STRUCTURE OF DECODERS FOR MULTIVALUED ENCODINGS*

Renato M. CAPOCELLI**
Department of Computer Science, Oregon State University, Corvallis, OR 97331, USA

Ugo VACCARO
Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84100 Salerno, Italy

Received 8 June 1984
Revised 31 March 1988

Multivalued encodings constitute an interesting generalization of ordinary encodings in that they allow each source symbol to be encoded by more than one codeword. In this paper the problem of decoding multivalued encodings is considered and three algorithms for constructing finite-state sequential decoders are provided.

1. Introduction

An encoding system is said to be multivalued if there may be two or more codewords corresponding to the same source symbol. In this paper the problem of existence and of construction of decoders for multivalued encodings is considered.

Multivalued encodings have been recently considered [2, 3, 8]. They seem to constitute an interesting generalization of ordinary codes. In particular, multivalued encodings appear very suitable for modeling transmission over noisy channels. As is well known, when a sequence of symbols is transmitted over a noisy channel, the output is not uniquely determined but can be any of a set of sequences, depending both on the transmitted sequence and the error pattern that has occurred. Notice that if the channel allows not only substitution errors but also deletion and insertion errors, the output sequences associated to an input sequence may have different lengths. Roughly speaking, the most general way to describe the behavior of a channel that suffers of insertion, deletion and substitution errors is to specify, for each input symbol, all the possible sequences that can occur at the output. This can be done by means of a multivalued encoding in which the set of codewords corresponding to a single source symbol represents the noisy version of the original encoding of that symbol. This approach, however, can be useful only if the set of sequences

* This work was supported in part by the Italian Ministry of Education, Project: "Progetto ed Analisi di Algoritmi".
** On leave from Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84100 Salerno, Italy.
associated with each source symbol is not too large. Generally speaking, one can prevent this situation by ignoring all sequences having small probability of occurrence.

Since their introduction, researchers have tried to characterize properties of multivalued encodings using various approaches. The situation is complicated, and made interesting, by the fact that for multivalued encodings unique decipherability is not equivalent to unique decomposability (i.e., a code message might be parsed in two different ways, both giving the same deciphering in terms of source symbols). A similar situation arises in the recently considered multiset decipherable codes, introduced by Lempel in [7], where every possible parsing of the message into codewords must yield the same multiset of codewords. Going back to multivalued encodings, we have that the nonequivalence between unique decomposability and unique decipherability implies directly that the extension to multivalued encodings of fundamental properties previously defined in the framework of ordinary encodings is not straightforward, neither does it appear possible to use the methods that have been successfully employed in the encoding case (see [5] for instance) to test whether a multivalued encoding possesses such properties. Nevertheless, several results have been obtained. Sato [8] gave a decision procedure to test whether a multivalued encoding has the property of being uniquely decipherable, Capocelli [2] characterized the property of decipherability with finite delay, Capocelli et al. [4] characterized the property of synchronizability. The above quoted papers left open the problem of the effective decoding of multivalued encodings. This is the problem we address in this paper. More specifically, we want to study the problem of the construction of decoders for multivalued encodings. Generally speaking, we consider a decoder to be a finite-state machine that, having as input any code message $\beta$, gives as output the source message that generated $\beta$, with an exception made for a finite terminal part of it. In this paper we provide a necessary and sufficient condition for a multivalued encoding to admit of decoders. Moreover, we give three algorithms for constructing such decoders. The algorithms produce decoders exhibiting various properties that may be desirable to have in different situations. It is worth pointing out that, for ordinary encodings, our algorithms reduce essentially to those given by Levenstein in [6].

2. Notations and definitions

Let $X$ be a finite nonempty set and let $X^+$ and $X^*$ be the free semigroup and the free monoid generated by $X$, respectively. We recall that the free semigroup $X^+$ denotes the set of all finite sequences of elements of $X$ and that $X^+=X^*-\{\lambda\}=\bigcup_{n=1}^{\infty} X^n$; where $\lambda$ and $X^n$ denote the empty word and the $n$th concatenation of $X$ with itself, respectively. We call the elements of $X$ code symbols and the elements of $X^+$ words. We denote by $l(w)$ the length of words $w$, i.e., if $w=x_1\ldots x_m$, $x_i \in X$, then $l(w)=m$. Given $w \in X^+$, and $p,s \in X^*$, if $ps=w$ then $p$ is
a prefix of $w$ and $s$ is a suffix of $w$. If $p$ is a prefix of $w$ we write $w \geq p$ and if $p \neq w$ we say that $p$ is a proper prefix of $w$.

For any $Z \subseteq X^+$ define

$$
\text{Pref}(Z) = \{ x \in X^* \mid \exists w \in Z \exists s \in X^* [w = xs] \},
$$

$$
\text{Prop}(Z) = \{ x \in X^* \mid \exists w \in Z \exists s \in X^* [w = xs] \},
$$

$$
\text{Suf}(Z) = \{ x \in X^+ \mid \exists w \in Z \exists p \in X^+ [w = px] \}.
$$

In words, Pref($Z$) is the set of all prefixes of elements of $Z$, Prop($Z$) is the set of all proper prefixes of elements of $Z$ and Suf($Z$) $\cup \{ \lambda \} \cup Z$ gives the set of all suffixes of elements of $Z$.

