On Special Families of Morphisms Related to δ-Matching and Don't Care Symbols

Richard Cole

Department of Computer Science, Courant Institute, New York University.
`cole@cs.nyu.edu`

Costas Iliopoulos [1]

Department of Computer Science, King's College London, UK.
`csi@dcs.kcl.ac.uk`

Thierry Lecroq [2]

LIFAR-ABISS, Université de Rouen, France. `Thierry.Lecroq@univ-rouen.fr`

Wojciech Plandowski

Instytut Informatyki, Warsaw University, Poland. `wojtekpl@mimuw.edu.pl`

Wojciech Rytter

Instytut Informatyki, Warsaw University, Poland, and Department of Computer Science, University of Liverpool, UK. `W.Rytter@mimuw.edu.pl`

Abstract

The δ-matching problem is a special version of approximate pattern-matching, motivated by applications in musical information retrieval, where the alphabet Σ is an interval of integers. We investigate relations between δ-matching and pattern-matching with don't care symbol $*$ (a symbol matching every symbol, including itself). We show that the δ-matching is reducible to k instances of pattern-matching with *don't cares*. We investigate how the numbers δ and k are related by introducing δ-distinguishing families $\mathcal{H}$ of morphisms. The size of $\mathcal{H}$ corresponds to k. We show that for minimal families $\mathcal{H}$ we have $|\mathcal{H}| = \Theta(\delta)$.

Key words: combinatorial problems, pattern-matching, δ-matching, don't care symbols

1 Introduction

The δ-matching problem is a special version of approximate pattern-matching, motivated by applications in musical information retrieval, where the alphabet Σ is an interval of integers (see [1,7]). A δ-match of a pattern P of length m in a text T is a position j in T such $|P[i+j-1]-T[j] \leq \delta$ for $1 \leq i \leq m$. Algorithms for solving this problem have been presented in [2,4,3]. We investigate relations between δ-matching and pattern-matching with don't care symbol $*$ (a symbol matching every symbol, including itself). We show a close correspondence between pattern-matching with *don't cares* and δ-matching. The δ-matching is reducible to k instances of pattern-matching with *don't cares*. We investigate how the numbers δ and k are related by introducing δ-distinguishing families $\mathcal{H}$ of morphisms. The size of $\mathcal{H}$ corresponds to k. We show that for minimal families $\mathcal{H}$ we have $|\mathcal{H}| = \Theta(\delta)$.

A similar problem has been also considered in [6] as the *subset matching*, where subsets are very specials: intervals of integers. In this paper we provide a novel approach to δ-matching through families $\mathcal{H}$ of morphisms, this gives considerably simpler algorithms, compared with [6].

We also contribute to the combinatorics on texts, in particular combinatorics of morphisms, by providing several bounds on $|\mathcal{H}|$.

This article is organized as follows: Section 2 introduces the basic notions on δ-matching and matching with *don't cares*. Section 3 is devoted to δ-distinguishing families of morphisms. In Section 4 we present mixed families of morphisms. We give final remarks in Section 5.

2 Basic notions

Assume the alphabet Σ is a set of integers $\Sigma = [1, 2, \ldots, s]$ and δ is an integer. For $a, b \in \Sigma$ we write $a \stackrel{\delta}{=} b$ if and only if $|a - b| \leq \delta$.

If u, w are two strings of same length over Σ then $u \stackrel{\delta}{=} w$ if and only if $u[i] \stackrel{\delta}{=} w[i]$ for each position i in u. For a pattern P of size m and text T of size n a δ-*match* is any position $1 \leq j \leq n - m + 1$ such that $P \stackrel{\delta}{=} T[j \ldots j + m - 1]$.

[1] The work of this author was partially supported by a NATO grant PST.CLG.977017, a Marie Curie Fellowship, a Royal Society and Welcome Foundation grants.
[2] The work of this author was partially supported by a NATO grant PST.CLG.977017.

