab$ and $\varphi(b)=ba$. The Thue-Morse word M is defined by iteration of φ as follows:

$\varphi^2(a)$ = abba

$\varphi^3(a)$ = abbabaab

$\varphi^4(a)$ = abbabaabbaababba

:

M = abbabaabbaababbabaababbaabbabaabbaababbaabbabaababbabaabbaabab.....

The number of factors in M is given in the next two propositions.

Propositon 4.2 (Brlek [2])

For $m \geq 3$, *the function* $\mathcal{F}(m)$ *is given by*

$$\mathcal{F}(m) = \begin{cases} 6.2^{r-1} + 4p & 0 < p \leq 2^{r-1} \\ 8.2^{r-1} + 2p & 2^{r-1} < p \leq 2^r \end{cases}$$

where r *and* p *are uniquely determined by the equation*

$$m = 2^r + p + 1 \qquad 0 < p \leq 2^r$$

Proposition 4.3 (Brlek [2]) *The function* $\mathcal{F}(m)$ *satisfies*

$$\underline{\lim}_{m\to\infty} \frac{\mathcal{F}(m)}{m-1} = 3 \qquad \overline{\lim}_{m\to\infty} \frac{\mathcal{F}(m)}{m-1} = \frac{10}{3}$$

and the bounds are respectively attained for sequences of values of m *given by*

$$m = 2^r+1 \quad \textit{and} \quad m = 3.2^{r-1}+1$$

Therefore the numbers of factors of M is a linearly growing function, and we can apply the corollary following Proposition 4.1. We have thus

$$\mathcal{F}_{\varphi^n(a)}(m) \leq \frac{10}{3}(m-1) \leq \frac{10}{3} m$$

and the chain length is

$$\ell(\varphi^n(a)) < 6\frac{10}{3}(2^n)^{\frac{1}{2}}$$

It gives a rough bound as we shall see in section 6.

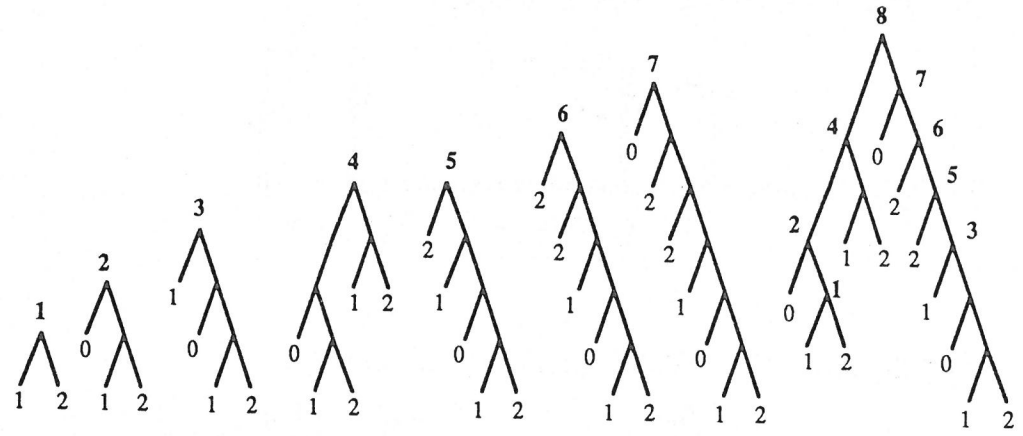
5. Word chains and binary trees

Every word chain corresponds to a binary tree constructed recursively by the following rule: given an alphabet A, for every pair of words $w_1, w_2 \in A^*$,

Exemple 6. A chain computing w = 012120221012 is

c = (0,1,2,12,012,1012,01212,21012,221012,0221012,012120221012)

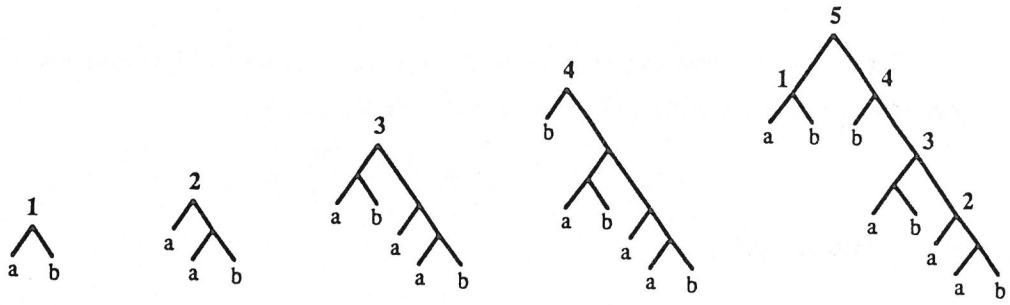
and the corresponding binary tree is constructed as follows