Let $x \in X$ and $Z \subseteq X^*$. With $Z \cdot x$ we denote the set

$$
Z \cdot x = \{ y \in X^+ \mid y = zx \text{ and } z \in Z \};
$$

and with $x^{-1} \cdot Z$ we denote the set

$$
x^{-1} \cdot Z = \{ y \in X^* \mid xy \in Z \}.
$$

Given a finite set $A$ of source symbols, a multivalued encoding is any mapping $F : A \rightarrow 2^X$ from the source alphabet $A$ into the set of all subsets of $X^+$, denoted by $2^X$. We assume that for each $a \in A$ the set $F(a)$ is finite. In order to define the encoding of strings of source symbols, we expand the domain of $F$ from $A$ to $A^*$ in the following way:

(i) $F(\lambda) = \{ \lambda \}$;

(ii) for each $x \in A^*$ and for each $y \in A$

$$
F(xy) = F(x) \cdot F(y) = \{ \alpha \beta \mid \alpha \in F(x) \text{ and } \beta \in F(y) \}.
$$

For each string of source symbols $x \in A^*$, $F(x)$ denotes the set of all possible encodings of the string $x$. It is obvious that the above definition reduces to the definition of ordinary encoding when sets $F(a)$ are singletons for each $a \in A$. Moreover, denote by $C$ the set of all codewords, i.e., $C = \bigcup_{a \in A} F(a)$ and by $C^+$ the set of all code messages, i.e. $C^+ = \bigcup_{x \in A^+} F(x)$.

Intuitively, the property of decipherability with finite delay assures the possibility of deciphering every code message sequentially, from left to right, with some delay. This implies that it is possible to start the decoding before the transmission ends. An immediate and quite informal definition of this property is the following: a multivalued encoding is decipherable with finite delay $P$ if and only if the individualization in any message of $P + n$ initial consecutive codewords suffices to determine the first $n$ symbols of the source sequence that generated the message. A more formal definition, in terms of generalized sequential machines, is the following:

**Definition 1.** A multivalued encoding $F$ is decipherable with finite delay $P$ \[2, 8\] if $P$ is the smallest integer for which the nondeterministic generalized sequential
machine (gsm) $M$ that implements $F$ has an inverse machine $M^{-1}$ such that the connection $M^{-1}M$ of $M$ and $M^{-1}$ in their initial state amounts to a delay machine with maximum delay $P$. A nondeterministic generalized sequential machine $D$ is called a delay machine with maximum delay $P$ if for any arbitrary input $x \in A^+$, $l(x) > P$, any associated output $y$ is a prefix of $x$, with $l(x) - l(y) \leq P$.

The gsm in the above definition is a nondeterministic machine (see [1]) which acts as an encoder, that is, associates to each string $x$ of source symbols one of the possible encodings contained in $F(x)$. Conversely, the inverse machine $M^{-1}$ corresponds to a decoder, and this correspondence will be fully analyzed in Section 4.

3. A preliminary result

In this section we will provide an auxiliary result that will be needed in the proof of the correctness of our algorithms. To this aim let us introduce the following definitions. Let there be given a multivalued encoding $F : A \to 2^{X^+}$. Define the decomposition of $\beta \in \text{Pref}(C^+)$ a sequence $d(\beta) = w_1 \ldots w_h \overline{\beta}$ ($h \geq 0$), such that $\beta = d(\beta)$, $w_i \in C$, $\overline{\beta} \in \text{Prop}(C)$. For any decomposition of $\beta \in \text{Pref}(C^+)$, say $d(\beta) = w_{i_1} \ldots w_{i_h} \overline{\beta}$, define the deciphering of $\beta$ as the corresponding source sequence $D_i(\beta) = a_{i_1} \ldots a_{i_h} x$, with $a_j \in A$, $w_i \in F(a_{i_1} \ldots a_{i_h})$ and $x \in A \cup \{\lambda\}$ ($x \in A$ if all codewords beginning with $\overline{\beta}$ belong to $F(x)$, $x = \lambda$ otherwise).

Define now mappings $G : \text{Pref}(C^+) \to A^*$ and $H : \text{Pref}(C^+) \to 2^{\text{Pref}(C^+)} \times 2^{\text{Suf}(C)}$ in the following way:

$$G(\beta) = a_1 \ldots a_s \in A^*,$$

where $a_1 \ldots a_s$ ($s \geq 0$), is the longest prefix of all decipherings of $\beta$;

$$H(\beta) = (R_1(\beta), R_2(\beta)),$$

where

$$R_1(\beta) = \{x \in \text{Pref}(C^+) \mid \exists w_1 \ldots w_{s+1} \in F(G(\beta)), \exists d(\beta)[d(\beta) = w_1 \ldots w_{s+1} x]\},$$

$$R_2(\beta) = \{x \in \text{Suf}(C) \mid \exists w_1 \ldots w_{s+1} \in F(G(\beta)), \exists d(\beta)[d(\beta) x = w_1 \ldots w_{s+1}]\}.$$

Further, let us extend the domain of $G$ and $H$. Let

$$Y \subseteq \text{Pref}(C^+), \quad Z \subseteq \text{Suf}(C) \cup \{\lambda\}, \quad Y \neq \emptyset \neq Z.$$

Define $G(Y, Z)$ as the longest common prefix, if any, of all sequences $G(\beta)$, $\beta \in Y$, if $Z = \emptyset$; $G(Y, Z) = \lambda$ if $Z \neq \emptyset$; and define

$$H(Y, Z) = \begin{cases} (R_1(\beta), R_2(\beta)), & \text{if } Z = \emptyset; \\ (Y \cup \{\lambda\}, Z - \{\lambda\}), & \text{if } \lambda \in Z; \\ (Y, Z), & \text{otherwise.} \end{cases}$$
where

\[ R_1(Y) = \{ x \in \text{Pref}(C^+) \mid \exists w_1 \ldots w_s \in F(G(Y, \emptyset)), \exists \beta \in Y, \exists d(\beta)[d(\beta) = w_1 \ldots w_s x] \}. \]

\[ R_2(Y) = \{ x \in \text{Suf}(C) \mid \exists w_1 \ldots w_s \in F(G(Y, \emptyset)), \exists \beta \in Y, \exists d(\beta)[d(\beta)x = w_1 \ldots w_s] \}. \]

Finally, for \( x \in X \) such that

\[ (Y \cdot x \cap \text{Pref}(C^+), x^{-1} \cdot Z) \in 2^{\text{Pref}(C^+)} \times 2^{\text{Suf}(C) \cup \{ \lambda \}} - (\emptyset, \emptyset) \]

define

\[ G(Y, Z, x) = G(Y \cdot x \cap \text{Pref}(C^+), x^{-1} \cdot Z), \]

and

\[ H(Y, Z, x) = H(Y \cdot x \cap \text{Pref}(C^+), x^{-1} \cdot Z). \]

The following key result holds, whose proof is given in Appendix A.

