Theoretical Computer Science 411 (2010) 1785–1792

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Transducers for the bidirectional decoding of prefix codes

Laura Giambruno *, Sabrina Mantaci

Dipartimento di Matematica ed Applicazioni, Università di Palermo, Italy

ARTICLE INFO

Article history:
Received 28 December 2009
Accepted 29 January 2010
Communicated by D. Perrin

Keywords:
Prefix codes
Girod's encoding
Transducers

ABSTRACT

We construct a transducer for the bidirectional decoding of words encoded by the method introduced by Girod (1999) in [5] and we prove that it is bideterministic and that it can be used both for the left-to-right and the right-to-left decoding.

We also give a similar construction for a transducer that decodes in both directions words encoded by a generalization of Girod's encoding method. We prove that it has the same properties as those of the previous transducer. In addition we show that it has a single initial/final state and that it is minimal.

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 (see [1,7]), a single bit error in the transmission of the coded word may cause catastrophic consequences during decoding, since the wrongly decoded symbol creates a loss 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 (RVLCs in short).

Recall that a *prefix code* (respectively *suffix code*) is a set of words such that none of its elements is the prefix (respectively suffix) of another one, and a *bifix code* is a code that is at the same time a prefix and a suffix. Prefix codes are widely used for compression (see [2]), since the Huffman code tree is usually mapped to a prefix code. Recall that the Huffman algorithm finds a minimum redundancy code for a given distribution of symbol probabilities.

It is well known that prefix codes can be easily decoded from left to right, but it is not trivial to decode them from right to left. This can be done with great delay and a great computational effort. For this reason, RVLCs assume a very central role in Information Theory, since they are the ones that can be decoded in both directions. In particular, fixed length codes are bifix codes. It is important to construct RVLCs that are as short as possible, in order to guarantee compression. Unfortunately RVLCs are not very efficient in terms of data compression because there is no efficient and simple algorithm for their construction. Fraenkel and Klein [4] show how to construct Huffman bifix codes from prefix codes using a very complex algorithm.

For these reasons it is useful to have a method that allows one to decode prefix codes from both sides. Such a method was introduced in 1999 by Berndt Girod in [5].

In short, this method uses an important property of the binary sum $\oplus$, that is that if $z = x \oplus y$, then $x = z \oplus y$ and $y = x \oplus z$. Let C be a binary prefix code and let L be the maximal length of the words of C. Consider the concatenation $x_1x_2 \cdots x_n$ of codewords and let $w = x_1x_2 \cdots x_n 0^L \oplus 0^L \tilde{x}_1\tilde{x}_2 \cdots \tilde{x}_n$, where $\tilde{x}$ denotes the reverse of the word x. It can be proved that w can be decoded in both directions. Usually for each code it is possible to construct a transducer for its decoding.

* Corresponding author. Tel.: +39 09123891055; fax: +39 091238 91024.
E-mail addresses: lgiambr@math.unipa.it (L. Giambruno), sabrina@math.unipa.it (S. Mantaci).

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.01.033

The aim of this paper is to construct the transducer for decoding words that have been encoded with Girod's method and to show some of its interesting properties that usually do not hold for transducers related to other coding methods. For instance, the transducer happens to be bideterministic (i.e. deterministic and co-deterministic) and the same transducer can be used for the left-to-right and for the right-to-left decoding.

In Section 2, we introduce the preliminary notions and definitions regarding codes and transducers, and the connection between the two notions.

In Section 3, we describe the method introduced by Girod for the bidirectional decoding of a prefix word and we provide an example with encoding and decoding.

In Section 4, we give the algorithm for the construction of the transducer that realizes the decoding of bitstreams obtained by Girod's encoding method. We prove that the transducer constructed by the algorithm is deterministic and co-deterministic. Another important property is that the transducer for the right-to-left decoding is isomorphic to the one for left-to-right decoding. This allows one to use the same transducer for decoding in both directions.

In Section 5, we discuss a generalization of Girod's method and provide its representation by transducers. We prove that the transducer has the same properties as those of the transducer constructed in Section 4. We moreover prove that the transducer has a unique initial/final state and that it is minimal.

