Theoretical Computer Science 86 (1991) 365-375
Elsevier

Note

Decoders with initial state invariance for multivalued encodings*

Renato M. Capocelli

Dipartimento di Matematica G. Castelnuovo, Università di Roma "La Sapienza," 00185 Roma, Italy

Luisa Gargano and Ugo Vaccaro

Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84081 Baronissi (SA), Italy

Communicated by D. Perrin
Received June 1990
Revised January 1991

Abstract

Capocelli, R.M., L. Gargano and U. Vaccaro, Decoders with initial state invariance for multivalued encodings (Note), Theoretical Computer Science 86 (1991) 365-375.

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 we characterize the class of multivalued encodings that admit invariant decoders and provide an algorithm for constructing such decoders. Invariant decoders have the useful property that their behavior does not depend on the state in which they are, thus exhibiting optimal tolerance to accidental state transitions and/or errors in the input sequence.

1. Introduction

An encoding system is called *multivalued* if there may be two or more codewords corresponding to the same source symbol. In this paper we characterize the class of multivalued encodings that admit of *invariant decoders* and provide an algorithm for constructing such decoders.

* This work was supported in part by the Italian National Council for Research, under contract no. 90.01552.12.

Multivalued encodings arise in many practical situations. In particular, they appear very suitable for modeling the effect of noise during the transmission of data. As is well known, when a sequence of symbols is transmitted over a noisy channel, the corresponding channel output is not uniquely determined but depends both on the transmitted sequence and the error pattern that has occurred. Notice that if the channel allows *insertions* and *deletions* then the output sequences that correspond to an input sequence may have different lengths. The most general way to describe the behavior of a channel that suffers of insertions, deletions and *substitutions* errors is to specify, for each input symbol, all possible sequences that may occur at the output. This can be done by means of a multivalued encoding in which the set of codewords corresponding to a source symbol represents the noisy versions of the original encoding of that symbol. However, this approach can be practical only if the set of sequences associated with each source symbol does not become too large. Generally speaking, one can prevent this situation by ignoring all channel output sequences having small probability of occurrence. Another important situation that can be modeled successfully by means of a multivalued encoding is the *homophonic* channel. In the homophonic channel the set of different codewords that correspond to a source symbol represents the *homophons* into which that symbol is encoded. The technique of homophonic substitution is an old technique used in cryptology for converting an actual *plaintext* sequence in a (more) random sequence in order to increase the message entropy. Amongst the randomization techniques it seems by far the most adequate. It has been very recently reconsidered and enriched. In particular a complete information-theoretic treatment [9] and a general universal algorithm for homophonic encoding [8] have been provided. The multivalued encoding formalism would permit to characterize the deciphering and synchronizing properties of the homophonic substitution.

It should be recalled that multivalued encodings arise also in molecular biology. Indeed, in the biological code, several groups of bases may correspond to the same amino acid. This situation is described by saying that the biological code is *degenerate* (see [13] and [14] for a detailed discussion of this property of the biological code).

Multivalued encodings have been introduced by Sato [12] and further analyzed in [2–4, 6]. The construction of decoders was considered by Capocelli and Vaccaro [5] who gave three algorithms for constructing them. Next, Capocelli et al. [7] considered the problem of constructing self-synchronizing decoders for multivalued encodings, i.e., decoders able to recover synchronization, once it has been lost, in a bounded time interval. The decoders proposed in [7] have the interesting property of permitting to bound the incorrect decoding of the code message, in case a misfunctioning of the decoder itself or errors in the input sequence have moved the decoder into an incorrect state.

In this paper we consider decoders which satisfy a stronger property: their behavior is independent from the state in which they are. Therefore, they completely eliminate decoding errors due to random transitions from a state to another and/or errors in

the input sequence. We characterize the class of multivalued encodings which admit decoders with such a property and give an algorithm for constructing them. We remark that in case of ordinary encodings, our algorithm reduces to that by Levenshtein [11].

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 λ and X^n respectively denote the empty sequence and the nth concatenation of X with itself. 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 $p, q, s \in X^*$, if $pqs = w$ then p is a *prefix* of w, is a *suffix of* w and q is an *infix* of w. If p is a prefix of w write $w \geqslant p$ and if $p \neq w$ we say that p is a *proper prefix* of w.