The problem of δ-*matching* consists in finding all δ-matches of P in T. The related problem is that of matching with *don't care* symbol $*$. The *don't care* symbol matches any symbol (including itself). We write $a \approx b$ if and only if $a = b$ or $a = *$ or $b = *$. For two strings of same length $u \approx w$ if and only if $u[i] \approx w[i]$. The problem of *pattern-matching with don't cares* consists in finding all positions j such that $P \approx T[j \ldots j+m-1]$. Denote by $IntMult(n)$ the time to multiply two n-bit binary numbers. The following fact has been shown by Fischer and Paterson, see [8,5].

Lemma 1 *The problem of pattern-matching with* don't cares *for a pattern P and a text T of length n over an alphabet Σ can be solved in time $O(\log|\Sigma| \times IntMult(n))$.*

We show that for small alphabet δ-matching is at least as difficult as matching with *don't cares.*

Theorem 2 *For binary alphabets $\{a, b\}$ the string-matching with* don't cares *is reducible in linear time to δ-matching for the alphabet $\Sigma = [1, 2, 3]$.*

PROOF. Replace a by 1, b by 3 and don't care symbol $*$ by 2 in an instance of string-matching with *don't cares.* Take $\delta = 1$. Then the symbol 2 "plays" the role of the *don't care* symbol, and the problem is reduced to δ-matching with $\delta = 1$. $\square$

The reduction of δ-matching to pattern matching with *don't cares* is by reducing the δ-matching problem for a given pattern P and text T to several pattern-matchings with *don't cares*: searching for $h_i(P)$ in $h_i(T)$ for several simple symbol-to-symbol encodings $h_1, h_2, \ldots, h_k$. We investigate the bounds on the number k, it corresponds to the question: how many instances of the pattern-matching with don't cares are needed to solve the δ-matching problem.

3 δ-Distinguishing Families of Morphisms

Let $\mathcal{H} = \{h_1, h_2, \ldots, h_k\}$ be a family of morphisms $h_i : \Sigma \to \Sigma_i \cup \{*\}$.

We say that $\mathcal{H}$ is δ-*distinguishing* if and only if for every $(a, b \in \Sigma)$

$$[a \stackrel{\delta}{=} b] \equiv [\forall (h \in \mathcal{H}) h(a) \approx h(b)] \,.$$

We can view the morphism presented linearly as $h(123 \ldots |\Sigma|)$. The family $\mathcal{H}$ is also treated, throughout the paper, as a $k \times s$ table, the i-th row corresponds

to the i-th morphism, viewed in its linear form. The symbols of Σ corresponds to columns in the table and to positions in the rows (in the morphisms). We usually use later the terminology "position" which corresponds naturally to a "symbol", since each symbol is an integer in the range $[1..|\Sigma|]$.

Denote by $\mathcal{M}_\delta(P,T)$ the set of starting positions of δ-occurrences of P in T. Similarly define by $\mathcal{D}(P,T)$ the set of matches with *don't care* symbol. Formally:

$$\mathcal{M}_\delta(P,T) = \{j \mid P \stackrel{\delta}{=} T[j \ldots j+m-1]\},$$

$$\mathcal{D}(P,T) = \{j \mid P \approx T[j \ldots j+m-1]\}.$$

It follows from definitions that if $\mathcal{H} = \{h_1, h_2, \ldots, h_k\}$ is δ-*distinguishing*, then
$\mathcal{M}_\delta(P,T) = \mathcal{D}(h_1(P), h_1(T)) \cap \mathcal{D}(h_2(P), h_2(T)) \cap \ldots \cap \mathcal{D}(h_k(P), h_k(T))$.

The δ-matching is now reduced to $|\mathcal{H}|$ instances of the string-matching with *don't care*. For each $h \in \mathcal{H}$ we solve an instance of string-matching with *don't cares*: check if $h(P)$ occurs in $h(T)$ (forgetting about δ).

