State complexities of transducers for bidirectional decoding of prefix codes

Laura Giambruno Sabrina Mantaci
Dipartimento di Matematica e Informatica, Università di Palermo
via Archirafi, 34 - 90123 Palermo - ITALY
email: lgiambr,sabrina@math.unipa.it

Abstract

In a previous article [6] a bi-deterministic transducer is defined for the bidirectional decoding of words encoded using prefix codes by the method introduced by Girod. Also a coding method, inspired to the Girod's one, is introduced, and a transducer that allows both right-to-left and left-to-right decoding by this method is defined. It is proved that this transducer is also minimal.

Here we consider the number of states of such a transducer, related to some features of the considered prefix code X. We find some bounds of such number of states in relation with the different notions of "size" of X, such as its cardinality, its length, i.e the sum of the lengths of its elements, the size of the tree associated to the code, and the length of the longest word in X. In particular we give an exact formula for the number of states of transducers associated to maximal prefix codes, and we consider the special cases of maximal uniform codes and a class of codes, that we named string-codes, showing that they represent the extremal cases to this number of states in the case of maximal codes in terms of the length of the code and the size of its corresponding tree. This fact formalizes somehow the intuition that for maximal prefix codes the farthest a code is from being uniform the greatest the number of states is in the correspondent transducer, respect to the different sizes of the code.

1 Introduction

There are many reasons for decoding a message in both directions. The most important is connected to data integrity. In fact when we use a variable length code (VLC in short) for source compression (cf. [1], [8]), a single bit error in the transmission of the coded word may cause catastrophic consequences during decoding, since the wrongly decoded symbol generate loose of synchronization; in this way the error is propagated to the following symbols till the end of the file. In order to limit this error propagation, the compressed file is usually divided into records. If a single error occurs in a record, the decoder tries to

read the record from the end to the beginning. If there is just one error in the coding, it is possible to avoid the error propagation and isolate it. In order to do this we need codes that can be easily decoded in both directions. These are called bifix codes or reversible variable length codes (RVLC in short). Actually bifix codes are usually big and difficult to be constructed, whereas prefix codes over a k-letter alphabet, i.e. sets of words where no word is a prefix of another one, are very easy to be found, since they are in bijection with k-ary trees. A word encoded by a prefix code can be easily decoded without any delay, but it looses this property when we try to decode it from right to left. In 1999 Girod in [6] introduced a method that encodes words by using prefix codes, that allows to decode the encoded word both from left to right and from right to left with a delay of at most the length of the longest word in the code.

In [5] we defined a transducer for the bidirectional decoding of words encoded by the Girod's encoding. We also introduced a variant of the Girod's coding method, and we defined a transducer that allows both right-to-left and left-to-right decoding by this method. We proved that this transducer is deterministic, co-deterministic and minimal.

For sake of completeness, in this paper we recall Girod's encoding method with its variant and the construction of the transducer associated to the decoding operation on a given code X. Here we are mainly interested to find some bounds to the number of states of this transducer, depending on different notions of "size" of the prefix code X, such as the cardinality of X, the length of X, i.e. the sum of the lengths of its words, the number of nodes of the tree representing X, and the length of the longest word in X. The study of the state complexity is interesting for an algorithmic point of view.

In Section 2 we introduce some preliminary definitions and properties regarding codes and transducers, and the connection between these two notions. We moreover describe the method introduced by Girod for the bidirectional decoding of a prefix word and its variant (see [5]). We describe the construction of the transducer associated to its variant and we recall some of its properties.

In Section 3 we prove some general results on the state complexity of the transducer associated to prefix codes. In particular we give a general upper bound that holds for any prefix code. We discuss some experimental results on codes associated to isomorphic trees. We focus in particular our attention to maximal prefix codes, for which we find a precise formula giving the number of states of the associated transducer. This allows to prove, for maximal codes, an exponential lower bound on the length of the longest codeword. The formula gives, as particular cases, the one of uniform maximal codes and one of the so called string-codes. In particular, transducers associated to maximal uniform trees have a number of states that is linear with the length of the code, and polynomial in the size of the tree associated to the code, whereas the size of the transducer associated to string-codes is exponential in these notions of size. Finally we give the state complexity of the transducer associated to a uniform (non maximal) code with two words, and an upper bound in the general case of non maximal prefix code.