Given a finite set A of source symbols, a multivalued encoding is a mapping $F: A \to 2^{X^+}$ 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 sequences 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 $y \in A$

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

For each sequence of source symbols $x \in A^*$, $F(x)$ denotes the set of all possible encodings of x. It is obvious that the above definition reduces to the definition of ordinary encoding when $F(a)$ is a singleton, for each $a \in A$. Finally, denote by C the set of all codewords and by C^+ the set of all code messages, i.e.,

$$C = \bigcup_{a \in A} F(a) \quad \text{and} \quad C^+ = \bigcup_{x \in A^+} F(x).$$

3. An algorithm for constructing invariant decoders

In this section we will provide a necessary and sufficient condition for the existence of an invariant decoder for a multivalued encoding and give an algorithm for constructing it. 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 is the state set;

s_0 is the initial state;

X is the input alphabet (= set of code letters);

A is the output alphabet (= set of source letters);
$f: S \times X \to S$ (= transition function);
$g: S \times X \to A^*$ (= output function).

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

Definition 3.1. The finite sequential machine D is a decoder for the multivalued encoding $F: A \to 2^{X^+}$ if and only if there exists an integer $t \geq 0$ such that for any $a_{i_1} a_{i_2} \ldots a_{i_k} \in A^+$, for any $w_{i_1} w_{i_2} \ldots w_{i_k} \in F(a_{i_1} a_{i_2} \ldots a_{i_k})$ and for all $\beta \in C^t$

$$g(s_0, w_{i_1} w_{i_2} \ldots w_{i_k} \beta) \geq a_{i_1} a_{i_2} \ldots a_{i_k}. \tag{1}$$

The smallest number t such that (1) holds is called the (decoding) delay of the decoder D.

In words, the meaning of Definition 3.1 is the following: The machine D is a decoder with delay t if and only if, having as input $k+t$ codewords, D is able to decode at least the first k codewords, leaving undeciphered at most t terminal codewords. Algorithms for constructing decoders for multivalued encodings have been given in [5].

Definition 3.2. Given a multivalued $F: A \to 2^{X^+}$, the decoder $D = (S, s_0, X, A, f, g)$ is called *invariant with respect to the initial state* (or simply an *invariant decoder*) if for each $s_j \in S$ the generalized sequential machine $D_j = (S, s_j, X, A, f, g)$ is a decoder for F.

Example 3.3. Let $A = \{0, 1\}$ be the set of source symbols, $X = \{a, b, c\}$ be the set of code symbols and consider the multivalued encoding $F(0) = \{cabb, ac, ab\}$, $F(1) = \{cbb\}$. A decoder for F that does not exhibit the invariance property is shown in Fig. 1. Indeed, $g(s_1, cbb) = 0$, whereas $cbb \in F(1)$. An invariant decoder for F is shown in Fig. 2.

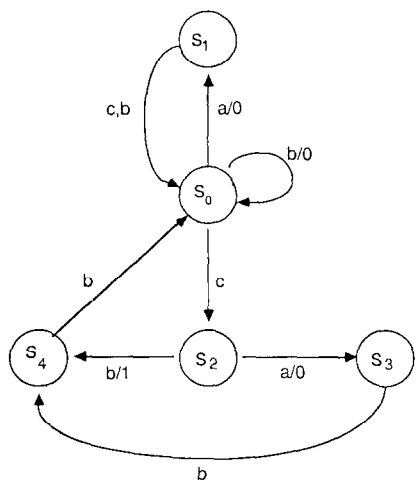

Fig. 1. Decoder for the multivalued encoding of Example 3.3.

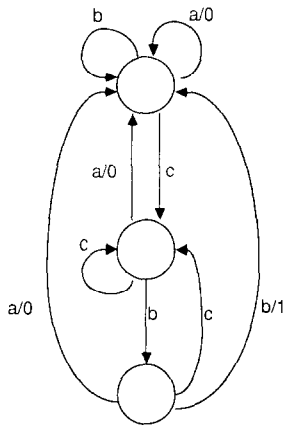

Fig. 2. Invariant decoder for the multivalued encoding of Example 3.3.