2. Preliminaries

Let B and A be two alphabets, which we call respectively the *source* alphabet and *channel* alphabet. Let $\gamma: B \to A^*$ 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 to be a *prefix set* (respectively a *suffix set*) if no element of Y is a prefix (respectively a suffix) of another element of Y. A set over A^* is called a *bifix set* if it is both a prefix set 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 *decoding* is the inverse operation to 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 $\widetilde{X} = \{\widetilde{x}_1, \widetilde{x}_2, \dots, \widetilde{x}_n\}$.

For each word u in A^* and for each $k \leq |u|$, we denote by $pref_k(u)$ the prefix of u of length k and by $pref_{-k}(u)$ the prefix of u of length $|u| - k$.

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 path is *successful* if it starts in an initial state and ends in a terminal state. 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. A literal transducer is called *unambiguous* if for each pair of states p, q and for each word w in A^* there is at most one path starting at p and ending at q with input label w. A literal transducer is called *deterministic* (respectively *co-deterministic*) if for each state p and for each input letter a there is at most one edge starting at (respectively 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 [1,6]).

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 an 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 (see [6]).

3. Girod's method

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 (see [5]), 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.

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 the 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 $w = b_{i_1} \ldots b_{i_k} \in B^*$, $\delta(w) = z$, 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.

This decoding procedure is better explained by the following example. Let $B = \{b_1, b_2\}$ and let us encode b_1 with 11 and b_2 with 011. The set $X = \{11, 011\}$ is a finite prefix code. The word $w = b_1 b_2 b_2 b_1$ in B^* is encoded in $y = 1101101111$ and the concatenation of the reverse of each codeword produces the word $y' = 1111011011$. The length of the longest word in X is $L = 3$. We consider the string $z = x \oplus x'$, where $x = y0^L$ and $x' = 0^L y'$, and we obtain

$$\begin{aligned} x &= 1101101111000 \\ x' &= 0001111011011 \\ z &= 1100010100011 \end{aligned}$$

We show in this example how Girod's left-to-right decoding of z works.

1. We know that the first $L = 3$ bits of x' are 000. This allows us to obtain the first 3 bits of x by the operation $110 \oplus 000 = 110$. We see that $11 = x_1$, so we can decode 110 as $x_1 1$, where 1 is a remainder for the next step. Now we append $\tilde{x}_1$ to the end of the known symbols of x'.

$$\begin{aligned} z &= 1100010100011 \\ x' &= 000\mathbf{11} \\ x &= 110 \end{aligned}$$

2. We sum the new bits 11 of x' with the corresponding bits in z in order to obtain more bits for x. We append these bits to the remaining bits of the previous step obtaining a word of length 3 that will contain at least a codeword. In particular, we obtain in x the bits $011 = x_2$. We append $\tilde{x}_2$ to the known symbols in x'.

$$\begin{aligned} z &= 1100010100011 \\ x' &= 00011\mathbf{110} \\ x &= 11011 \end{aligned}$$

By repeating the same procedure, we have the following steps:

$$\begin{aligned} z &= 1100010100011 \\ x' &= 0001111\mathbf{0110} \\ x &= 11011011 \end{aligned}$$

$$\begin{aligned} z &= 1100010100011 \\ x' &= 000111101\mathbf{11} \\ x &= 1101101111 \end{aligned}$$

$$\begin{aligned} z &= 1100010100011 \\ x' &= 0001000100110 \\ x &= 110110111\mathbf{000} \end{aligned}$$

Let us note that in each step we have $L = 3$ bits in x to be decoded and, by definition of L, this fact assures that these bits will contain at least a codeword. Notice moreover that in the last step we obtain the last three 0 bits that indicates the successful ending of the decoding. If in the last step we do not have a string of zeros then some error has occurred.

Remark 1. 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'.

Remark 2. We can apply a generalization of Girod's coding method for alphabet A of any finite cardinality for which there exists a binary function $f : A \times A \longrightarrow A$ that results in being bijective when restricted to a component.