**Lemma 1.** For any \( \beta \in \text{Pref}(C^+) \), for any \( b \in X \) such that \( \beta b \in \text{Pref}(C^+) \) the following results hold:

\[ G(\beta b) = G(\beta)G(H(\beta), b), \quad (1) \]

\[ H(\beta b) = H(H(\beta), b). \quad (2) \]

The following example illustrates the above introduced concepts.

**Example 1.** Let \( A = \{0, 1\} \) be the set of source symbols, \( X = \{a, b, c, d\} \) be the set of code symbols and \( C = \{aa, aab, bb, bbc, cd\} \) be the set of codewords. Let the multivalued encoding \( F \) be defined by

\[ F(0) = \{aa, aab, bb\}, \quad F(1) = \{bbc, cd\}. \]

Consider \( aa \in \text{Pref}(C^+) \); \( aa \) has two decompositions, one given by the codeword \( aa \) and the other given by the prefix \( aa \) of the codeword \( aab \). Both decompositions give the same deciphering, namely \( 0 \). It follows that

\[ G(aa) = 0 \quad \text{and} \quad H(aa) = (\{\lambda\}, \{b\}). \]

If we consider \( G(\{\lambda\}, \{b\}) \) we obtain, by definition

\[ G(\{\lambda\}, \{b\}, b) = G(\{\lambda\} \cdot b, b^{-1} \cdot \{b\}) = G(\{b\}, \{\lambda\}) = \lambda \]

and

\[ H(\{\lambda\}, \{b\}, b) = H(\{\lambda\} \cdot b, b^{-1} \cdot \{b\}) = H(\{b\}, \{\lambda\}) = (\{b, \lambda\}, \emptyset). \]

On the other hand, \( aab \in \text{Pref}(C^+) \) has two decompositions, one given by the codeword \( aa \) followed by the prefix \( b \) of codeword \( bb \) and the other given by the codeword \( aab \). Both decompositions give the same deciphering \( 0 \). It follows that

\[ G(aab) = 0 \quad \text{and} \quad H(aab) = (\{b, \lambda\}, \emptyset). \]

Notice that

\[ G(aab) = G(aa)G(H(aa), b), \quad H(aab) = H(H(aa), b). \]
4. Construction of decoders of multivalued encodings

In this section we shall develop three algorithms for the construction of sequential decoders for multivalued encodings. Let us first state the formal definition of a decoder.

Let \( D = \{S, s_0, X, A, f, g\} \) be a (deterministic) finite sequential machine, where
- \( S \) = state set;
- \( s_0 \) = initial state;
- \( X \) = input alphabet (= set of code letters);
- \( A \) = output alphabet (= set of source letters);
- \( f : S \times X \rightarrow S \) (= transition function);
- \( g : S \times X \rightarrow A^* \) (= output function).

Notice that our definition of sequential machine is substantially equivalent to that of finite transducer, as defined in [1].

**Definition 2.** The finite sequential machine \( D \) is a decoder for the multivalued encoding \( F : A \rightarrow 2^X^+ \) if and only if there exists an integer \( \tau \geq 0 \) such that for any \( a_i, a_{i+1}, \ldots, a_{i+k} \in A^+ \), for any \( w_i, w_{i+1}, \ldots, w_k \in F(a_i, a_{i+1}, \ldots, a_k) \) and for all \( \beta \in C' \):

\[
g(s_0, w_i, w_{i+1}, \ldots, w_k, \beta) \geq a_i, a_{i+1}, \ldots, a_k.
\]

The smallest number \( \tau \) such that (3) holds will be called the (decoding) delay of the decoder \( D \).

In words, the meaning of the above definition is the following: the machine \( D \) is a decoder with delay \( \tau \) if and only if, having as input \( k \) codewords followed by at least \( \tau \) other codewords, \( D \) is able to decode at least the first \( k \) codewords, leaving undeciphered at most \( \tau \) terminal codewords. The following lemma says that \( D \) is a decoder for the multivalued encoding \( F \) if and only if \( D \) is an inverse machine for the nondeterministic gsm \( M \) that implements \( F \).

**Lemma 2.** Let \( M = (Q, q_0, A, X, \lambda, \delta) \) be the nondeterministic gsm that implements \( F \) and \( M^{-1} = (S, s_0, X, A, f, g) \) be a deterministic sequential machine. The serial connection \( M^{-1}M \) of \( M \) and \( M^{-1} \) in their initial states is a delay machine with (maximum) delay \( \tau \) if and only if \( M^{-1} \) is a decoder with delay \( \tau \) for \( F \).

**Proof.** Let \( M^{-1} \) be a decoder with delay \( \tau \) for \( F \) and \( z = a_i, a_{i+1}, \ldots, a_{i+k+1}, \ldots, a_{i+k} \in A^+ \), \( w_i, \ldots, w_{i+k}, w_{i+k+1}, \ldots, w_{i+k+1} \in \delta(q_0, z) = F(z) \). By the definition of a decoder one has that

\[
g(s_0, w_i, w_{i+1}, \ldots, w_{i+k}, \beta) \geq a_i, a_{i+1}, \ldots, a_k.
\]
That is, for any \( z \in A^+ \), \( l(z) > \tau \), the output of the machine \( M^{-1}M \) with \( M \) and \( M^{-1} \) in their initial states is a prefix \( y \) of \( z \) with \( l(z) - l(y) \leq \tau \). It follows that \( M^{-1}M \) is a delay machine with delay \( \tau \).