Let $F: A \to 2^{X^+}$ be a multivalued encoding. For each source symbol $a \in A$ and for each codeword $w \in F(a)$, denote by $p_1(w)$ the shortest prefix of w such that, for some integer $k \geq 0$, it holds that

$$w = p_1(w)\gamma_1 \ldots \gamma_k,$$

where each γ_i is a prefix, different from λ, of some codeword $w_i \in F(a_i)$, with $a_i \neq a$ for each i, $1 \leq i \leq k$. Denote by $P_1(C)$ the set

$$P_1(C) = \{p_1(w) \mid w \in C\}.$$

Now, for each $a \in A$, $w \in F(a)$ and for each integer $i \geq 2$, define recursively $p_i(w)$ as the shortest proper prefix of $p_{i-1}(w)$ when either there exist $u, v \in F(a)$ and $\xi_1, \xi_2 \in X^*$ with

$$p_i(w)v = p_{i-1}(w)\xi_1 p_{i-1}(u)\xi_2,$$

or there exist $c \in A$, $c \neq a$, and $v \in F(c)$ with $p_i(w)v = p_{i-1}(w)\gamma$, $\gamma \in X^*$; $p_i(w)$ is defined to be $p_{i-1}(w)$ otherwise. Denote by $P_i(C)$, for each integer $i \geq 2$, the set

$$P_i(C) = \{p_i(w) \mid w \in C\}.$$

Finally, define

$$P(C) = \{p(w) \mid w \in C\} = P_n(C),$$

where n is the smallest integer such that $P_n(C) = P_{n+1}(C)$. It is easy to see that if C is finite then there exists an integer n such that $P_n(C) = P_{n+1}(C)$.

Definition 3.4. A multivalued encoding $F: A \to 2^{X^+}$ is *fault-tolerant* if and only if the following two conditions are satisfied:

(a) for each source symbol $a \in A$ and for each codeword $w \in F(a)$ there do not exist $b \in A$ and $v \in F(b)$, with $a \neq b$, such that $p(v)$ is an infix of w;

(b) for each source symbol $a \in A$ there do not exist codewords, u, v, $w \in F(a)$ such that $p(u)\xi p(v)$ is an infix of w, for some $\xi \in X^*$.

The following example clarifies this definition:

Example 3.5. Let $A = \{0, 1\}$ be the set of source symbols and $X = \{a, b, c\}$ be the set of code symbols. The multivalued encoding defined by $F(0) = \{ac, ab\}$ and $F(1) = \{cbb, cabb\}$ is not fault-tolerant. Indeed

$$P_1(C) = \{a, ab, cbb, cabb\} = P_2(C) = P(C)$$

and then $p(ac) = a$, $cabb = cp(ac)bb$, with $ac \in F(0)$ and $cabb \in F(1)$. It follows that condition (a) of Definition 3.4 is not satisfied.

The multivalued encoding defined by $F(0) = \{cabcb, ab, cac\}$, $F(1) = \{bbc\}$ is fault-tolerant. Indeed

$$P_1(C) = \{cabc, a, cac, bb\}, \qquad P_2(C) = \{cab, a, ca, bb\} = P_3(C) = P(C)$$

and it is easy to see that both conditions (a) and (b) of Definition 3.4 are satisfied.

The following theorem states that a multivalued encoding admits an invariant decoder if and only if it is fault-tolerant. The sufficiency part of the theorem will also provide an algorithm for constructing such decoders.

Theorem 3.6. *Let $F: A \to 2^{X^+}$ be a multivalued encoding. A necessary and sufficient condition for an invariant decoder to exist is that F is fault-tolerant.*