In Section 4 we give some conclusions and open problems.

2 Preliminaries

2.1 Codes and transducers

Let B and A be two alphabets, that we call respectively *source* alphabet and *channel* alphabet. Let $\gamma\colon B \to A^*$ be a map that associates to each element b in B a nonempty word on A. We extend this map to words over B by $\gamma(b_1 \dots b_n) = \gamma(b_1) \dots \gamma(b_n)$. We say that γ is an *encoding* if $\gamma(w) = \gamma(w')$ implies that $w = w'$. For each b in B, $\gamma(b)$ is said a *codeword* and the set of all codewords is said a *variable length code*, or simply a *code*. In what follows we denote by $x_i = \gamma(b_i)$ and by $X = \{x_1, \dots, x_m\}$ the code defined by γ. A set Y over A^* is said a *prefix set* (resp. *suffix set*) if no element of Y is a prefix (resp. a suffix) of another element of Y.

A set over A^* is called a *bifix set* if it is both a prefix and a suffix set. It can be easily proved that prefix, suffix and bifix sets are codes, called respectively *prefix codes*, *suffix codes* and *bifix codes*. A code is *maximal* if it is not contained in another code. A prefix code is maximal if and only if it is maximal as a prefix code. A *decoding* is the inverse operation than encoding i.e. the decoding of γ is the function γ^{-1} restricted to $\gamma(B^*)$.

Throughout this paper we consider codes over a binary alphabet, that is $A = \{0, 1\}$. For each word u we denote by $\tilde{u}$ the reverse of u. For $X = \{x_1, x_2, \dots, x_n\}$, we define by $\widetilde{X}$ the set $\widetilde{X} = \{\widetilde{x}_1, \widetilde{x}_2, \dots, \widetilde{x}_n\}$.

A finite transducer T uses an input alphabet A and an output alphabet B. It consists of a quadruple $T = (Q, I, F, E)$ where Q is a finite set whose elements are called *states*, I and F are two distinguished subsets of Q called the sets of *initial and terminal states*, and E is a set of elements called *edges* which are quadruples (p, u, v, q) where p and q are states, u is a word over A and v is a word over B. We call u the *input label* and v the *output label*. An edge is commonly denoted by $p \xrightarrow{u|v} q$. Two edges $p \xrightarrow{u_1|v_1} q$ and $r \xrightarrow{u_2|v_2} s$ are *consecutive* if $q = r$. A *path* in a transducer is a sequence of consecutive edges. The *label of the path* is obtained by concatenating separately the input and the output labels. We denote it by a pair with first element the input alphabet and second element the output alphabet. A transducer T defines a binary relation between words on the two alphabets as follows: a pair (u, v) is in the relation if it is the label of a successful path. This is called the *relation realized by* T. A transducer is called a *literal transducer* if each input label is a single letter.called *deterministic* (resp. *codeterministic*) if for each state p and for each input letter a there is at most one edge starting at (resp. ending at) p with input letter a.

We can represent encoding and decoding using transducers. An encoding γ can be represented by a one-state literal transducer with loops on the state with labels $(b, \gamma(b))$, for each b in B. Transducers for decoding are more interesting. In case of decoding, A represents the channel alphabet and B the source

alphabet. An interesting result is that for any encoding there exists a literal unambiguous transducer which realizes the associated decoding (see [1], [7]).