Conversely, let us assume \( M^{-1}M \) is a delay machine with delay \( \tau \). By definition, any output \( y \) associated with an arbitrary input \( z = a_1 \ldots a_k \in A^+ \), \( a \in A^+ \), has \( a_i \ldots a_k \) as prefix. One gets that \( a_i \ldots a_k \) is a prefix of the output of \( M^{-1} \) associated to any encoding of \( z \) that enters \( M^{-1} \). It follows that \( M^{-1} \) is a decoder for \( F \).

From the definition of decipherability delay of a multivalued encoding and from Lemma 2 one gets the following:

**Corollary 1.** The decipherability delay \( \tau \) of a multivalued encoding \( F \) is given by

\[
P = \min \{ \tau \mid \tau \text{ is the decoding delay of a decoder for } F \},
\]

where the minimization takes place over all decoders for \( F \).

We are left with the problem of the effective construction of decoders for multivalued encodings. The following theorem (the sufficient part) will provide three algorithms for designing sequential decoders for multivalued encodings. We also remark that conditions expressed in Theorem 1 are necessary and sufficient conditions for a multivalued encoding to have finite decipherability delay.

**Theorem 1.** Let \( F \) be a multivalued encoding. A necessary and sufficient condition for a decoder to exist for \( F \) is that there exists \( t \geq 0 \) such that for any \( k \), for any \( a_1 \ldots a_k \in A^+ \), for any \( w_1 \ldots w_k \in F(a_1 \ldots a_k) \) and for any \( \beta \in C^+ \) the following holds

\[
G(w_1 \ldots w_k \beta) \geq a_1 \ldots a_k.
\]

The smallest number \( t \) such that (4) holds is equal to the decipherability delay of \( F \).

**Proof.** Necessity. We shall first provide that for any decoder \( D = \langle S, s_0, X, A, f, g \rangle \) for \( F \) one has that

\[
\forall \beta \in \text{Pref}(C^+) \quad G(\beta) \geq g(s_0, \beta).
\]

Indeed, assume that (5) is not true; that is, either there exists \( \beta \in \text{Pref}(C^+) \) such that

\[
g(s_0, \beta) = a_1 a_2 \ldots a_h a_{h+1} \ldots a_k,
\]

\[
G(\beta) = a_1 a_2 \ldots a_h, \quad 0 \leq h < k,
\]

or there exists \( \beta \in \text{Pref}(C^+) \) such that

\[
g(s_0, \beta) = a_1 \ldots a_h a_{h+1} \ldots a_k,
\]

\[
G(\beta) = a_1 \ldots a_h b_1 \ldots b_r, \quad b_1 \neq a_{h+1}.
\]
We shall discuss the first case only since the second can be handled similarly. Let \( \tau \) be the decoding delay of \( D \). It is possible to distinguish the following situations:

(i) There exists a decomposition of \( \beta \), say \( d(\beta) = w_i \ldots w_{ih} w_{ih+1} \ldots w_{in} \), such that \( w_i \ldots w_{ih} \in F(a_i \ldots a_h) \), \( w_{ih+1} \in F(a_{ih+1}) \), \( a_{h+1} \in A \), and \( a_{ih+1} \neq a_{h+1} \). Let \( \mu \) be such that \( \beta \mu \in C^\tau \). From the definition of decoder it follows that

\[
g(s_0, \beta \mu) = g(s_c, w_i \ldots w_{ih} w_{ih+1} \ldots w_n \beta \mu) \geq a_1 \ldots a_h a_{ih+1}
\]

and

\[
g(s_0, \beta \mu) \geq g(s_0, \beta) = a_1 \ldots a_h a_{h+1} \ldots a_k 
\]

that contradicts the assumption that \( a_{ih+1} \neq a_{h+1} \).

(ii) There exists a decomposition of \( \beta \), say \( d(\beta) = w_1 \ldots w_h \), with \( w_1 \ldots w_h \in F(a_i \ldots a_h) \) and \( \beta \) such that \( \beta x = w \in F(a) \), \( a \in A \), \( a \neq a_{h+1} \), for some \( x \in X^+ \). Let \( \mu \in C^\tau \). One has that

\[
g(s_0, \beta x \mu) = g(s_0, w_1 \ldots w_h w \mu) \geq a_1 \ldots a_h a
\]

and

\[
g(s_0, \beta x \mu) \geq g(s_0, \beta) \geq a_1 \ldots a_h a_{h+1} \ldots a_k 
\]

that contradicts the assumption \( a \neq a_{h+1} \).

(iii) There exists a decomposition of \( \beta \), say \( d(\beta) = w_1 \ldots w_{h-1} \), with \( w_1 \ldots w_{h-1} \in F(a_i \ldots a_{h-1}) \) and such that all codewords beginning with \( \beta \) belong to \( F(a_h) \). It follows that for \( w \in F(a) \), \( w_h - \beta x \in F(a_h) \), \( a \neq a_{h+1} \), and \( \mu \in C^\tau \) it results that

\[
g(s_0, \beta x \mu) = g(s_0, w_1 \ldots w_{h-1} w_h w \mu) \geq a_1 \ldots a_{h-1} a_h a 
\]

and

\[
g(s_0, \beta x \mu) \geq g(s_0, \beta) \geq a_1 \ldots a_h a_{h+1} \ldots a_k 
\]

that contradicts the assumption \( a \neq a_{h+1} \).

Having proved (9), the necessity of (4) follows immediately from the definition of a decoder and from (5).

**Sufficiency.** Let us assume (4) true. We shall provide three algorithms to construct decoders for \( F \) with minimum decoding delay. By Corollary 1, the decoding delay of the decoders coincides with the decipherability delay \( P \) of \( F \).