Exemple 7. One of the shortest chain computing w = abbabaab is

c = (a,b,ab,aab,abaab,babaab,abbabaab)

and the corresponding tree is obtained by the sequence



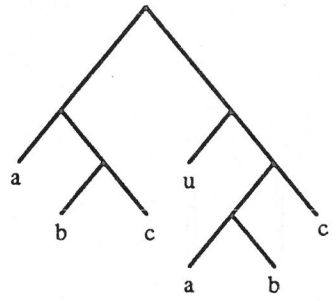
Remarks.

1. The word is read on the leaves from left to right, following the natural order .

2. It is always possible to rearrange the order of some elements in the chain, in such a way that concatenation operations correspond to the postfix order on the tree. In example 6 it suffices to commute operations 3 and 4.

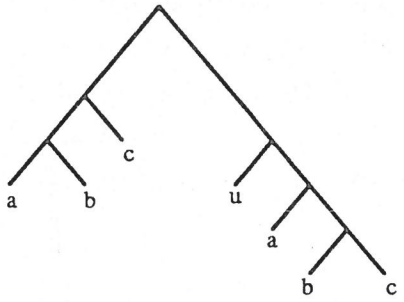
3. In general, this correspondance is not a bijection. Indeed, every binary tree with |w| leaves correspond to a parenthesing of the word w. However, distinct trees can yield the same chain as is shown in the following example

(a(bc))(u((ab)c))



c = (a,b,c,u,bc,abc,ab,uabc,abcuabc)

((ab)c))(u(a(bc)))



c = (a,b,c,u,ab,abc,bc,uabc,abcuabc)

This is due to the fact that the word *abcuabc* contains two occurences of a factor of length 3, which can be computed, according to the associative rule, in two different manners. Therefore, the factors *ab* et *bc* can commute in the chain. So we must take in account the order of elements in the chain and consequently the tree corresponding to the ordered chain **c**(w) will be denoted by $T_{\mathbf{c}(w)}$.

This last example indicates also that the chain length is not equal to the number of distinct sub-trees in general, because the factor *abc* is counted once, but it is represented by two distinct sub-trees. This is no longer true if the chain is minimal and we have the following caracterisation.

Proposition 5.1 *Let* A *be a finite alphabet, and* $w \in A^*$, *then*

$$\ell(w) = \min_{\mathbf{c}(w)} \left(\#\ \textit{distinct sub-trees of}\ T_{\mathbf{c}(w)} \right)$$

$C(w)$, $C_r(w)$, $C_{min}(w)$, will denote the number of chains respectively, distinct, of length r, and minimal, computing w. The number of addition chains for an integer n will be denoted by $C(n)$.

Proposition 5.2 *Let* A *be a* q*-lettrer alphabet , and* $w \in A^*$ *a word of length* $|w| = n$. *Then*

$$C(n) \leq C(w) \leq \frac{(2n-2)!}{n!\,(n-1)!}$$

Proof. If the letters of w are distinct, then $|w| = n \leq q$ and there is a bijection between word chains and binary trees with n leaves. Therefore the Catalan numbers provide the upper bound. On the other hand, if $w = a^n$, then $C(w) = C(n)$.

Among all chains computing a word, some are minimal, and their length and number is closely related to the strucure of the word. Indeed, if the letters of w are distinct, then all trees represent minimal chains and their length is $n-1$. In the case of the one-letter word $w = a^n$, its length $\ell(n)$ is given by (cf. Knuth [4])

$$\lceil \log_2 n \rceil \leq \ell(n) \leq \lfloor \log_2 n \rfloor + v(n) - 1$$

where $v(n)$ is the number of 1's in the binary representation of n.