A *sequential transducer* over A, B is a triple $T = (Q, i, F)$ together with a partial function $Q \times A \longrightarrow B^* \times Q$ which breaks up into a *next state* function $Q \times A \longrightarrow Q$ and an output function $Q \times A \longrightarrow B^*$. In addition, the initial state $i \in Q$ has attached a word λ called the *initial prefix* and F is partial function $F : Q \longrightarrow B^*$ called the *terminal function*. Thus, an additional prefix and additional suffix can be attached to all the outputs. By definition, a sequential transducer is deterministic. There is a unique *minimal sequential transducer* equivalent to a given one i.e. with the minimal number of states among the sequential transducers realizing the same relation as it (cf. [7]).

2.2 Girod's method and transducers

It is well known that a prefix code can be decoded without delay in a left-to-right parsing while it can not be as easily decoded from right to left. In this section we describe a coding method, due to Girod (cf. [6]), where, given a finite prefix code X, any sequence of codewords in X is transformed in a bitstring that can be decoded in both directions, with a delay of the length of the longest string in the code.

Such a method is based on a well-known property of the binary sum. The binary sum operation $\oplus$ is a binary operation on $\{0, 1\}$ that returns a bit in this way: for a, b either both 1 or both 0, $a \oplus b$ returns 0 and in the other cases it returns 1. We will use the following property of $\oplus$: if $c = a \oplus b$ then $b = a \oplus c$ and $a = b \oplus c$.

Let $X = \{x_1, \ldots, x_m\}$ be a finite prefix code defined by an encoding γ over an alphabet $B = \{b_1, \ldots, b_m\}$. Consider a word $w = b_{i_1} \ldots b_{i_k}$ in B^* and its encoding $y = \gamma(b) = x_{i_1} \ldots x_{i_k}$ where x_{i_j}'s are codewords in X. By concatenating the reverse of each codeword x_{i_j}, we obtain the word $y' = \tilde{x}_{i_1} \ldots \tilde{x}_{i_k}$. Let $z = y \oplus y'$. The idea would be to decode y from z using the relation $y = z \oplus y'$. Anyway we cannot apply this idea since we should know y' in order to decode y. However we know that the elements in y' are strictly related to those in y. If we lightly modify y and y' we obtain the solution given by Girod.

Let us denote by L the length of the longest codeword in $\{x_{i_1}, \ldots, x_{i_k}\}$ and let us append the word 0^L to y as a suffix and to y' as a prefix. Then consider the words $x = y0^L$, $x' = 0^L y'$ and $z = x \oplus x'$ and define the encoding δ from B^* to A^* such that, for any $w = b_{i_1} \cdots b_{i_k} \in B^*$, $\delta(w) = z$, where z is defined as before.

Since the first L bits of x' are 0's, then the first L bits of z are equal to the first L bits of x. By the definition of L, those L bits contain as prefix at least the first codeword x_{i_1} in y. We concatenate its reverse $\tilde{x}_{i_1}$ to x'. In this way x' has again L unread symbols, that can be summed to the next L symbols of z. As before, this sum contains as prefix at least the second codeword x_{i_2}. Its reverse can be again concatenated to x' and have again L unread bits in x'. By proceeding in this way we obtain the left-to-right decoding of z.

Similarly we can decode z from right to left: in this case we invert the roles of x and x' and apply the operation $\oplus$ to z and to bits of x from right to left in order to obtain new bits for x'.

In [5], we remark that by using the properties of $\oplus$ the method can be analogously applied when any word of length L is used in the place of 0^L. We choose to use among the words of maximal length in X, the one that is minimal in lexicographic order (given a code, it is univocally determined). We refer to it as the *Girod's generalized method*.

We describe (see [5]) how to construct a transducer for the generalization of Girod's left-to-right decoding. For the description of the transducer for the classic Girod's left-to-right decoding see [5].

Let $X = \{x_1, \ldots, x_m\}$ be a finite prefix code defined by an encoding γ over an alphabet $B = \{b_1, \ldots, b_m\}$. Let L be the length of the longest word in X and let x_L be the smallest word in the lexicographic order among the words in X of length L. For any sequence y of codewords in X we consider the encoding δ_{x_L} as defined by the generalization of Girod's method. In order to simplify the notation we use δ_L instead of δ_{x_L}. The transducer $T = (Q, i, F, E)$ for the left-to-right decoding of δ_L is defined as follows.