**First method.** Let us consider the finite-state machine \( D = (S, S_0, X, A, f, g) \) defined as follows

\[
S = \{ s(Y, Z) \mid (Y, Z) \in M \}, \quad s_0 = s(\{\lambda\}, \emptyset)
\]

where \( Y \subset \text{Pref}(C^+) \), \( Z \subset \text{Suf}(C) \), and \( M \) is constructed according to the following rules (a) and (b):

(a) \( (\{\lambda\}, \emptyset) \in M \);

(b) for any \( b \in X \), for any \( (Y, Z) \in M \), if \( b^{-1} \cdot Z \) and \( Y \cdot b \cap \text{Pref}(C^+) \) are not both empty, then \( H(Y, Z, b) \in M \).
Decoders for multivalued encodings

It is easy to see that \( M \) (and therefore \( S \)) is finite (see Appendix B). The transition function \( f \) and the output function \( g \), both defined on the set of pairs \((s_{(Y,Z)}, b) \in S \times X\) such that \( b^{-1} \cdot Z \) and \( Y \cdot b \cap \text{Pref}(C^+) \) are not both empty, are determined in the following way

\[
g(s_{(Y,Z)}, b) = G(Y, Z, b),
\]

\[
f(s_{(Y,Z)}, b) = s_{H(Y,Z,b)}.
\]

Let us notice that the above definitions are consistent. Indeed, if \((s_{(Y,Z)}, b) \in S \times X\), (6) gives \( g(s_{(Y,Z)}, b) \in A^* \), whereas (7), together with (b), gives \( f(s_{(Y,Z)}, b) \in S \). We shall show that \( D \) is a decoder with decoding delay equal to the decipherability delay \( P \) of the multivalued encoding \( F \). To this aim, because of (4), it is sufficient to show that

\[
\forall \beta \in \text{Pref}(C^+) \quad g(s_0, \beta) = G(\beta).
\]

We shall prove the formula (8) simultaneously with the formula

\[
f(s_0, \beta) = s_{H(\beta)}
\]

by induction on the length of word \( \beta \). Let \( l(\beta) = 0 \) (i.e. \( \beta = \lambda \)). From the above definitions one gets

\[
g(s_0, \lambda) = \lambda = G(\lambda), \quad f(s_0, \lambda) = s_0 = s_{(\lambda), 0} = s_{H(\lambda)}.
\]

Let us now assume that (8) and (9) hold for all \( \beta \in \text{Pref}(C^+) \), \( l(\beta) = n \). We shall prove that they hold also for all \( \beta - \beta b \in \text{Pref}(C^+) \), \( l(\beta) = n \), \( b \in X \) and \( l(\beta) = n + 1 \). By Lemma 2 one has

\[
f(s_0, \beta) = f(s_0, \beta b) = f(f(s_0, \beta), b) = f(s_{H(\beta), b}) = s_{H(\beta b)}
\]

and then (9) follows. Moreover,

\[
g(s_0, \beta) = g(s_0, \beta b) = g(s_0, \beta)g(f(s_0, \beta), b)
\]

\[
= G(\beta)g(s_{H(\beta), b}) = G(\beta)G(H(\beta), b) = G(\beta b),
\]

which proves (8).

From (5) and (8) one gets that the decoding delay of the decoder built according to the above described method is minimum with respect to all decoders for \( F \). From Corollary 1 it follows that our algorithm produces decoders with decoding delay equal to the decipherability delay of the multivalued encoding \( F \).

**Second method.** Let us define a mapping

\[
G_1 : 2^{\text{Pref}(C^+)} \times 2^{\text{Suf}(C^+) \cup \{\lambda\}} \setminus (\emptyset, \emptyset) \to A^*
\]

as follows

\[
G_1(Y, Z) = \begin{cases} 
  x \in A, & \text{if } G(Y, Z) = xy, \quad y \in A^*; \\
  \lambda, & \text{otherwise.}
\end{cases}
\]
For any nonempty $Y \subset \text{Pref}(C^+)$ let us consider
\[
R_1^{(1)}(Y) = \{ x \in \text{Pref}(C^+) \mid \exists w_1 \ldots w_k \in F(G_1(Y, \emptyset)), \exists \beta \in Y, \exists d(\beta)[d(\beta) = w_1 \ldots w_k x]\},
\]
\[
R_2^{(1)}(Y) = \{ x \in \text{Suf}(C) \mid \exists w_1 \ldots w_k \in F(G_1(Y, \emptyset)), \exists \beta \in Y, \exists d(\beta)[d(\beta) x = w_1 \ldots w_k]\}.
\]

Define a mapping $H_1 : 2^{\text{Pref}(C^+)} \times 2^{\text{Suf}(C)} \cup \{\emptyset\} \rightarrow 2^{\text{Pref}(C^+)} \times 2^{\text{Suf}(C)}$ as follows
\[
H_1(Y, Z) = \begin{cases} 
(R_1^{(1)}(Y), R_2^{(1)}(Y)), & \text{if } Z = \emptyset, \\
(Y \cup \{\lambda\}, Z - \{\lambda\}), & \text{if } \lambda \in Z, \\
(Y, Z), & \text{otherwise.}
\end{cases}
\]

Let us extend the domain of $G_1$ and $H_1$ by setting, for any $(Y, Z, b)$ such that
\[
(Y \cdot b \cap \text{Pref}(C^+), b^{-1} \cdot Z) \in 2^{\text{Pref}(C^+)} \times 2^{\text{Suf}(C)} \cup \{\lambda\} - (\emptyset, \emptyset)
\]
the values of $G_1$ and $H_1$ as follows
\[
G_1(Y, Z, b) = G_1(Y \cdot b \cap \text{Pref}(C^+), b^{-1} \cdot Z),
\]
\[
H_1(Y, Z, b) = H_1(Y \cdot b \cap \text{Pref}(C^+), b^{-1} \cdot Z).
\]

Let us finally construct the decoder $D_1 = (S_1, s_0, X, A, f_1, g_1)$ in the following way
\[
S_1 = \{ s_{(Y,Z)} \mid (Y,Z) \in M_1 \}, \quad s_0 = s_{(\{\lambda\}, \emptyset)}
\]
where $M_1$ is constructed according to the following (i) and (ii).

(i) $(\{\lambda\}, \emptyset) \in M_1$.