There is a δ-match of P starting at position j in T if and only if there is a don't-care-match of $h(P)$ in $h(T)$ at position j for each $h \in \mathcal{H}$.

Example 1

Let $\Sigma = [1,2,3,4]$ and $\delta = 1$. Take the family $\mathcal{H}_1$ of 2 morphisms:
$h_1 : 1 \to 1, 2 \to 1, 3 \to *, 4 \to 2,$ $\qquad h_2 : 1 \to 1, 2 \to *, 3 \to 2, 4 \to 2.$

If we write morphisms as string $h(1) \cdot h(2) \cdots h(s)$, then we can write h_1 and h_2 as $h_1 = 11*2$ and $h_2 = 1*22$. The graphical representation of morphisms h_1 and h_2 is:

Σ	1	2	3	4
h_1	1	1	*	2
h_2	1	*	2	2

The family $\mathcal{H}_1$ is 1-*distinguishing* for the alphabet $\Sigma = [1,2,3,4]$.

Figure 1 illustrates how δ-matching is related to two instances of *don't care* matching for an example pattern $P = 1432$ and text $T = 233423431$. We have: $h_1(T) = 1**21*2*1$, $h_1(P) = 12*1$, $h_2(T) = *222*2221$ and $h_2(P) = 122*$.

Let $\mathcal{H} = \{h_1, h_2, \ldots, h_k\}$ be a family of morphisms. We say that $\mathcal{H}$ is a δ-regular family if and only if it satisfies the following properties:

P1 : each morphism h is of a form

T	2	3	3	4	2	3	4	3	1	T	1	*	*	2	1	*	2	*	1	T	*	2	2	2	*	2	2	1	1
P					1	4	3	2		P		1	2	*	1					P	1	2	2	*				
										P			1	2	*	1				P				1	2	2	*	
										P					1	2	*	1										
										P						1	2	*	1									

Fig. 1. There is only one 1-match of the pattern $P = 1432$ in the text $T = 233423431$ since $\{5\}$ is the intersection of sets of starting occurrences with don't cares of $h_1(P)$ in $h_1(T)$ and $h_2(P)$ in $h_2(T)$. Formally: $\mathcal{D}(12*1, 1**21*2*1) = \{2,3,5,6\}$, $\mathcal{D}(122*, *222*2221) = \{1,5\}$, $\mathcal{M}_\delta(P,T) = \{2,3,5,6\} \cap \{1,5\} = \{5\}$.

$h = \ **\ldots*11\ldots1**\ldots*22\ldots2**\ldots*33\ldots3**\ldots*44\ldots$
The internal blocks of $*$'s are exactly of length δ, the boundary blocks of $*$'s are of length at most δ.

P2 : If $q - p > \delta$ then for some i $h_i(p) \neq *$, $h_i(q) \neq *$, and $h_i(r) = *$ for some $p < r < q$.

The family $\mathcal{H}_1$ from Example 1, as well as the family in Figure 1 are δ-regular. The following lemma gives the relationship between δ-regular and δ-distinguishing.

Lemma 3 *If a family $\mathcal{H}$ is δ-regular then it is a δ-distinguishing family.*

PROOF. Let $p, q \in \Sigma$ be two different symbols such that $p < q$. Two cases are two be considered whether $p \overset{\delta}{=} q$ or not.

- If $p \overset{\delta}{=} q$ then $q - p \leq \delta$. For each $h \in \mathcal{H}$, if $h(p) = *$ then $h(p) \approx h(q)$. If $h(p) = i \neq *$ then either $h(q) = i$ or $h(q) = *$ thus $h(q) \approx h(q)$.
- If $p \overset{\delta}{\neq} q$ then $q - p > \delta$. By property P2 there exists $h \in \mathcal{H}$ such that $h(p) = i$ and $h(q) = i + 1$ thus $h(p) \not\approx h(q)$.