The states in Q are pairs of words (u, v) such that:

- u is a proper prefix of a word in X;
- v is a suffix of a word in $\tilde{x}_L \widetilde{X}^*$ of length $L - |u|$;

The unique initial and final state i is $(\epsilon, \tilde{x}_L)$.

The edges in E are defined as follows:

1. $((u, av), c, \epsilon, (ud, v))$, with $a \oplus c = d$, if $ud \notin X$ and ud is a prefix of a word in X
2. $((u, av), c, b_i, (\epsilon, vd\tilde{u}))$, with $a \oplus c = d$, if $ud = x_i \in X$

In all remaining cases the transitions are not defined.

In Figure 1 we show the transducer T for the decoding of $X = \{11, 011\}$.

In [5] it is proved the following:

Theorem 2.1 *The transducer $\mathcal{T}$ realizes the function φ defined by $\varphi(z) = \delta_L^{-1}(z)b_L$, where δ_L^{-1} is the decoding of δ_L from left to right and b_L is the word $\gamma^{-1}(x_L)$. Moreover this transducer is deterministic, co-deterministic and minimal.*

In a similar way we can define a transducer for the right to left encoding. In [5] we prove that:

Theorem 2.2 *The transducers for the left-to-right and for the right-to-left decoding are isomorphic.*

This means that we can use the same transducer for decoding a word in both directions.

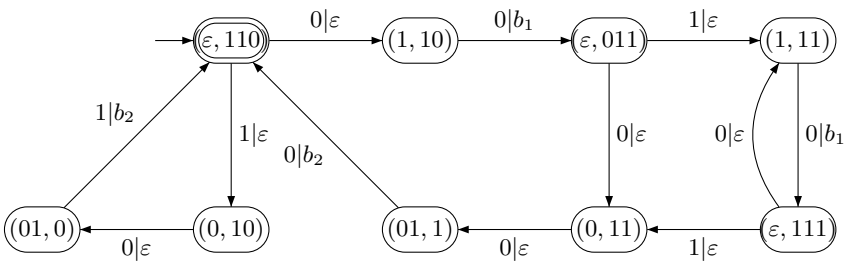

Figure 1: The transducer T for the left-to-right decoding by the Girod's generalized method of $X = \{11, 011\}$ over $B = \{b_1, b_2\}$

3 Bounds for the number of states

In this section we are interested in counting the number of states of a transducer associated to a given prefix code X, depending on some of its features. We consider the dependence of this number on the different notions of "size" of X, such as the number of codewords in X, the sum of their lengths, the length of the longest word, and the dimension of the tree representing X. We first consider some general bounds, then we consider maximal prefix codes for which we have an exact formula. We consider as particular cases uniform codes and "string-codes", representing respectively a good and a bad case. For uniform maximal codes we show a polynomial dependence of the number of states to the sum of the lengths of the codewords and to the size of the associated tree, whereas we have an exponential dependence on these sizes in the case of "string-codes".

Given a prefix code $X = \{x_1, x_2, \ldots, x_m\}$ we can measure the *size* of X in different ways: $|X|$, the *cardinality* X; $\|X\| = \sum_{x \in X} |x|$, the *length* of X; $|T_X|$, the size of the binary tree T_X naturally associated to X, i.e. the number of its nodes; $L = \max_{x \in X} |x|$, the length of the longest word in X.

For X prefix code, let $\mathcal{T}_X = (Q, i, F, E)$ be the transducer for the bidirectional decoding by the Girod's generalized method, as defined before, with $(\varepsilon, \widetilde{x}_L)$ as initial state. We are interested to find a bound to $|Q|$.

The following theorem gives a general upper bound for the number of states of the transducer associated to any prefix code.