(ii) For any $(Y,Z) \in M_1$, for any $b \in X$, if $Y \cdot b \cap \text{Pref}(C^+)$ and $b^{-1} \cdot Z$ are not both empty, then $H_1(Y, Z, b) \in M_1$.

It is easy to see that $M_1$ (and then $S_1$) is finite (see Appendix B). The transition function $f_1$ and the output function $g_1$, defined on the set of pairs $(s_{(Y,Z)}, b) \in S_1 \times X$ such that $b^{-1} \cdot Z$ and $Y \cdot b \cap \text{Pref}(C^+)$ are not both empty, are determined in the following way:
\[
f_1(s_{(Y,Z)}, b) = s_{H_1(Y, Z, b)}, \quad g_1(s_{(Y,Z)}, b) = G_1(Y, Z, b).
\]

It is easily seen that $D_1$ decodes every message with minimum delay.

**Third method.** Let $P$ be the decipherability delay of the multivalued encoding $F$. Let us define a mapping $G_2 : 2^{\text{Pref}(C^+)} \times 2^{\text{Suf}(C)} \cup \{\lambda\} - (\emptyset, \emptyset) \rightarrow A^*$ as follows
\[
G_2(Y, Z) = \begin{cases} 
x \in A, & \text{if } \exists \beta \in Y \text{ such that } \beta = w\bar{\beta} \text{ with } w \in F(x) \text{ and } \bar{\beta} \in C^P; \\
\lambda, & \text{otherwise.}
\end{cases}
\]
For any nonempty $Y \subset \text{Pref}(C^+)$ let us consider

$$R_1^{(2)}(Y) = \{ x \in \text{Pref}(C^+) \mid \exists w_1 \ldots w_k \in F(G_2(Y, \emptyset)), \exists \beta \in Y, \exists d(\beta)[d(\beta) = w_1 \ldots w_k] \},$$

$$R_2^{(2)}(Y) = \{ x \in \text{Suf}(C) \mid \exists w_1 \ldots w_k \in F(G_2(Y, \emptyset)), \exists \beta \in Y, \exists d(\beta)[d(\beta)x = w_1 \ldots w_k] \},$$

Define a mapping $H_2 : 2^{\text{Pref}(C^+)} \times 2^{\text{Suf}(C) \cup \{\lambda\}} \to 2^{\text{Pref}(C^+) \times 2^{\text{Suf}(C)}}$ as follows

$$H_2(Y, Z) = \begin{cases} (R_1^{(2)}(Y), R_2^{(2)}(Y)), & \text{if } Z = \emptyset, \\
(Y \cup \{\lambda\}, Z - \{\lambda\}), & \text{if } \lambda \in Z, \\
(Y, Z), & \text{otherwise}. \end{cases}$$

Let us extend the domain of $G_2$ and $H_2$ by setting, for any $(Y, Z, b)$ such that $(Y \cdot b \cap \text{Pref}(C^+), b^{-1} \cdot Z) \in 2^{\text{Pref}(C^+) \times 2^{\text{Suf}(C) \cup \{\lambda\}}} - (\emptyset, \emptyset)$

the values of $G_2$ and of $H_2$ as follows

$$G_2(Y, Z, b) = G_1(Y \cdot b \cap \text{Pref}(C^+), b^{-1} \cdot Z),$$

$$H_2(Y, Z, b) = H_1(Y \cdot b \cap \text{Pref}(C^+), b^{-1} \cdot Z).$$

Fig. 1. Decoder obtained according to the first method.
Let us finally construct the decoder \( D_2 = (S_2, s_0, X, A, f_2, g_2) \) in the following way

\[
S_2 = \{ s_{(Y,Z)} \mid (Y, Z) \in M_2 \}, \quad s_0 = s_{(\lambda, \emptyset)},
\]

where \( M_2 \) is constructed according to (j) and (jj).

(j) \( (\{\lambda\}, \emptyset) \in M_2 \).

(jj) For any \( (Y, Z) \in M_2 \), for any \( b \in X \), if \( Y \cdot b \cap \text{Pref}(C^+) \) and \( b^{-1} \cdot Z \) are not both empty, then \( H_2(Y, Z, b) \in M_2 \).

It is easy to see that \( M_2 \) (and then \( S_2 \)) is finite (see Appendix B). The transition function \( f_2 \) and the output function \( g_2 \), defined on the set of pairs \( (s_{(Y,Z)}, b) \in S_2 \times X \) such that \( b^{-1} \cdot Z \) and \( Y \cdot b \cap \text{Pref}(C^+) \) are not both empty, are determined in the following way:

\[
f_2(s_{(Y,Z)}, b) = s_{H_2(Y,Z,b)}, \quad g_2(s_{(Y,Z)}, b) = G_2(Y,Z,b).
\]

![Decoder obtained according to the second method.](image-url)
It is easily seen that $D_2$ decodes every message with minimum delay. It is also possible to show that the size of $S$ is smaller than or equal to the size of $S_1$ which, in turn, is smaller than or equal to the size of $S_2$. □

**Example 2.** Let $A = \{0, 1\}$, $X = \{a, b, c, d\}$, $C = \{aa, aab, bb, bbc, cd\}$. The multivalued encoding $F$ given by

$$F(0) = \{aa, aab, bb\}, \quad F(1) = \{bbc, cd\}$$

has decipherability delay 1. The decoders for $F$ constructed according the first, second and third method are shown in Fig. 1, Fig. 2, Fig. 3, respectively.

![Decoder obtained according to the third method.](image-url)
5. Conclusions

In this paper three algorithms for constructing minimum delay sequential decoders for multivalued encodings have been presented. The algorithms produce decoders exhibiting various features. In particular, decoders obtained according to the second and third method output, at each transition, either a single letter or the empty word; whereas decoders obtained according to the first method may output words of greater and unpredictable lengths. In addition, a decoder built according to the third method has the following property: if $x$ is the output of the decoder corresponding to an input message $\beta$ and $y$ is the longest deciphering of $\beta$, then $l(y) - l(x)$ is constant and equal to $P$. This property resembles a sort of "constant decipherability delay" of the decoder. These useful properties are obtained at the expense of some increase in the number of states of the decoders. For instance, decoders considered in Example 2 have 10, 13, 23 states, respectively. It should also be remarked that, in case of ordinary codes, our decoders behave essentially as the ones considered by Levensthein in [6].