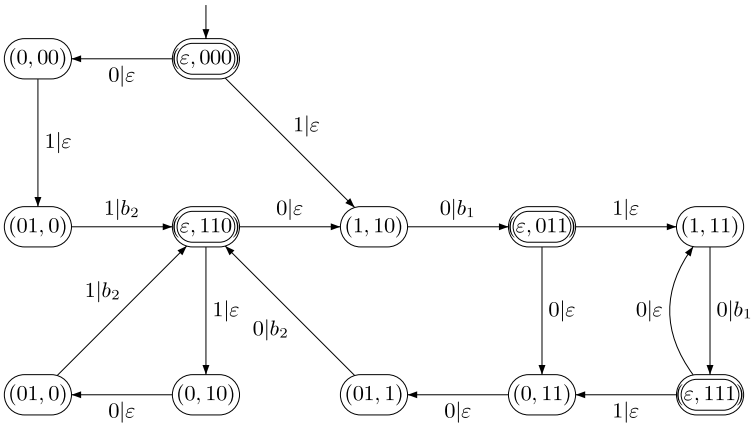

Fig. 1. The transducer T for the left-to-right decoding of $X = \{11, 011\}$ over $B = \{b_1, b_2\}$.

4. Transducers for decoding with Girod's method and main theorem

In this section we first describe an algorithm to construct a transducer for Girod's left-to-right decoding. We then give the construction of the transducer for the right-to-left decoding and we prove that the two transducers are isomorphic.

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. For any sequence y of codewords in X we consider the word z as defined by Girod's method. The transducer we are going to define is able to decode the encoded words z except their suffix of length L. In fact, note that the last L symbols of z correspond to the encoding of the suffix 0^L that is not part of the original word x.

The transducer $T = (Q, i, F, E)$ for the left-to-right decoding is defined as follows. The states in Q are pairs of words (u, v) such that

- u is either the empty word ϵ or a proper prefix of a word in X;
- v is a suffix of a word in $0^L\widetilde{X}^*$, of length $L - |u|$.

The initial state i is $(\epsilon, 0^L)$.
The terminal states in F are the accessible pairs (ϵ, v). The edges in E are defined as follows.

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

In all remaining cases the transitions are not defined.

In Fig. 1 we show an example of the transducer T for the decoding of words in X^*, where $X = \{11, 011\}$.

Remark 3. If there is in T a path from the initial state $(\epsilon, 0^L)$ to a state of the form (ϵ, u), with label $(z, b_{i_1} \ldots b_{i_k})$, then u is the suffix of length L of $0^L\tilde{x}_{i_1} \ldots \tilde{x}_{i_k}$.

By construction and by the properties of the binary sum we get the following.

Lemma 1. *For each state (ϵ, v) in T, there is a path starting at (ϵ, v) and ending at a state (ϵ, u) with label (z, b_i) if and only if $z = x_i \oplus pref_{|x_i|}(v)$.*

Lemma 2. *There is a path from the initial state to a state (ϵ, u) with label $(z, b_{i_1} \ldots b_{i_k})$ if and only if $z = x_{i_1} \ldots x_{i_k} \oplus pref_{-L}(0^L\tilde{x}_{i_1} \ldots \tilde{x}_{i_k})$.*

Proof. Let $w = b_{i_1} \ldots b_{i_k}$. We prove the lemma by induction on $|w|$. For $|w| = 1$ the statement follows from Lemma 1.

Let (z, w) be the label of a path from the initial state to a state (ϵ, u), with $|w| = k$. Let us consider the subpath from the initial state to the last intermediate state of the form (ϵ, v).

By construction, such a path is labelled by $(z', b_{i_1} \ldots b_{i_{k-1}})$, where, by the inductive hypothesis, $z' = x_{i_1} \ldots x_{i_{k-1}} \oplus pref_{-L}(0^L\tilde{x}_{i_1} \ldots \tilde{x}_{i_{k-1}})$. The last part of the path from (ϵ, v) to (ϵ, u) is labelled by (z'', b_k), and, by Lemma 1, we have that $z'' = x_k \oplus pref_{|x_k|}(u)$.

By composing these two relations we obtain $z = z'z'' = x_{i_1} \ldots x_{i_{k-1}}x_{i_k} \oplus pref_{-L}(0^L\tilde{x}_{i_1} \ldots \tilde{x}_{i_{k-1}})pref_{|x_k|}(u)$. By Remark 3, u is the suffix of length L of $0^L\tilde{x}_{i_1} \ldots \tilde{x}_{i_{k-1}}$; then we have $z = x_{i_1} \ldots x_{i_{k-1}}x_{i_k} \oplus pref_{-(L-|x_{i_k}|)}(0^L\tilde{x}_{i_1} \ldots \tilde{x}_{i_{k-1}})$, and so $z = x_{i_1} \ldots x_{i_{k-1}}x_{i_k} \oplus pref_{-L}(0^L\tilde{x}_{i_1} \ldots \tilde{x}_{i_{k-1}}\tilde{x}_k)$, which is the thesis.