Theorem 3.1 *If X is a prefix code then $|Q| \leq L\ 2^L$.*

The following Lemma shows that the number of states of the transducer grows when adding words to the prefix code:

Lemma 3.2 *Let $Y \subseteq X$ be prefix codes such that the length of the longest word in X is the length of the longest word in Y. Then $\mathcal{T}_Y$ is contained in $\mathcal{T}_X$ and the number of states of $\mathcal{T}_Y$ is strictly less than the number of states of $\mathcal{T}_X$.*

Given two binary trees T_1 and T_2, we say that they are isomorphic if T_2 can be obtained from T_1 by choosing some of its nodes and, for each of them switching the right and the left subtree. We say that two prefix codes are isomorphic if the associated trees are isomorphic. We have noticed, by experimental results, that, if X_1 and X_2 are two isomorphic prefix codes then the corresponding transducers have the same number of states. We conjecture that this is a general property (that we have proved, in the following, for some particular cases) and, in particular, we guess that the corresponding transducers are isomorphic as unlabeled graphs.

3.1 Maximal prefix codes

We consider now maximal prefix codes, i.e. codes where no word can be added in order to still have a prefix code. It is well known that maximal prefix codes are represented by complete trees, i.e. trees where each node has two or zero subtrees. For maximal prefix codes we get an exact formula that computes the number of states of the associated transducer. As a consequence, we get an exponential lower bound in L for this number. This depends on the well known fact (see [2]) that any word w in A^* is in $X^*Pref(X)$ that is, it can be written as concatenation of codewords and a prefix of a codeword. Consequently $\tilde{w}$ is in $Suff(\tilde{X})\tilde{X}^*$.

Lemma 3.3 *If X is a maximal prefix code then, for each pair of words (u, v) with $u = \epsilon$ or u proper prefix of X and $v \in A^{L-|u|}$, we have that $(u, v) \in Q$.*

Given a prefix set X, let us denote by $Pref_i(X)$ the set of proper prefixes of elements in X of length i and let us denote by $Level_i(X)$ the set of nodes in T_X at level i. As a consequence of Lemma 3.3 we have that the number of states for the transducer associated to a maximal prefix code is given by the following:

Theorem 3.4 *If X is a maximal prefix code then,*

$$|Q| = \sum_{i=0}^{L-1} |Pref_i(X)| \cdot 2^{L-i} = \sum_{i=0}^{L-1} |Level_i(X)| \cdot 2^{L-i}.$$

By Theorem 3.4 we can deduce an exponential lower bound in L for maximal prefix codes:

Corollary 3.5 *If X is a maximal prefix code then $|Q| \geq 2^L$.*

Following Theorem 3.4, we identify two classes of maximal codes that represent respectively the best and the worst case for the state complexity for maximal prefix codes: the uniform codes and the string-codes.

We say that a prefix code X is *uniform* if all the words in X have the same length L. A maximal uniform code whose words have length L contain all the words of this length, i.e. $X = A^L$. Then $|X| = 2^L$, $\|X\| = L\ 2^L$ and

$|T_X| = 2^{L+1} - 1$. Since, for each i, the number of nodes at level i in T_X is 2^i we get:

Theorem 3.6 *If X is a uniform maximal code then $|Q| = L2^L$.*

Corollary 3.7 *The bound given by Theorem 3.1 is tight for uniform maximal codes.*

The state complexity of the transducer associated to a maximal prefix code, computed in terms of the different sizes is given by the following:

Corollary 3.8 *Let X be a uniform maximal prefix code. The number of states of $\mathcal{T}_X$ is equal to*

- $\|X\|$
- $L2^L = |X|2^{|X|}$
- $|T_X| \log(|T_X| + 1) - 1$*;*

This means that for maximal uniform codes we have a linear dependence between the size of the transducer associated to X and the length of X. Moreover there is a polynomial dependence on the size of the corresponding tree.