Appendix A

Proof of Lemma 1. Given $\beta \in \text{Pref}(C^+)$, assume $G(\beta) = a_1 \ldots a_k$ ($k \geq 0$) and $H(\beta) = (Y, Z)$. In order to prove (1) it is convenient to distinguish two cases: $b^{-1} \cdot Z = \emptyset$ and $b^{-1} \cdot Z \neq \emptyset$;

Case 1: $b^{-1} \cdot Z = \emptyset$. This implies $Y \cdot b \cap \text{Pref}(C^+) \neq \emptyset$. Indeed, it is easily seen that, if $\beta b \in \text{Pref}(C^+)$, the sets $Y \cdot b \cap \text{Pref}(C^+)$ and $b^{-1} \cdot Z$ cannot be both empty. Let us assume

$G(H(\beta), b) = G(Y, Z, b) = G(Y \cdot b \cap \text{Pref}(C^+), b^{-1} \cdot Z) = G(Y \cdot b \cap \text{Pref}(C^+), \emptyset) = a_1 \ldots a_n$.

By definition, $a_1 \ldots a_n$ is the longest common prefix of all decipherings of $\beta$, whereas $a_1 \ldots a_k$ is the longest common prefix of all decipherings of sequences $\beta b \in Y \cdot b \cap \text{Pref}(C^+)$ obtained from $\beta b$ by removing the first $k$ words. Therefore, $G(\beta)G(H(\beta), b) = a_1 \ldots a_k a_1 \ldots a_n$ is the longest common prefix of all decipherings of $\beta b$, that is, by definition is $G(\beta b)$ itself.

Case 2: $b^{-1} \cdot Z \neq \emptyset$. This implies that $b$ is a prefix of a suffix borrowed by $\beta$ (with respect to $G$) and then $G(\beta b) = G(\beta)$. Since $b^{-1} \cdot Z \neq \emptyset$ implies $G(H(\beta), b) = \lambda$, one gets (1).

In order to show (2) it is convenient to distinguish three cases: $b^{-1} \cdot Z = \emptyset$; $b^{-1} \cdot Z \neq \emptyset$ and $b \in Z$; $b^{-1} \cdot Z \neq \emptyset$ and $b \notin Z$.

Case 1: $b^{-1} \cdot Z = \emptyset$. Then (2) gives

$H(\beta b) = H(H(\beta), b) = H(Y, Z, b) = H(Y \cdot b \cap \text{Pref}(C^+), \emptyset)$.
and then implies
\[ R_1(\beta b) = R_1(R_1(\beta) \cdot b \cap \text{Pref}(C^+)) \]
and
\[ R_2(\beta b) = R_2(R_1(\beta) \cdot b \cap \text{Pref}(C^+)). \]

Let \( x \in R_1(\beta b) \). Then there exists \( w_i \ldots w_n \in F(G(\beta b)) \) such that \( w_i \ldots w_n x = \beta b \) with \( w_i \ldots w_i \beta = \beta \) for some \( \beta_i \in R_1(\beta) \) and \( k \leq n \). Hence, it follows that \( w_{i+1} \ldots w_n x = \beta_i b \), since \( \beta_i b \in R_1(\beta) \cdot b \cap \text{Pref}(C^+) \), gives \( x \in R_1(R_1(\beta) \cdot b \cap \text{Pref}(C^+)) \).

Conversely, let \( x \in R_1(R_1(\beta) \cdot b \cap \text{Pref}(C^+)) \). Then there exists \( w_i \ldots w_n \in F(G(H, b)) \) such that \( w_i \ldots w_n x = \beta_i b \) for some \( \beta_i \in R_1(\beta) \) for which \( \beta_i b \in R_1(\beta) \cdot b \cap \text{Pref}(C^+) \). It follows that there exists \( w_i \ldots w_i \beta = \beta \). Therefore, \( w_i \ldots w_i x = w_i \ldots w_i \beta \beta_i b = \beta b \) and then \( x \in R_1(\beta b) \).

Let \( x \in R_2(\beta b) \). Since, by hypothesis, \( b^{-1} \cdot Z = \emptyset \), there exists \( w_i \ldots w_n \in F(G(\beta b)) \) such that \( w_i \ldots w_n x = \beta_i b \) with \( w_i \ldots w_i \beta = \beta \) for some \( \beta_i \in R_1(\beta) \).

It follows that \( w_i \ldots w_n = w_i \ldots w_i \beta \beta_i x \) and \( w_{i+1} \ldots w_n = \beta_i x \).

Therefore, one gets \( x \in R_2(R_1(\beta) \cdot b \cap \text{Pref}(C^+)) \).

Conversely, assume \( x \in R_2(R_1(\beta) \cdot b \cap \text{Pref}(C^+)) \). One has that there exists \( w_i \ldots w_n \in F(G(H, b)) \) such that \( w_i \ldots w_n x = \beta_i b \) for some \( \beta_i b \in R_1(\beta) \cdot b \cap \text{Pref}(C^+) \).

Since \( \beta_i \in R_1(\beta) \) one has that there exists \( w_i \ldots w_n \in F(G(\beta)) \) such that \( w_i \ldots w_i \beta = \beta \). Therefore,
\[ w_i \ldots w_n = w_i \ldots w_i \beta \beta_i x \]
and then \( x \in R_2(\beta b) \).