Conversely, let z be the encoding by δ of $w = b_{i_1} \ldots b_{i_k}$ except the last L symbols, i.e. $z = pref_{-L}(\delta(w)) = x_{i_1} \ldots x_{i_k} \oplus pref_{-L}(0^L\tilde{x}_{i_1} \ldots \tilde{x}_{i_{k-1}}\tilde{x}_k)$. Notice that the last L symbols of $\delta(w)$ would encode the word 0^L added to the end. We prove the statement by induction on $|w|$.

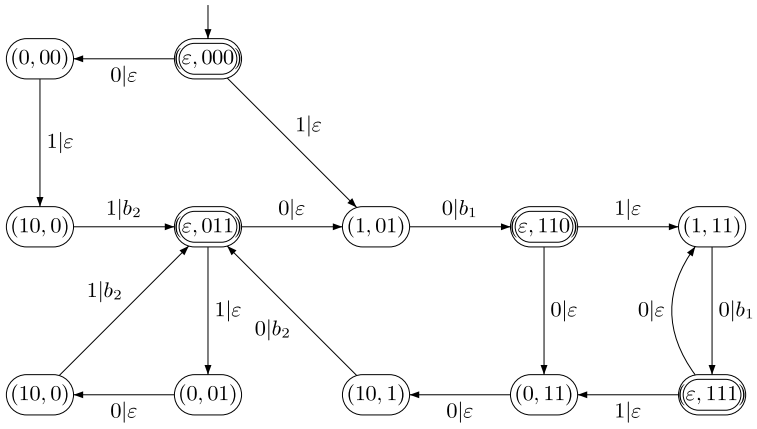

Fig. 2. The transducer T' for the right-to-left decoding of $X = \{11, 011\}$ over $B = \{b_1, b_2\}$.

If $|w| = 1$ then the statement follows from Lemma 1. Let $z = x_{i_1} \dots x_{i_k} \oplus pref_{-L}(0^L \tilde{x}_{i_1} \dots \tilde{x}_{i_{k-1}} \tilde{x}_k)$. We decompose z as $z'z''$, where z' is the prefix of z that encodes $b_{i_1} \dots b_{i_{k-1}}$; that is, $z' = x_{i_1} \dots x_{i_{k-1}} \oplus pref_{-L}(0^L \tilde{x}_{i_1} \dots \tilde{x}_{i_{k-1}})$. By induction, there exists a path from $(\epsilon, 0^L)$ to a state (ϵ, u) labelled by $(z', b_{i_1} \dots b_{i_{k-1}})$. By Remark 3, u is the suffix of $0^L \tilde{x}_{i_1} \dots \tilde{x}_{i_{k-1}}$ of length L, and we have that $z'' = x_{i_k} \oplus pref_{|x_{i_k}|}(u)$.

Then, by Lemma 1, there exists a path from (ϵ, u) to a state (ϵ, v) with label (z'', b_k), and this concludes the proof. □

By Lemma 2 we obtain the following.

Proposition 3. *The transducer T realizes the decoding δ^{-1} from left to right on the encoded words z, except their suffix of length L.*

In the following we prove that the transducer T is co-deterministic; that is, for each state p and for each letter $a \in \{0, 1\}$, there is at most one edge entering at p with label (a, b), with $b \in B \cup \{\epsilon\}$.

Theorem 4. *The transducer T is co-deterministic.*

Proof. Let us consider a state (u, v) with u nonempty word. By construction (cases 1 and 3), there exists at most one state predecessor of (u, v) with a fixed label.

Consider now a state of the form (ϵ, v), and suppose that there exist two different edges ending at (ϵ, v) with the same input label. Thus there exist two words w, w' and x_i, x_j in X such that $v = w\tilde{x}_i = w'\tilde{x}_j$ (case 2). Let $\tilde{x}_i$ be the shortest word between $\tilde{x}_i$ and $\tilde{x}_j$. The previous equality implies that $\tilde{x}_i$ is a suffix of $\tilde{x}_j$, which means that x_i is a prefix of x_j. This is a contradiction since X is a prefix code. □

Remark 4. Note that there are at most two edges ending at a given state and they have the same output label. In fact, if the state has as first component the empty word and if there exist two edges ending at it with two different output labels x_i and x_j, respectively, there exist two words w, w' such that $w\tilde{x}_i = w'\tilde{x}_j$, and that implies that $x_i = x_j$ since X is a prefix code. If a state has the first component different from the empty word then all its in-edges have the same output label, ε.