In both cases if $\mathcal{H}$ is δ-regular then it is δ-distinguishing. $\square$

Denote by $\alpha(\delta, s)$ the size of a minimal δ-*distinguishing* family of morphisms for the alphabet Σ of size s. Let $\alpha(\delta) = \max_s \alpha(\delta, s)$.

Theorem 4 *The size of a minimal δ-distinguishing family of morphisms is at most $2\delta + 1$: $\alpha(\delta) \leq 2\delta + 1$.*

PROOF. The structure of the morphisms, for $\delta = 3$, is illustrated in Figure 2.

Σ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
h_1	*	*	*	1	1	1	1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	4	4
h_2	1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	*	*	*	5	5	5	5	5
h_3	1	1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	*	*	*	5	5	5	5
h_4	1	1	1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	*	*	*	5	5	5
h_5	1	1	1	1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	*	*	*	5	5
h_6	*	1	1	1	1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	*	*	*	5
h_7	*	*	1	1	1	1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	*	*	*

Fig. 2. The structure of 7 morphisms (rows) constituting a 3-*distinguishing* family. For instance $14-12 \leq \delta = 3$ and $h_i(12) \approx h_i(14)$ for $1 \leq i \leq 7$ and $16-12 > \delta$ and $h_6(12) \not\approx h_6(16)$.

The first morphism, written as a linear array is:

$$h = **\ldots*11\ldots1**\ldots*22\ldots2**\ldots*33\ldots3**\ldots*,$$

where we take groups of δ stars, and the symbols between stars are $\delta+1$ consecutive integers (the same for each group of non-stars). The next 2δ morphisms result by shifting h_1 by $1, 2, \ldots, 2\delta$ places to the right. Obviously the family satisfies property P1. It satisfies also the property P2 due to the following useful properties:

(1) the number of $*$'s in each column is at most δ;

(2) if $p - q > \delta$ then for every i there is a symbol $*$ between positions p, q in h_i.

Due to property (1) if we take two columns p, q then there should be a row i which contains symbols different from $*$ at columns p and q. Then, due to property (2), the property P2 is satisfied by the morphism h_i. This completes the proof. $\square$

Example 2

For $\delta = 3$ and $s = 30$ we can take the family which consists of 7 morphisms whose structure is illustrated in Figure 2. Each row is a table of morphism, the stars are *don't cares*, and each maximal consecutive group of non-stars gets the next integer number. The morphism are presented by linear arrays, the i-th symbol is the morphic value of the input symbol i.

We now show that $\alpha(3, 30) \leq 7$ and generally $\alpha(3) \leq 7$, since the construction is periodic and works for arbitrarily large alphabet Σ.

Lemma 5 *The size of a minimal* 3-*distinguishing family of morphisms is at most* 6*:* $\alpha(3) \leq 6$.

h_1				*	*	*								*	*	*						*	*	*								*	*	*					
h_2					*	*	*		*	*	*		*	*	*			*	*	*			*	*	*		*	*	*		*	*	*			*	*	*	
h_3						*	*	*			*	*	*		*	*	*		*	*	*				*	*	*			*	*	*		*	*	*		*	*	*
h_4		*		*	*		*	*	*			*	*	*			*	*	*		*	*	*		*	*	*			*	*	*			*	*	*		*
h_5								*	*	*						*	*	*								*	*	*					*	*	*			
h_6	*	*		*						*	*	*								*	*	*						*	*	*						*	*

period period

Fig. 3. The periodic structure of 6 morphisms (rows) representing together the 3-distinguishing family of morphisms. Only *don't care* symbols are shown.

PROOF. The construction of 6 morphisms for $\delta = 3$ is illustrated in Figure 3. The resulting family is δ-regular.

The sets of *stars* in each column have been selected from the family:

$\mathcal{F} = [\{1,2,3\}, \{1,2,4\}, \{1,3,5\}, \{1,4,6\}, \{2,3,4\}, \{2,3,6\}, \{2,4,5\}, \{2,5,6\}, \{3,4,5\}, \{3,4,6\}]$.