6. A case study: the Thue-Morse word

In general chains are not stable under morphism iteration. However if the morphism is defined by one (and only one) monoid operation, then, new chains are constructed by iteration. In the case of the Thue-Morse word M ,we have

Proposition 6.1 *Let* $A = \{a,b\}$ *and* $\varphi : A^* \to A^*$ *defined by* $\varphi(a)=ab$, $\varphi(b)=ba$. *If* $\mathbf{c}$ *is a chain computing* $\varphi^n(a)$, *then* $A \cup \varphi(\mathbf{c})$ *computes* $\varphi^{n+1}(a)$, *and* $|A \cup \varphi(\mathbf{c})| = |\mathbf{c}| + 2$. *Moreover, if* $\mathbf{c}$ *is a minimal chain, then*

$$\ell(\varphi^{n+1}(a)) \leq |A \cup \varphi(\mathbf{c})|$$

Proof. Let $\mathbf{c}=(a,b, w_1, ..., \varphi^n(a))$. Then $A \cup \varphi(\mathbf{c})=(a,b, ab,ba, \varphi(w_1), ..., \varphi^{n+1}(a))$, where $\varphi(w_i)=\varphi(w_j w_k)=\varphi(w_j)\varphi(w_k)$. Moreover $\ell(\varphi^{n+1}(a)) \leq \ell(\varphi^n(a))+2 = |A \cup \varphi(\mathbf{c})|$.

Clearly, $\ell(\varphi^n(a)) \leq 2n-1$, because $2n-1$ is the length of the particular chain

$$\mathbf{c} = (a,b,ab,ba, \ldots, u_i, v_i, \ldots, u_n) \qquad (3)$$

which computes $u_n = \varphi^n(a)$. And A.A. Diwan [3] proposed the following concecture.

Conjecture 6.2. *The length of a shortest chain computing* $\varphi^n(a)$ *is*

$$\ell(\varphi^n(a)) = 2n-1.$$

The next result is immediate.

Proposition 6.3 *Under the same assumption, let* $S = \{ \varphi^i(a) : 1 \leq i \leq n \}$ *then*

(i) $\ell(S) = 2n-1$,

(ii) $\ell(\{ \varphi^n(a), \varphi^n(b) \}) = 2n$.

Proof. (i) In the chain $\mathbf{c} = (a,b,\varphi(a), \ldots, \varphi^2(a), \ldots, \varphi^i(a), \ldots, \varphi^{i+1}(a), \ldots, \varphi^n(a))$, for every i, $\varphi^{i+1}(a)$ is not square. Hence, there is at least one word between $\varphi^i(a)$ and

$\varphi^{i+1}(a)$, and consequently, $\ell(S) \geq 2n-1$. On the other hand, chain given by (3) computes S and its length is 2n-1. Point (ii) is easy to get by recurrence.

The following table lists the number of chain of each length for $\varphi^n(a)$. They have been computed on a SUN 3/50 workstation.

		longueur													
n	$\lvert\varphi^n(a)\rvert$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2	1													
2	4			5											
3	8					19	116	294							
4	16							19	342	?	?	?	?	?	?
5	32									19	?	?	?	?	?
6	64											19	?	?	?

Table 2. $C_r(\varphi^n(a))$ = number of chains of length r for $\varphi^n(a)$.

It contains the particular results:

1. $C(\varphi^n(a)) = \text{Catalan}(2^n)$, $n \leq 3$.
2. $C_{min}(\varphi^n(a)) = 19$, $n = 3,4,5,6$.

The first is not surprising, since $\varphi^n(a)$ has no factor of length 3, occuring more than once. The second is remarkable, since, according to proposition 6.1, it would mean that φ is stable for minimal chains. Therefore, we propose the conjecture:

Conjecture 6.4 $C_{min}(\varphi^n(a)) = 19$, $n \geq 3$.