Proof. (*Necessity*). Let $D = (S, s_0, X, A, f, g)$ be an invariant decoder for F. We first show that for each $s \in S$, $a \in A$ and $w \in F(a)$ it holds

$$g(s, w) = a. \tag{2}$$

Indeed, if t is the decoding delay of D, one has that for each $b \in A$, $v \in F(b)$ and $\gamma \in C^t$ it holds that

$$g(s, wv\gamma) \geqslant ab. \tag{3}$$

On the other hand, since D is an invariant decoder, it also results

$$g(s, wv\gamma) = g(s, w)g(f(s, w), v\gamma) \geqslant g(s, w)b. \tag{4}$$

From (3) and (4) one gets (2). Strengthening the above proof, it is possible to show that for each $s \in S$, $a \in A$ and $w \in F(a)$ it holds that

$$g(s, p(w)) = a. \tag{5}$$

The proof is by inductive argument. We shall prove that for each $s \in S$, $a \in A$, $w \in F(a)$ and $i \geqslant 1$ it holds that

$$g(s, p_i(w)) = a. \tag{6}$$

Let $i = 1$. By definition of $p_1(w)$ one gets

$$w = p_1(w)\gamma_1 \ldots \gamma_k$$

where each γ_i is a prefix, different from λ, of some codeword $w_j \in F(b_j)$, with $b_j \neq a$. Since $g(s, w) = g(s, p_1(w))\gamma_1 \ldots \gamma_k) = a$, in order to prove (6) it suffices to show that

$$g(f(s, p_1(w)\gamma_1 \ldots \gamma_{j-1}), \gamma_j) = \lambda, \quad \text{for each } 1 \leqslant j \leqslant k. \tag{7}$$

Suppose, by contradiction, that (7) is not true. From (2) one then has that

$$g(f(s, p_1(w)\gamma_1 \ldots \gamma_{j-1}), \gamma_j) = a, \tag{8}$$

for some $1 \leqslant j \leqslant k$. From the definition of γ_j, one gets that there exist $b_j \in A$, $b_j \neq a$, and $w_j \in F(b_j)$ such that $w_j = \gamma_j\gamma$, for some $\gamma \in X^*$. Moreover, from (2), one gets that

$$g(f(s, p_1(w)\gamma_1 \ldots \gamma_{j-1}), w_j) = b_j,$$

whereas, from (8) one obtains

$$g(f(s, p_1(w)\gamma_1 \ldots \gamma_{j-1}), w_j) \geqslant g(f(s, p_1(w)\gamma_1 \ldots \gamma_{j-1}), \gamma_j) = a$$

that contradicts the assumption that $a \neq b_j$. Thus (6) is true for $i = 1$. We now prove that if (6) is satisfied for $i-1$ it is also satisfied for i.

If $p_i(w) = p_{i-1}(w)$ then (6) it is trivially true. Assume $p_{i-1}(w) = p_i(w)\gamma$, for some $\gamma \in X^+$. It is possible to distinguish the following two situations:

Case (i): *There exist codewords* $u, v \in F(a)$ *such that*

$$p_i(w)v = p_{i-1}(w)\xi_1 p_{i-1}(u)\xi_2 = p_i(w)\gamma\xi_1 p_{i-1}(u)\xi_2, \quad \text{for some } \xi_1, \xi_2 \in X^*.$$

We shall show that $g(s, p_i(w)) = a$. Assume by contradiction that $g(s, p_i(w)) = \lambda$. From the inductive hypothesis we get that $g(f(s, p_i(w)), \gamma) = g(s, p_{i-1}(w)) = a$. Thus we obtain

$$g(f(s, p_i(w)), v) = g(f(s, p_i(w)), \gamma\xi_1 p_{i-1}(u)\xi_2) \geqslant aca, \quad \text{for some } c \in A^*.$$

On the other hand, from (2) one has $g(f(s, p_i(w)), v) = a$ which contradicts the above relation.

Case (ii): *There exist* $b \in A$, $b \neq a$, *and* $v \in F(b)$ *such that* $p_i(w)v = p_{i-1}\alpha$, *for some* $\alpha \in X^*$. Let $p_i(w)\gamma = p_{i-1}(w)$, for some $\gamma \in X^+$. Because of the inductive hypothesis, one has that $g(s, p_{i-1}(w)) = a$, so that in order to prove (6) it suffices to show that $g(f(s, p_i(w)), \gamma) = \lambda$. Suppose, by contradiction, that $g(f(s, p_i(w)), \gamma) = a$; one gets that

$$g(f(s, p_i(w)), v) \geqslant g(f(s, p_i(w)), \gamma) = a.$$

On the other hand, by definition of invariant decoder one has that $g(f(s, p_i(w)), v) = b \neq a$, this contradicts the assumption.