Let u be a word in $A = \{0, 1\}$. We define X_u the *string-code* of u as $X_u = \{u\} \cup \{va \mid v\bar{a} \in Pref(u)\}$, where $Pref(u)$ is the set of prefixes of u. In this case L is the length of u. If X is a string-code, then $\|X\| = L(L+1)/2 + L$, $|X| = L + 1$ and $|T_X| = 2L$. Since, for each i, the number of nodes at level i in T_X is 1 we get:

Theorem 3.9 *If X_u is a string-code then $|Q| = 2^{L+1} - 2$.*

This means that $|Q| = \mathcal{O}(2^{\sqrt{\|X_u\|}})$, and $|Q| = \mathcal{O}(2^{|T_{X_u}|})$ and $|Q| = \mathcal{O}(2^{|X|})$. Thus these codes seem to have the worst behavior in terms of the number of states in relation with the different definitions of size of the code. As consequence of Theorem 3.9 we get that string-codes associated to words of the same lengths, that is isomorphic string codes, have all the same number of states in the associated transducer.

The theorems proved in this section formalizes somehow the intuition that the farthest a code is from being uniform and maximal, the greatest the number of states is in the correspondent transducer depending either on $|T_X|$ or on $\|X\|$.

If we take into account that isomorphism between two trees keeps the property of completeness of the tree and the number of nodes at a certain level in the tree, by Theorem 3.3, we have that:

Corollary 3.10 *Transducer associated to isomorphic maximal prefix codes have the same number of states.*

3.2 Uniform codes

Theorem 3.6 proved above give a value of the size of $\mathcal{T}_X$ when X is uniform and maximal

Let us consider now X a uniform non-maximal prefix code. For uniform codes of two words we have a precise result for the state complexity:

Theorem 3.11 *Let $X = \{x_1, x_2\}$ be a uniform code and let u be the longest common prefix between x_1 and x_2. Then:*

$$|Q| = \begin{cases} |T_X| - 3|u| + 2L - 3 & \text{if } |u| < L/2 \\ |T_X| - |u| + L - 2 & \text{if } |u| \geq L/2 \end{cases}$$

In general, we have the following proposition stating an upper bound to the state complexity of $\mathcal{O}(|X||T_X|)$ for non maximal prefix codes:

Proposition 3.12 *If X is a non maximal uniform code then $|Q| \leq |X||T_X| - |X|^2$. This bound is tight for codes of two words beginning with different letters.*

The tightness of Proposition 3.12 follows by Theorem 3.11.

4 Conclusions and open problems

We are trying to give a bound to the growth of the number of states of a transducer associated to a prefix code when we add a new word to the code, depending on how long is the common prefix between the new word and the words already in the code.

Our last will is to find upper and lower bounds for general prefix codes taking into account also the size of the tree representing the code, that gives information on how long common prefixes between pairs of words in X are.

It would be also interesting to do an average study of the number of states for different distributions on prefix codes.

References

[1] M-P Béal, J. Berstel, B. H. Marcus, D. Perrin, C. Reutenauer and P. H. Siegel. Variable-length codes and finite automata. *In I. Woungang (ed), Selected Topics in Information and Coding Theory, World Scientific.* To appear.

[2] J. Berstel and D. Perrin. *Theory of Codes.* Academic Press, 1985.

[3] J. A. Brzozowski. Canonical regular expressions and minimal state graphs for definite events. *In J. Fox, editor, Proc. of the Sym. on Mathematical Theory of Automata*, volume 12 of MRI Symposia Series, pages 529-561, NY, 1963. Polytechnic Press of the Polytechnic Institute of Brooklyn.

[4] A. S. Fraenkel and S. T. Klein. Bidirectional Huffman Coding, *The Computer Journal*, 33:296307.(1990)

[5] L. Giambruno and S. Mantaci. Transducers for the bidirectional decoding of prefix codes, *Theoretical Computer Science*, 411:17851792.(2010)

[6] B. Girod. Bidirectionally decodable streams of prefix code words. *IEEE Communications Letters.*, 3(8):245–247, August 1999.

[7] M. Lothaire. *Applied combinatorics on words*, Vol 104 of Encyclopedia of mathematics and its applications. Cambridge University Press, 2005.

[8] D. Salomon. *Variable-length codes for data compression.* Springer, 2007.