Définition. *A leaf in a binary tree will be called* special *if its brother is not a leaf.*

The absence of special leaves is related to morphism iteration. Indeed, if **c** is a chain computing $\varphi^n(a)$, applying φ to the corresponding tree, consists in replacing each leaf by its image under φ (the leaves grow). Therefore, the tree corresponding to the chain $A \cup \varphi(\mathbf{c})$ computing $\varphi^{n+1}(a)$, has no special leaf. Conversly we have:

Proposition 6.5 *Let* **c** *be a chain computing* $\varphi^n(a)$, *such that the corresponding tree has no special leaf. Then, there exist* **c'**, *a chain computing* $\varphi^{n-1}(a)$, *such that*

$$\mathbf{c} = \varphi(\mathbf{c'}) \cup A$$

Proof. Since the tree has no special leaf, elements in the chain have even length. By proposition 3.1 (iv) in [2], there is no factor of the type aa nor bb in the chain. The only factors of length 2 are ab et ba and it suffices to cut the leaves.

Proposition 6.6 *Let* c *be a chain computing* $\varphi^n(a)$, *then*

(i) $\tilde{\bar{c}}$ *computes* $\varphi^n(a)$, *if* n *is odd*

(ii) $\tilde{c}$ *computes* $\varphi^n(a)$, *if* n *is even*

where the operations inversion ($\bar{}$) *and "mirror image"* ($\tilde{}$) *are defined by relations*

$$\bar{a} = b \; ; \quad \bar{b} = a \; ; \quad \overline{w_1 w_2} = \bar{w}_1 \bar{w}_2$$

$$\tilde{a} = a \; ; \quad \tilde{b} = b \; ; \quad w = w_1 w_2 \Leftrightarrow \tilde{w} = \tilde{w}_2 \tilde{w}_1$$

Proof. The chain property is stable under the operations "inversion" ($\bar{}$) and "mirror image" ($\tilde{}$)

$$w_i = w_j w_k \Leftrightarrow \bar{w}_i = \bar{w}_j \bar{w}_k$$

$$\Leftrightarrow \tilde{w}_i = \tilde{w}_k \tilde{w}_j$$

Finally we have eiher (ii) when n is even because $\varphi^n(a)$ is a palindrome, or (i) since $\varphi^n(a)$ is the mirror image of its inverse, when n is odd (cf. Lothaire[5]).

Remark that operation "mirror image" on a chain amounts to take the symmetric tree of the corresponding tree.

To prove Diwan's conjecture it suffices to establish the next result.

Conjecture 6.7 *Let* c_1 *be a minimal chain computing* $\varphi^n(a)$ *having a factor of odd length. Then there exist a minimal chain* c_2 *having no factor of odd length.*

Under these assumptions, the corresponding tree has at least two special leaves, and it suffices to show that it can be reduced, using operations preserving chain length (i.e. the number of distinct sub-trees) to a tree with no special leaf.

If conjecture 6.7 is true, then it is easy to deduce conjecture 6.4 on the chain length for $\varphi^n(a)$. Indeed, we proceed by contradiction: let m be the smallest integer such that $\ell(\varphi^m(a)) < 2m-1$ and c_2 a chain having no factor of odd length. By proposition 6.5, $c_2 = \varphi(c') \cup A$ and therefore

$$\ell(\varphi^{m-1}(a)) < (2m-1) - 2 = 2(m-1) - 1$$

Contradiction.

Acknowledgements

Many improvements came out from fruitful discussions with Jean Berstel, François Bergeron, and Christine Duboc. I am also indebted to Eduardo Dubuc, who spent some frustrating nights in front of the SUN workstation from the "Groupe de Combinatoire de l'UQAM", in order to produce an efficient program to compute word chains.

Bibliography

1. **Berstel J., Brlek S.** *On the length of word chains,* Information Proc. Letters, 1987 (to appear).
2. **Brlek, S.**. *Enumeration of factors in the Thue-Morse word,* Actes du Colloque de Combinatoire et Informatique de l'Université de Montréal (April 27th May 2nd 1987), Discrete and Applied Mathematics (submitted).
3. **Diwan A.A.** *A new combinatorial complexity measure for languages,* Tata Institute, Bombay, India (1986).
4. **Knuth D.E.** *The Art of Computer Programming* , Vol.2, 2nd ed. (Addison-Wesley, Reading, MA, 1981).
5. **Lothaire M.** *Combinatorics on Words,* (Addison-Wesley, Reading, MA,1983).

Mailing address:

Département de Mathématiques et Informatique
Université du Québec à Montréal
C.P. 8888 Succ. A, Montréal
H3C 3P8

Courrier Electronique:

R36274@UQAM.BITNET