Let us now define a transducer for Girod's right-to-left decoding by $T' = (Q', i', F', E')$. The states in Q' are pairs of words (u, v) such that

- u is either the empty word ϵ or a proper suffix of a word in $\tilde{X}$;
- v is a prefix of a word in X^*0^L, of length $L - |u|$.

Let us note that, if u is a proper suffix of a word in $\tilde{X}$, then $\tilde{u}$ is a proper prefix of a word in X.
The initial state i' is $(\epsilon, 0^L)$.
The final states F' are the accessible pairs (ϵ, v).
The edges in E' are defined as follows.

1. $((u, wa), c, \epsilon, (du, w))$, with $a \oplus c = d$, if $au \notin \tilde{X}$ and au is a suffix of a word in $\tilde{X}$;
2. $((u, wa), c, b_i, (\epsilon, \tilde{u}dw))$, with $a \oplus c = d$, if $du = \tilde{x}_i \in \tilde{X}$.

As before, we can prove the following.

Proposition 5. *The transducer T' realizes δ^{-1} from right to left on the encoded words z, except their prefix of length L.*

In Fig. 2 we can see an example of transducer T' for the right-to-left decoding of $X = \{11, 011\}$. We can notice that such a transducer is isomorphic to the transducer of Fig. 1 that realizes the left-to-right decoding of $X = \{11, 011\}$. This is not a case, as the following proposition states.

Proposition 6. *The transducers T and T' are isomorphic.*

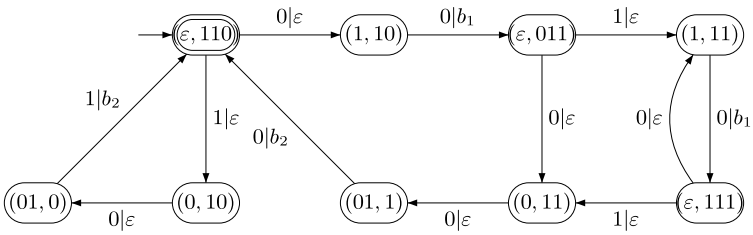

Fig. 3. The transducer S for the left-to-right decoding of $X = \{11, 011\}$ over $B = \{b_1, b_2\}$.

Proof. Let $\phi : Q \longrightarrow Q'$ such that $\phi((u, v)) = (\tilde{u}, \tilde{v})$.

The function ϕ is well defined since if u is a prefix of a word in X then $\tilde{u}$ is a suffix of a word in $\tilde{X}$, and if v is a suffix of a word in $0^L\tilde{X}^*$ of length $L - |u|$ then $\tilde{v}$ is a prefix of a word in X^*0^L of length $L - |u|$.

It is injective since if $\phi((u, v)) = \phi((t, r))$ then $(\tilde{u}, \tilde{v}) = (\tilde{t}, \tilde{r})$, which means that $(u, v) = (t, r)$.

It is surjective since if (u, v) is in Q' then u is either the empty word ϵ or a proper suffix of a word in $\tilde{X}$ and v is a prefix of a word in X^*0^L. Thus $\tilde{u}$ is either the empty word ϵ or a proper prefix of a word in X and $\tilde{v}$ is a prefix of a word in $0^L\tilde{X}^*$, and we have that $(\tilde{u}, \tilde{v})$ is a state in Q such that $\phi((\tilde{u}, \tilde{v})) = (u, v)$.

In order for there to be an isomorphism of transducers we have to prove that ϕ preserves the edges. In particular, we have to prove that if $((u, v), a, x, (u', v'))$ is an edge in T then $(\phi((u, v)), a, x, \phi((u', v')))$ is an edge in T'. We prove it for $a = 0$; the other case will be analogous.