It easy to see that for each $X \in \mathcal{F}$ the complement of X is not in $\mathcal{F}$. Hence $\mathcal{F}$ has the intersecting property: any two sets have a nonempty intersection. This implies that for any two columns p, q there is a row i such that $h_i(p) \neq *$ and $h_i(q) \neq *$.

It can be checked that each two columns at distance at least 4 are distinguished by non-star entries. The construction is periodic, hence it works for arbitrarily large alphabet. $\square$

Theorem 6 *If k is divisible by 3, then $\alpha(\delta) \leq 2\delta$.*

PROOF. Due to Lemma 5 it is enough to prove that $\alpha(r \cdot \delta) \leq r \times \alpha(\delta)$. We use linear representation of morphisms. Assume $\{h_1, h_2, \ldots, h_k\}$ is a δ-distinguishing family, where each h_i is identified with $h_i(123\ldots|\Sigma|)$. We receive a $r\delta$-distinguishing family by first replacing each symbol x in each of h_i by r copies of x. Then each of the obtained linear representations of morphisms is cyclically shifted by j, $1 \leq j < r$. In this way we obtain $k \times r$ morphisms, except that their linear representations could be too long. In this case each resulting morphism is cut at the end to have its linear representation exactly of length of the alphabet. $\square$

Theorem 7 *The size of a minimal δ-distinguishing family of morphisms is at least $\delta + 2$: $\alpha(\delta) \geq \delta + 2$.*

PROOF. It is easier to deal with δ-regular families.

Claim. If there is a δ-distinguishing family of size k then there exists a δ-regular family of k morphisms.

Assume $\mathcal{H}$ is A δ-distinguishing family $\mathcal{H}$ is converted into a δ-regular one of the same size. We convert individually each morphism. We demonstrate it for $\delta = 3$. If we have a block with too many stars, for example $a * * * * * * b$ is converted into $aaaa * * * b$, if we have a block with less than δ stars, for example $a * * b$, we convert it into $aaab$ (observe that in this case it should be $a = b$). Once we have blocks of $*$'s of "good" size we change other symbols to groups of 1's, 2's etc. In this way we obtain a δ-regular family which is δ-distinguishing, according to Lemma 3. This completes the proof of the claim.

We go now to the main part of the proof, which we carry for $\delta = 3$, without loss of generality. There should be a morphism (a row) which distinguishes between positions 1 and 5, so one of the rows, say h_1, should start with the sequence $h_1 = 1 * * * 2 \ldots$ We should distinguish between 2 and 6, so there should be another morphism, say h_2, which is of a form $h_2 = x1 * * * 2 \ldots$ Continuing in this manner we should have two other morphisms of the form: $h_3 = xx1 * * * 2 \ldots$, $h_4 = xxx1 * * * 2 \ldots$, where x is a symbol (integer) or $*$.

We show, by contradiction, that we need one extra morphism at least. Assume we have only the morphisms $h_1, \ldots, h_4$ as above, we know already how they start and now we can force them, using the same arguments, to have the following structure:

Σ	1	2	3	4	5	6	7	8	9	10	11	12
h_1		*	*	*		*	*	*				
h_2			*	*	*		*	*	*			
h_3				*	*	*		*	*	*		
h_4					*	*	*		*	*	*	

However now the symbols (positions) 4 and 9 are not distinguishable by the morphism. This contradicts the assumption that the family is δ-distinguishing. Hence we need 5 morphisms for $\delta = 3$. The same argument works for any δ to show that $\delta + 2$ morphisms are needed. This completes the proof of Theorem 7. $\square$

The next result is a consequence of Theorems 4 and 7.