Therefore, we have proved (6) in all cases. Since for each $a \in A$ and $w \in F(a)$ there exists an integer n such that $p(w) = p_n(w)$, it follows that also (5) is true.

Using (5) we can prove that a necessary condition for a multivalued encoding to admit an invariant decoder is that the encoding is fault-tolerant. Suppose, by contradiction, that the encoding is not fault-tolerant. It is possible to distinguish the following two situations:

Case (1): *Condition* (*a*) *of Definition* 3.4 *is not satisfied.* This means that there exist source symbols $a, b \in A$, $a \neq b$, and codewords $w \in F(a)$ and $v \in F(b)$ such that

$$w = \beta_1 p(v) \beta_2, \quad \text{for some } \beta_1, \beta_2 \in X^*.$$

Let $s \in S$. Using (5) one gets

$$\begin{aligned} g(s, w) &= g(s, \beta_1 p(v) \beta_2) = g(s, \beta_1) g(f(s, \beta_1), p(v)) g(f(s, \beta_1 p(v)), \beta_2) \\ &= g(s, \beta_1) b g(f(s, \beta_1 p(v)), \beta_2). \end{aligned} \tag{9}$$

On the other hand, from (2) one has that $g(s, w) = a$, which contradicts (9).

Case (2): *Condition*(*b*) *of Definition* 3.4 *is not satisfied.* This means that there exists a source symbol $a \in A$ and codewords $u, v, w \in F(a)$ such that

$$w = \xi_1 p(u) \xi_2 p(v) \xi_3,$$

for some $\xi_1, \xi_2, \xi_3 \in X^*$. Let $s \in S$. Using (5) one gets

$$\begin{aligned} g(s, w) &= g(s, \xi_1 p(u) \xi_2 p(v) \xi_3) \\ &\geqslant g(s, \xi_1 p(u) \xi_2 p(v)) = g(s, \xi_1) a g(f(s, \xi_1 p(u)), \xi_2) a. \end{aligned} \tag{10}$$

On the other hand, from (2) it results $g(s, w) = a$ which contradicts (10).

Therefore, the property of fault-tolerance is a necessary condition for a multivalued encoding to admit of an invariant decoder.

(*Sufficiency*). Let $F: A \to 2^{X^+}$ be a fault-tolerant multivalued encoding. We shall give an algorithm for constructing an invariant decoder for F.

Let M be the set defined in the following way

$$\begin{aligned} M = \{ x \in X^* \mid \exists y \in P(C), \exists z \in X^+ \text{ such that } y = xz \\ \text{and } \forall \beta_1, \beta_2 \in X^*, \forall \alpha \in P(C) \text{ it holds that } x \neq \beta_1 \alpha \beta_2 \}, \end{aligned}$$

i.e., M is the set of all proper prefixes of elements in $P(C)$ that have no elements of $P(C)$ as infix.

For each $\beta \in X^+$ let $suff(\beta)$ denote the longest suffix of β that belongs to M. Define the sequential machine $D = (S, s_0, X, A, f, g)$ in the following way:

(a) $S = \{ s_y \mid y \in M \}$, $s_0 = s_\lambda$;

(b) for each $(s_y, b) \in S \times X$, the transition function f and the output function g are determined as follows:

$$f(s_y, b) = \begin{cases} s_\lambda & \text{if } yb \text{ has a suffix belonging to } P(C); \\ s_{suff(yb)} & \text{otherwise;} \end{cases}$$

$$g(s_y, b) = \begin{cases} a \in A & \text{if } yb = \beta p(w), \text{ for some } a \in A, w \in F(a), \beta \in X^*; \\ \lambda & \text{otherwise.} \end{cases}$$

This definition implies that the sequential machine D performs as follows: When it receives a string β of code symbols which has no infixes belonging to $P(C)$ then

$$f(s_\lambda, \beta) = s_{suff(\beta)}, \tag{11}$$

$$g(s_\lambda, \beta) = \lambda. \tag{12}$$

Moreover, if $b \in X$ is such that βb has a suffix $p(w) \in P(C)$, for some $w \in F(a)$ and $a \in A$, then

$$f(s_\lambda, \beta b) = f(s_{suff(\beta)}, b) = s_\lambda, \tag{13}$$

$$g(s_\lambda, \beta b) = g(s_{suff(\beta)}, b) = a. \tag{14}$$

Finally, if $\beta \in M$, then (11) becomes $f(s_\lambda, \beta) = s_{suff(\beta)} = s_\beta$.