If $ua \notin X$ and ua is a prefix of a word in X then in T there is the edge $((u, aw), 0, \epsilon, (ua, w))$. We have that $\phi((u, aw)) = (\tilde{u}, \tilde{w}a)$ and $\phi((ua, w)) = (a\tilde{u}, \tilde{w})$. Under these hypotheses, $a\tilde{u} \notin \tilde{X}$ and $a\tilde{u}$ is a suffix of a word in $\tilde{X}$. So we have in T' the edge $((\tilde{u}, \tilde{w}a), 0, \epsilon, (a\tilde{u}, \tilde{w}))$.

If $ua \in X$ then in T there is the edge $((u, aw), 0, ua, (\epsilon, wa\tilde{u}))$. We have that $\phi((u, aw)) = (\tilde{u}, \tilde{w}a)$ and $\phi((\epsilon, wa\tilde{u})) = (\epsilon, ua\tilde{w})$. Since $a\tilde{u} \in \tilde{X}$, we have in T' the edge $((\tilde{u}, \tilde{w}a), 0, ua, (\epsilon, ua\tilde{w}))$.

The proof is analogous for the edges with input label 1. □

By Proposition 6 it follows that T' is deterministic and co-deterministic.

5. A generalization of Girod's method: The transducer for decoding

In this section we consider a generalization of Girod's coding method. This generalization, as for Girod's classical method, also allows bidirectional decoding of prefix codes, but it has the advantage of allowing the construction of a transducer that has more interesting properties than the ones described in the previous section.

This generalization comes from the remark that if the coding keyword 0^L is substituted by any word w of length L (or longer than L), a "Girod-like" coding method works anyway, and also bidirectional decoding is possible, provided that the coding key w is known.

Our idea is to use as coding key the word w in the code, having length L and that is the least in lexicographic order. In what follows, for a given w, we denote such encoding as δ_w.

In this section we describe how to construct a transducer for the generalization of Girod's left-to-right decoding. Also in this case we describe the transducer for the right-to-left decoding and we prove that the two transducers are isomorphic.

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 $S = (Q, i, F, E)$ 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 $\tilde{x}_L\tilde{X}^*$ 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 Fig. 3 we show the transducer S for the decoding of $X = \{11, 011\}$. One can note that the transducer S is sequential by taking as the terminal function F defined to the only final state i as $F(i) = \epsilon$ and by taking as λ the empty word. Let us note that Lemmas 1 and 2 still hold for S.

Proposition 7. *The transducer S is deterministic and 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 $\gamma^{-1}(x_L)$.*

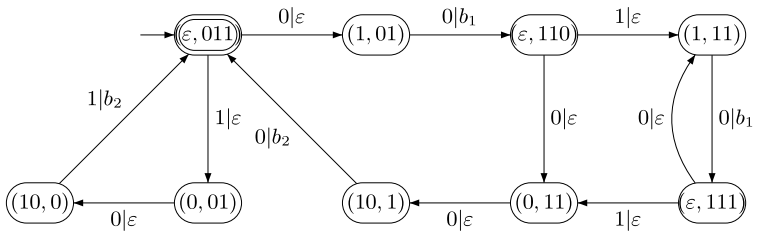

Fig. 4. The transducer S' for the right-to-left decoding of $X = \{11, 011\}$ over $B = \{b_1, b_2\}$.

Proof. The determinism of S follows by construction. In order to prove the second part of the proposition we have to prove that (z, w) is a pair in the relation realized by S if and only if $w = \varphi(z)$.

If $|w| = 1$ then, by construction and by Lemma 1, we have that $z \oplus \tilde{x}_L = x_L$; that is, z is the encoding of the empty word.

Let us consider $w \in B^*$ with $|w| \geq 2$, $w = b_{i_1} \dots b_{i_k}$. The pair (z, w) is a pair in the relation realized by S if and only if there is a path from the initial state $(\epsilon, \tilde{x}_L)$ to a state (ϵ, v) with label $(z', b_{i_1} \dots b_{i_{k-1}})$ and a path from (ϵ, v) to $(\epsilon, \tilde{x}_L)$ with label (z'', b_{i_k}) with $z = z'z''$ and $x_{i_k} = x_L$.