Corollary 8 *For each $\delta \geq 1$ we have: $\delta + 2 \leq \alpha(\delta) \leq 2\delta + 1$. For $\delta \in \{1, 2\}$ we have $\alpha(\delta) = 2\delta + 1$.*

4 Mixed Families of Morphisms

For the string matching with *don't cares* the size of the alphabet is relevant. This motivates the introduction of two separate families: arbitrary morphisms, those with many output symbols, and binary morphisms, those with only two output symbols (plus the *don't care*). The latter morphisms, as well as related families of morphisms, are called *binary*.

Theorem 9 *For each $\delta \geq 1$ there are two families $\mathcal{H}_1$ and $\mathcal{H}_2$ of morphism such that*

(1) The family $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ is δ-distinguishing;
(2) $\mathcal{H}_2$ consists only of binary morphisms;
(3) $|\mathcal{H}_1| = 3$ and $|\mathcal{H}_2| = 2\delta + 1$.

PROOF. Take $\mathcal{H}_1 = \{h_1, h_2, h_3\}$, where:

$$\begin{aligned} h_1 &= 1^{2\delta} \cdot *^{\delta} \cdot 2^{2\delta} \cdot *^{\delta} \cdot 3^{2\delta} \cdots \\ h_2 &= *^{\delta} \cdot 1^{2\delta} \cdot *^{\delta} \cdot 2^{2\delta} \cdot *^{\delta} \cdots \\ h_3 &= 1^{\delta} \cdot *^{\delta} \cdot 2^{2\delta} \cdot *^{\delta} \cdot 3^{2\delta} \cdot *^{\delta} \cdots \end{aligned}$$

Additionally we take $2\delta + 1$ morphisms whose structure is the same as that used in the proof of Theorem 4, see Figure 2. The only difference is that we replace groups of non-star entries alternately by a and b. We obtain a family of binary morphism, which for positions at distance at most 2δ "behaves" in the same way as the family from Theorem 4. $\square$

As a corollary of Theorem 9 we have the following fact.

Theorem 10 *The problem of δ-matching with the input alphabet Σ can be solved in time $O((\delta + \log|\Sigma|) \cdot IntMult(n))$.*

Example 3.

For $\delta = 3$ and $s = 30$ we take the family $\mathcal{H}_1$ given by:

$$\begin{aligned} h_1 &= 1\ 1\ 1\ 1\ 1\ 1\ *\ *\ *\ 2\ 2\ 2\ 2\ 2\ 2\ *\ *\ *\ 3\ 3\ 3\ 3\ 3\ 3\ *\ *\ *\ 4\ 4\ 4 \\ h_2 &= *\ *\ *\ 1\ 1\ 1\ 1\ 1\ 1\ *\ *\ *\ 2\ 2\ 2\ 2\ 2\ 2\ *\ *\ *\ 3\ 3\ 3\ 3\ 3\ 3\ *\ *\ * \\ h_3 &= 1\ 1\ 1\ *\ *\ *\ 2\ 2\ 2\ 2\ 2\ 2\ *\ *\ *\ 3\ 3\ 3\ 3\ 3\ 3\ *\ *\ *\ 4\ 4\ 4\ 4\ 4\ 4 \end{aligned}$$

The family $\mathcal{H}_2$ in this case consists of 7 ($= 2\delta + 1$) binary morphisms:

$h_1 = * * * 1 1 1 1 * * * 2 2 2 2 * * * 1 1 1 1 * * * 2 2 2 2 * *$
$h_2 = 1 * * * 2 2 2 2 * * * 1 1 1 1 * * * 2 2 2 2 * * * 1 1 1 1 *$
$h_3 = 1 1 * * * 2 2 2 2 * * * 1 1 1 1 * * * 2 2 2 2 * * * 1 1 1 1$
$h_4 = 1 1 1 * * * 2 2 2 2 * * * 1 1 1 1 * * * 2 2 2 2 * * * 1 1 1$
$h_5 = 1 1 1 1 * * * 2 2 2 2 * * * 1 1 1 1 * * * 2 2 2 2 * * * 1 1$
$h_6 = * 1 1 1 1 * * * 2 2 2 2 * * * 1 1 1 1 * * * 2 2 2 2 * * * 1$
$h_7 = * * 1 1 1 1 * * * 2 2 2 2 * * * 1 1 1 1 * * * 2 2 2 2 * * *$