In order to show that D is an invariant decoder for the multivalued encoding F, we shall prove that for any state $s_y \in S$, source symbol $a \in A$ and codeword $w \in F(a)$ it holds that

$$g(s_y, w) = a.$$

Let α be the shortest prefix of yw such that there exist $b \in A$ and $v \in F(b)$ for which $p(v)$ is a suffix of α. One can write

$$yw = \alpha\beta_2 = \beta_1 p(v)\beta_2, \quad \text{for some } \beta_1, \beta_2 \in X^*.$$

In order to show that $g(s_y, w) = a$, it is convenient to distinguish the following two situations:

Case (i): *$p(v)$ is an infix of w.* From the definition of fault-tolerant multivalued encoding, it results that also v belongs to $F(a)$. Let $w = \beta p(v)\beta_2$, for some $\beta \in X^*$. Since no infix of $y\beta$ belongs to $P(C)$, from (11) and (12) one gets

$$f(s_\lambda, y\beta) = s_{suff(y\beta)} \in S, \qquad g(s_\lambda, y\beta) = \lambda;$$

whereas, from (13) and (14) one gets

$$f(s_\lambda, y\beta p(v)) = s_\lambda = f(f(s_\lambda, y), \beta p(v)) = f(s_y, \beta p(v)),$$

$$g(s_\lambda, y\beta p(v)) = a = g(s_\lambda, y)g(f(s_\lambda, y), \beta p(v)) = g(s_y, \beta p(v)).$$

To prove that $g(s_y, w) = a$, it suffices to show that $g(s_\lambda, \beta_2) = \lambda$. From the definition of fault-tolerance it follows that no word $u \in C$ exists for which $w = \xi_1 p(v)\xi_2 p(u)\xi_3$, for any $\xi_1, \xi_2, \xi_3 \in X^*$. This implies that no element of $P(C)$ is infix of β_2. From (11) and (12) it results

$$f(s_\lambda, \beta_2) = s_{suff(\beta_2)} \in S, \qquad g(s_\lambda, \beta_2) = \lambda.$$

Case (ii): *$\beta_1 p(v) = y\eta$, for some $\eta \neq \lambda$ proper prefix of w.* From (11) and (12) one obtains

$$f(s_\lambda, y) = s_y, \qquad g(s_\lambda, y) = \lambda.$$

Moreover, it is possible to show that $v \in F(a)$. Indeed, assuming on the contrary that $v \in F(b)$ for some $b \neq a$, one has that $p(v)\beta_2 = \delta w$, for some δ proper prefix of $p(v)$, which contradicts the definition of $p(v)$. From (13) and (14) one gets

$$f(s_\lambda, \beta_1 p(v)) = s_\lambda, \qquad g(s_\lambda, \beta_1 p(v)) = a.$$

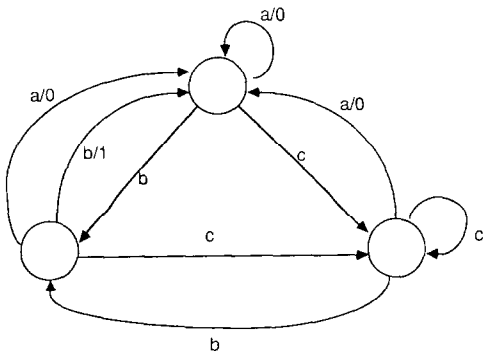

Fig. 3. Invariant decoder for the multivalued encoding of Example 3.7 obtained applying the algorithm of Theorem 3.6.

Moreover,

$$g(s_\lambda, \beta_1 p(v)\beta_2) = g(s_\lambda, yw) = g(f(s_\lambda, y), w) = g(s_y, w),$$

$$g(s_\lambda, \beta_1 p(v)\beta_2) = g(s_\lambda, \beta_1 p(v))g(f(s_\lambda, \beta_1 p(v)), \beta_2) = ag(s_\lambda, \beta_2),$$

that is $g(s_y, w) = ag(s_\lambda, \beta_2)$.

