State complexities of transducers for bidirectional decoding of prefix codes

Laura Giambruno Sabrina Mantaci

Dipartimento di Matematica ed Applicazioni, Università di Palermo
via Archirafi, 34 - 90123 Palermo - ITALY
email: lgiambr,sabrina@math.unipa.it

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 introduce a variant of the Girod's coding method, and we define a transducer that allows both right-to-left and left-to-right decoding by this method. We prove 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.

We recall now some classical notion that we use in the paper. Let B and A be the *source* and the *channel* alphabets. Let $\gamma\colon B \to A^+$ be a map that we extend 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 code* (resp. *suffix code*) if no element of Y is a prefix (resp. a suffix) of another element of Y. A set over A^* is called a *bifix code* if it is both a prefix and a suffix set. A *decoding* of γ is the inverse operation than encoding i.e. the function γ^{-1} restricted to $\gamma(B^*)$. A code is *maximal* if it is not contained in another code. Throughout this paper we consider prefix codes over a binary alphabet $A = \{0, 1\}$. For each word u we denote by $\tilde{u}$ the reverse of u. For $X = \{x_1, x_2, \ldots, x_n\}$, we define by $\widetilde{X}$ the set $\widetilde{X} = \{\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_n\}$.

A finite transducer $\mathcal{T}$ on the input alphabet A and the output alphabet B is a quadruple $\mathcal{T} = (Q, I, F, E)$ where Q is a finite set of *states*, I and F are two subsets of Q called the sets of *initial and terminal states*, and E is a set of *edges* defined as (p, u, v, q) where $p, q \in Q$, $u \in A^*$ is the *input label* and $v \in B^*$ is the *output label.* Two edges (p, u_1, v_1, q) and (r, u_1, v_1, 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, and is denoted by a pair (u, v) with $u \in A^*$ and $v \in B^*$. A path is *successful* if it starts in an initial state and ends in a terminal state. We say that a pair (u, v) is in the *relation realized by* $\mathcal{T}$ if it is the label of a successful path. A transducer is called a *literal* if each input label is a single letter. A literal transducer is 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. A transducer is sequential if it is literal, deterministic, has a unique initial state i and can have an input label associated to i and output labels associated to final states. There is a unique *minimal sequential transducer* equivalent to a given one (cf. [7]).

The binary sum is the operation on $\{0, 1\}$ defined by: $a \oplus b = 0$ if $a = b$ and 1 otherwise. Notice that 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 words in X. Consider also the word $y' = \tilde{x}_{i_1} \ldots \tilde{x}_{i_k}$. Let L be the length of the longest word in $\{x_{i_1}, \ldots, x_{i_k}\}$. Consider the words $x = y0^L$, $x' = 0^L y'$ and $z = x \oplus x'$. The *Girod's encoding* $\delta : B^* \to A^*$ is the application, $\delta(w) = z$, where $w = b_{i_1} \cdots b_{i_k}$ and z is defined as above (see [6], [8]).

The left to right (right to left, resp.) decoding of z is allowed by the presence of 0^L in the beginning of x' (in the end of x, resp.). Starting from this known prefix of x' we reconstruct iteratively x and x' as follows. 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 the prefix 0^L of 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'.

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.*

Let X be a finite prefix code and let x_L be the smallest word in the lexicographic order among the words in X of maximal length L. For any sequence y of words in X consider the encoding δ_L as defined by the Girod's generalized method. The transducer $\mathcal{T} = (Q, i, F, E)$ for the left-to-right decoding of δ_L is defined by: the states in Q are pairs of words (u, v) such that 1) u is a proper prefix of a word in X and 2) 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:

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

Theorem 0.1 *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 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.

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 number of nodes of the binary tree T_X naturally associated to X; $L = \max_{x \in X} |x|$, the length of the longest word in X. For X prefix code, let $\mathcal{T}_X = (Q, i, F, E)$ be the transducer, as defined before, with $(\varepsilon, \widetilde{x}_L)$ as initial state. We are interested to find a bound to $|Q|$.

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 length L, i.e. $X = A^L$. Then $|X| = 2^L$, $\|X\| = L\ 2^L$ and $|T_X| = 2^{L+1} - 1$. We have the following:

Theorem 0.2 *If X is a prefix code then the number of states of $\mathcal{T}_X$ is less than or equal to $L\ 2^L$. If X is a uniform maximal code, this bound is tight.*

This means that, for uniform codes, $|Q| = \|X\|$, $|Q| \leq |T_X| \log |T_X|$ and $|Q| = |X| \log(|X|)$. Moreover Theorem 0.2 gives an exponential upper bound in L.

For maximal prefix codes we get an exponential lower bound in L. This depends on the well known fact (see [2]) that any word w in A^* is in $X^*Pref(X)$ (and consequently $\tilde{w}$ is in $Suff(\tilde{X})\tilde{X}^*$). Using this fact and the construction of the transducer we get the following lower bound:

Theorem 0.3 *If X is a maximal prefix code then, $|Q| \geq 2^L$.*

From this theorem we expect that the farthest a code is from being uniform, the greatest the number of states is in the correspondent transducer.

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

Lemma 0.4 *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 to the one 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 have noticed, by experimental results, that, if X_1 and X_2 are two prefix codes corresponding to isomorphic trees then the corresponding transducers have the same number of states. We conjecture that the corresponding transducers are, in particular, isomorphic as unlabeled graphs.

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 0.5 *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:*

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

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

Proposition 0.6 *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 0.6 follows by Theorem 0.5.

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. We denote by L the length of u. In this case $\|X\| = L(L+1)/2 + L$, $|X| = L+1$ and $|T_X| = 2L$. These codes are maximal, then by Theorem 0.3 we have that $|Q| \geq 2^L$. Moreover by experimental results we have noticed that for all these codes $|Q| = 2^{L+1} - 2$. This is probably a general property that we will try to prove. This would mean that $|Q| = \mathcal{O}(2^{\sqrt{\|X\|}})$, and $|Q| = \mathcal{O}(2^{|T_X|})$ 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. Moreover, if such a formula is true, then we would get that string-codes defined by words of the same lengths have all the same number of states in the associated transducer.

We are trying to give a bound on 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 general 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.