5 Final Remarks

The δ-matching problem is a very special instance of approximate string-matching, and its algorithmic complexity is still not well understood. We contribute to the algorithmics of this problem by introducing a novel approach. We have introduced in this paper δ-distinguishing families of morphisms which constitute a useful tool in the transformation of δ-matching to the pattern-matching with *don't cares*. This gives simple and efficient algorithms for δ-matching using the integer multiplication procedure as a *black-box*. The crucial parameter is the cardinality of such minimal families, denoted by $\alpha(\delta)$. We have shown that $\alpha(\delta)$ is of the same order as δ: $\delta + 2 \leq \alpha(\delta) \leq 2\delta + 1$.

This gives exact values: $\alpha(1) = 3$ and $\alpha(2) = 5$. For $\delta = 3$ we have $5 \leq \alpha(3) \leq 6$, due to the fact, shown in the paper, that $\alpha(3r) \leq 6r$. The exact formula for $\alpha(\delta)$ remains as an open problem. A simpler problem is to compute the exact value of $\alpha(4)$. From the practical point of view more important is the cardinality of the family of binary morphisms. We have shown that there is always a δ-distinguishing family consisting of 3 morphisms with large alphabet (but at most $|\Sigma|$) and $2\delta + 1$ binary morphisms. For δ's which are multiple of 3 this can improved to 2δ.

References

[1] E. Cambouropoulos, T. Crawford and C.S. Iliopoulos, Pattern Processing in Melodic Sequences: Challenges, Caveats and Prospects, in: G. Wiggins, ed., *Proceedings of the Artificial Intelligence and Simulation of Behaviour Symposium* (Edinburgh, UK, 1999) 42–47.

[2] E. Cambouropoulos, M. Crochemore, C. S. Iliopoulos, L. Mouchard and Y. J. Pinzon, Algorithms for computing approximate repetitions in musical sequences, in: R. Raman and J. Simpson, eds., *Proceedings of the 10th Australasian Workshop On Combinatorial Algorithms* (Perth, WA, Australia, 1999) 129–144.

[3] M. Crochemore, C.S. Iliopoulos, T. Lecroq, W. Plandowski and W. Rytter, Three Heuristics for delta-matching: delta-BM algorithms, in: A. Apostolico

and M. Takeda, eds., *Proceedings 13th Annual Symposium on Combinatorial Pattern Matching* (Fukuoka, Japan, 2002) LNCS 2373 Springer-Verlag, Berlin, 178-189.

[4] M. Crochemore, C.S. Iliopoulos, T. Lecroq and Y.J. Pinzon, Approximate string matching in musical sequences in: M. Balík and M. Simánek, eds., *Proceedings of the Prague Stringology Conference* (Prague, Tcheque Republic, 2001) Annual Report DC-2001-06, 26–39.

[5] M. Crochemore, W.Rytter, *Text algorithms*, (Oxford Press, 1994).

[6] R. Cole, R. Hariharan, P. Indyk, Tree pattern-matching and subset matching in $O(n \log^3 n)$ time, in *Proceedings 10th Annual ACM-SIAM Symposium on Discrete Algorithms* (Baltimore, Maryland, 1999) 245–254.

[7] T. Crawford, C.S. Iliopoulos, R. Raman, String matching techniques for musical similarity and melodic recognition, *Computing in Musicology*, **11** (1998) 73–100.

[8] M.J. Fischer, M.S. Paterson, String-matching and other products, in: R. Karp, ed., *Complexity of Computation: Proceedings of a Symposium in Applied Mathematics of the AMS and the SIAM*, (1974) 113–125.