By Lemma 2, there is a path from the initial state $(\epsilon, \tilde{x}_L)$ to a state (ϵ, v) with label $(z', b_{i_1} \dots b_{i_{k-1}})$ if and only if $z' = x_{i_1} \dots x_{i_{k-1}} \oplus \mathit{pref}_{-L}(\tilde{x}_L \tilde{x}_{i_1} \dots \tilde{x}_{i_{k-1}})$. Moreover, by Lemma 1, there is a path from (ϵ, v) to $(\epsilon, \tilde{x}_L)$ with label (z'', b_{i_k}) if and only if $z'' \oplus v = x_L$.

By composing the two equalities, we get that the pair (z, w), with $|w| \geq 2$, is a pair in the relation realized by T if and only if $z = z'z'' = x_{i_1} \dots x_{i_{k-1}} x_L \oplus \tilde{x}_L \tilde{x}_{i_1} \dots \tilde{x}_{i_{k-1}}$, and the statement follows. □

The transducer is co-deterministic (see proof of Proposition 4) and the Remark 4 holds. Moreover we have the following.

Theorem 8. *The transducer S is minimal.*

Proof. Since the transducer – and so the input automaton – is bideterministic and it has a unique initial and final state then, by the Brzozowski minimization algorithm (see [3]), the input automaton is minimal. This implies that the transducer is also minimal, since the output is produced as soon as possible (i.e. the transducer is normalized in the sense of [6]). □

Let X be a prefix code and let us now define a transducer for Girod's right-to-left decoding of δ_L by $S' = (Q', i', F', E')$. The states in Q' are pairs of words (u, v) such that

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

Let us note that, if u is a proper suffix of a word in $\tilde{X}$, then $\tilde{u}$ is a proper prefix of a word in X.

The unique initial and final state i' is $(\epsilon, \tilde{x}_L)$. The edges in E' are defined as follows.

1. $((u, wa), c, \epsilon, (du, w))$, with $a \oplus c = d$, if $au \notin \tilde{X}$ and au is a suffix of a word in $\tilde{X}$;
2. $((u, wa), c, b_i, (\epsilon, \tilde{u}dw))$, with $a \oplus c = d$, if $du = \tilde{x}_i \in \tilde{X}$.

The following propositions hold. The proofs are analogues to those of Proposition 5, Proposition 7 and Proposition 6.

Proposition 9. *The transducer S' is deterministic and 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 right to left and b_L is the word $\gamma^{-1}(x_L)$.*

Proposition 10. *The transducers S and S' are isomorphic.*

By the last proposition it follows that S' is deterministic and co-deterministic.

In Fig. 4 we can see an example of transducer S' for the right-to-left decoding of $X = \{11, 011\}$. Such a transducer is isomorphic to the transducer of Fig. 3 that realizes the left-to-right decoding of $X = \{11, 011\}$ as stated in Proposition 10.

Remark 5. Let us denote by S^t the transducer obtained by S by reversing all the edges. Since S is co-deterministic and has a unique initial/final state, the transducer S^t is deterministic and realizes $b_L \delta_L^{-1}$, where δ_L^{-1} is the decoding of δ_L from right to left.

Notice that, despite both S^t and S' being useful to decode from right to left a bitstream obtained by Girod's generalized method, they are not equal, since they recognize different languages.

Acknowledgements

We are very grateful to Dominique Perrin for proposing the problem to us and for introducing us to Girod's encoding. We also thank him for the useful friendly discussions about this problem.

References

[1] M.-P. Béal, J. Berstel, B.H. Marcus, D. Perrin, C. Reutenauer, P.H. Siegel, Variable-length codes and finite automata, in: I. Woungang (Ed.), Selected Topics in Information and Coding Theory, World Scientific, 2010 (in press).
[2] J. Berstel, 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 (Ed.), Proc. of the Sym. on Mathematical Theory of Automata, in: MRI Symposia Series, vol. 12, Polytechnic Press of the Polytechnic Institute of Brooklyn, NY, 1963, pp. 529–561.
[4] A.S. Fraenkel, S.T. Klein, Bidirectional Huffman coding, The Computer Journal 33 (1990) 296–307.
[5] B. Girod, Bidirectionally decodable streams of prefix code words, IEEE Communications Letters 3 (8) (1999) 245–247.
[6] M. Lothaire, Applied combinatorics on words, in: Encyclopedia of Mathematics and its Applications, vol. 104, Cambridge University Press, 2005.
[7] D. Salomon, Variable-Length Codes for Data Compression, Springer, 2007.