Finally, to show that $g(s_y, w) = a$, it suffices to prove that $g(s_\lambda, \beta_2) = \lambda$. Since the multivalued encoding is fault-tolerant, it is possible to see that, for any $b \in A$, $b \neq a$, no $u \in F(b)$ exists such that $p(u)$ is an infix of β_2. Otherwise, one would get that $p(u)$ is infix of $w \in F(a)$, which contradicts the fault-tolerance assumption. Suppose now that there exists $u \in F(a)$ such that $p(u)$ is an infix of β_2. It follows that

$$p(v)\beta_2 = \delta w = p(v)\xi_1 p(u)\xi_2$$

with δ proper prefix of $p(v)$, which contradicts the definition of $p(v)$. Since β_2 has no infix belonging to $P(C)$, from (11) and (12) one gets

$$f(s_\lambda, \beta_2) = s_{suff(\beta_2)} \in S, \qquad g(s_\lambda, \beta_2) = \lambda.$$

This completes the proof of the theorem. □

Example 3.7. Given $A = \{0, 1\}$, $X = \{a, b, c\}$, consider the multivalued encoding given by

$$F(0) = \{cabcb, ab, cab\}, \qquad F(1) = \{bbc\}.$$

One has that $C = \{cabcb, ab, cab, bbc\}$ and $P(C) = \{ca, a, ca, bb\}$ and $M = \{\lambda, c, b\}$. The invariant decoder for F obtained applying the algorithm presented in Theorem 3.6 is shown in Fig. 3.

References

[1] A.V. Aho and J.D. Ullman, *The Theory of Parsing, Translation and Compiling, Vol. 1: Parsing* (Prentice-Hall, Englewood Cliffs, NJ, 1972).

[2] R.M. Capocelli, A decision procedure for finite decipherability and synchronizability of multivalued encodings, *IEEE Trans. Inform. Theory* **28** (1982) 307-318.
[3] R.M. Capocelli and U. Vaccaro, Finite decipherability of multivalued encodings, in: *Proc. 21st Ann. Allerton Conf. on Communication, Control and Computing*, Urbana, IL (1983) 528-536.
[4] R.M. Capocelli, L. Gargano and U. Vaccaro, Synchronizability of multivalued encodings, in: *Proc. 10th Prague Conf. on Information Theory, Statistical Decision Functions and Random Processes* (1988) Vol. A, 225-234.
[5] R.M. Capocelli and U. Vaccaro, Structure of decoders for multivalued encodings, *Discrete Appl. Math.* **23** (1989) 55-71.
[6] R.M. Capocelli, L. Gargano and U. Vaccaro, A test for the unique decipherability of multivalued encodings, preprint, 1990.
[7] R.M. Capocelli, L. Gargano and U Vaccaro, Self-synchronizing decoders for multivalued encodings preprint 1990.
[8] C.G. Günther, A universal algorithm for homophonic coding, in: *Advances in Cryptology, Eurocrypt '88*, Lecture Notes in Computer Science (Springer, Berlin 1988) 405-414.
[9] H.N. Jendal, Y.J.B. Kuhn and J.L. Massey, An information-theoretic treatment of homophonic substitution, *Advances in Cryptology, Eurocrypt '89*, Lecture Notes in Computer Science (Springer, Berlin 1989) 382-394.
[10] V.I. Levensthein, Some properties of codings and self-adjusting automata for decoding messages, *Problemi Kibernetiki* **11** (1964) 63-121.
[11] V.I. Levensthein, Decoding automata with initial state invariance, in: *Proc. All Union Conf. on Theory of Coding and its Applications*, Odessa (1963) 125-136.
[12] K. Sato, A decision procedure for the unique decipherability of multivalued encodings, *IEEE Trans. Inform. Theory* **25** (1979) 356-360.
[13] L. Stryer, *Biochemistry* (Freeman, San Francisco 1975).
[14] M. Yčas, *The Biological Code* (North-Holland, Amsterdam, 1969).