Case 2: \( b^{-1} \cdot Z \neq \emptyset \) and \( b \notin Z \). Then (2) gives the equivalent relations
\[ R_1(\beta b) = R_1(\beta) \cdot b \cap \text{Pref}(C^+), \quad R_2(\beta b) = b^{-1} \cdot R_2(\beta). \]

Let \( x \in R_1(\beta b) \). Then there exists \( w_i \ldots w_k \in F(G(\beta b)) \) such that \( w_i \ldots w_k x = \beta b \).
Since \( G(\beta b) = G(\beta) \) and \( b \notin Z \) one has \( x = x' b \) and \( w_i \ldots w_k x' = \beta \), for some \( x' \in X^+ \). This implies that \( x' \in R_1(\beta) \); therefore \( x \) belongs to \( R_1(\beta) \cdot b \cap \text{Pref}(C^+) \).

Conversely, let \( x \in R_1(\beta) \cdot b \cap \text{Pref}(C^+) \), for instance assume \( x = \beta_i b \) with \( \beta_i \in R_1(\beta) \). Then, it follows that there exists \( w_i \ldots w_k \in F(G(\beta)) \) such that \( w_i \ldots w_k x = w_i \ldots w_i \beta \beta_i b = \beta b \) and therefore \( x \in R_1(\beta b) \).

Let \( x \in R_2(\beta b) \). One has that there exist \( x \in \text{Suf}(C) \) and \( w_i \ldots w_k = \beta b x \). Hence it follows \( bx \in R_2(\beta) = Z \) and then \( x \in b^{-1} \cdot R_2(\beta) \).

Conversely, let \( x \in b^{-1} \cdot R_2(\beta) \). If follows that \( bx \in R_2(\beta) \). That is, that exists \( w_i \ldots w_k \in F(G(\beta)) \) such that \( w_i \ldots w_k = \beta b x \). From this, in the hypothesis that \( b \notin R_2(\beta) = Z \), it follows that \( x \neq \lambda \) and therefore \( x \in R_2(\beta b) \).

Case 3: \( b^{-1} \cdot Z \neq \emptyset \) and \( b \in Z \). Then (2) is equivalent to
\[ R_1(\beta b) = R_1(\beta) \cdot b \cap \text{Pref}(C^+) \cup \{ \lambda \}, \quad R_2(\beta b) = b^{-1} \cdot R_2(\beta) \cup \{ \lambda \}. \]
From $b \in Z$ one gets that there exists $w_i \ldots w_k = \beta b$. That is, $\lambda \in R_1(\beta b)$, that together with $G(\beta) = G(\beta b)$ gives

$$R_1(\beta b) = R_1(\beta) \cdot b \cap \text{Pref}(C^+) \cup \{\lambda\}, \quad R_2(\beta b) = b^{-1} \cdot R_2(\beta) - \{\lambda\}. \quad \square$$

Appendix B

In order to show that $M$ is finite we note first that for all $(Y, Z) \in M$ there exists $\beta \in \text{Pref}(C^+)$ such that

$$H(\beta) = (Y, Z). \quad (B.1)$$

Relation (B.1) can be easily proved by induction on the length of $\beta$. Let $l_{\text{max}}$ be the length of the longest word in $C$ and $P$ the decipherability delay of the multivalued encoding $F$. We want to prove that

$$\forall \alpha \in Y \quad l(\alpha) \leq (P + 1)l_{\text{max}} - 1, \quad (B.2)$$

$$\forall \gamma \in Z \quad l(\gamma) \leq l_{\text{max}} - 1. \quad (B.3)$$

(B.3) is obvious since, by definition, $Z \subseteq \text{Suf}(C)$. In order to prove (B.2), let us suppose, for contradiction, that there exists $(Y, Z) = \neq (Y, Z) \in M$ and $\alpha \in Y$ such that $l(\alpha) \geq (P + 1)l_{\text{max}}$. It follows that it is possible to write $\alpha = w_1 \ldots w_P w_{P+1} \xi, w_i \in C, \xi \in \text{Pref}(C^+)$. Further, let $G(\beta) = a_1 \ldots a_n$. One gets that there exists $w_j \ldots w_n \in F(G(\beta))$ such that

$$\beta = w_{j_1} \ldots w_{j_n} \alpha = w_{j_1} \ldots w_{j_n} w_1 \ldots w_P w_{P+1} \xi.$$

Then

$$G(\beta) = G(w_{j_1} \ldots w_{j_n} w_1 \ldots w_P w_{P+1} \xi) = a_{j_1} \ldots a_n$$

that contradicts (4).

Next, let $(Y, Z) \in M$ and let $Y = \{\alpha_1, \ldots, \alpha_n\}, Z = \{\gamma_1, \ldots, \gamma_m\}$. Let us assume

$$l(\alpha_1) < l(\alpha_2) < \cdots < l(\alpha_n), \quad l(\gamma_1) < l(\gamma_2) < \cdots < l(\gamma_m).$$

By definition of $Y$ and $Z$ one has that

$$\alpha_i$$ is a suffix of $\alpha_{i+1}, \quad i = 1, 2, \ldots, n - 1,$

$$\gamma_j$$ is a prefix of $\gamma_{j+1}, \quad j = 1, 2, \ldots, m - 1.$

Therefore, since $l(\alpha_n) \leq (P + 1)l_{\text{max}} - 1$ and $l(\gamma_m) \leq l_{\text{max}} - 1$, one obtains that

$$|Y| \leq (P + 1)l_{\text{max}} \quad (B.4)$$

$$|Z| \leq l_{\text{max}} - 1, \quad (B.5)$$
From (B.2)–(B.5) one finally gets that \( M \) is finite. In particular it is possible to show that

\[
|M| \leq \left[ N \frac{(2N)^{\lambda_{\text{max}} - 1} - 1}{2N - 1} + 1 \right] \left[ \frac{(2N)^{(P + 1)\lambda_{\text{max}} - 1}}{2N - 1} + 1 \right] - 1,
\]

(B.6)

where \( N \) is the cardinality of \( X \).

In a similar way it is possible to show that (B.6) holds for sets \( M_1 \) and \( M_2 \), too.

**References**


