Relations Between δ-Matching and Matching with Don't Care Symbols: δ-distinguishing Morphisms

(Preliminary Version)

Richard Cole,[1] ⋆ Costas S. Iliopoulos,[2] ⋆⋆ Thierry Lecroq,[3] ⋆⋆⋆ Wojciech Plandowski,[4] and Wojciech Rytter[5] †

[1] Computer Science Department, Courant Institute of Mathematical Sciences, New York University,
251 Mercer Street, New York, NY 10012-1185, U.S.A.
`cole@cs.nyu.edu`
`http://www.cs.nyu.edu/cs/faculty/cole`

[2] Dept. Computer Science, King's College London, London WC2R 2LS, England, and School of Computing, Curtin University of Technology, GPO Box 1987 U, WA.
`csi@dcs.kcl.ac.uk`
`www.dcs.kcl.ac.uk/staff/csi`

[3] LIFAR–ABISS, Faculté des Sciences et Techniques, Université de Rouen, 76821 Mont-Saint-Aignan Cedex, France
`Thierry.Lecroq@univ-rouen.fr`,
`http://www-igm.univ-mlv.fr/~lecroq`.

[4] Instytut Informatyki, Warsaw University, Banacha 2, 02–097, Warszawa, Poland
`W.Plandowski@mimuw.edu.pl`
`http://www.mimuw.edu.pl/~wojtekpl`

[5] Instytut Informatyki, Warsaw University, Banacha 2, 02–097, Warszawa, Poland, and Department of Computer Science, University of Liverpool,L69 7ZF, Liverpool, UK
`W.Rytter@mimuw.edu.pl`
`http://www.mimuw.edu.pl/~rytter`

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. The δ-match is an occurrence of the pattern such that $|a-b| \le \delta$ for each pair of letters (integers) whose positions match in this occurrence. 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 $|\mathcal{H}| = \Theta(\delta)$.

⋆ This work was supported in part by NSF grant CCR-9800085.

⋆⋆ The work was partially supported by a NATO grant PST.CLG.977017, a Marie Curie Fellowship, a Royal Society and a Wellcome Foundation grant.

⋆⋆⋆ The work of this author was partially supported by a NATO grant PST.CLG.977017.

† Partially supported by a Royal Society grant.

1 Introduction

The *δ-matching* problem is a version of *approximate string-matching* motivated by information retrieval in musical data, see [5]. It has been also considered in [4] as *subset matching*, where subsets were very special: 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 [4]. We also contribute to combinatorics on texts, in particular combinatorics of morphisms, by providing several upper and lower bounds on $\mathcal{H}$.

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

$$a \overset{\delta}{=} b \text{ iff } |a - b| \leq \delta.$$

If u, w are two strings of the same length over Σ then

$$u \overset{\delta}{=} w \text{ iff } u[i] \overset{\delta}{=} w[i] \text{ for each position } i \text{ in } u.$$

For a pattern P of size m and a text T of size n a *δ-match* is any position $1 \leq i \leq n-m+1$ such that

$$P \overset{\delta}{=} T[i \ldots i+m-1].$$

The problem of *δ-matching* consists of 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 $a \approx b$ iff $a = b$ or $a = *$ or $b = *$. For two strings of the same length $u \approx w$ iff $u[i] \approx w[i]$ for each position i. The problem of *pattern-matching with don't cares* consists of finding all positions i such that $P \approx T[i \ldots i+m-1]$. Denote by $IntMult(n)$ the time to multiply two n-bit binary numbers. The following fact has been shown by Fischer and Paterson, see [6, 3].

Lemma 1. *The problem of pattern-matching with don't cares for strings over an alphabet Σ can be solved in time $O(\log|\Sigma| \cdot IntMult(n))$.*

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

Theorem 1. *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 dont'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 $\delta = 1$.

2 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* iff for every $(a, b \in \Sigma)$

$$[a \overset{\delta}{=} b] \;\equiv\; [\forall\ (h \in \mathcal{H})\ h(a) \approx h(b)]$$

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. Similarly, 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:

$$\mathcal{M}_\delta(P,T) \;=\; \{i \;:\; P \overset{\delta}{=} T[i \ldots i+m-1]\}$$
$$\mathcal{D}(P,T) \;=\; \{i \;:\; P \approx T[i \ldots i+m-1]\}$$

Theorem 2. *If* $\mathcal{H} \;=\; \{h_1, h_2, \ldots, h_k\}$ *is δ-distinguishing, then*

$$\mathcal{M}_\delta(P,T) \;=\; \mathcal{D}(h_1(P), h_1(T)) \;\cap\; \mathcal{D}(h_2(P), h_2(T)) \;\cap\; \ldots \mathcal{D}(h_k(P), h_k(T))$$

For a given pattern P and text 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 i in T iff there is a don't-care-match of $h(P)$ in $h(T)$ at position i 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:

$$h_1 :\; 1 \to a,\; 2 \to a,\; 3 \to *,\; 4 \to b,$$
$$h_2 :\; 1 \to a,\; 2 \to *,\; 3 \to b,\; 4 \to b,$$

We also write morphisms as the string $h(1) \cdot h(2) \cdots h(s)$; in this way we can write $h_1,\ h_2$ as $\quad h_1 \;=\; aa * b, \quad h_2 \;=\; a * bb$

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 the pattern

$P \;=\; 1432$ and the text $T \;= 233423431$.

$$h_1(T) \;=\; a**ba*b*a, \quad h_1(P) \;=\; ab*a, \quad h_2(T) \;=\; *bbb*bbba, \quad h_2(P) \;=\; abb*$$

$$\mathcal{M}_\delta(P,\ T) = \mathcal{D}(ab*a,\ a**ba*b*a) \;\cap\; \mathcal{D}(abb*,\ *bbb*bbba) = \{2,3,5,6\} \cap \{1,5\} = \{5\}\ .$$

Let $\mathcal{H} \;=\; \{h_1, h_2, \ldots, h_k\}$ be a family of morphisms. Define $S_p(\mathcal{H}) \;=\; \{i \;:\; h_i(p) = *\}$ for each $1 \le p \le |\Sigma|$. Next, possible properties P1-P4 of $\mathcal{H}$ are defined.

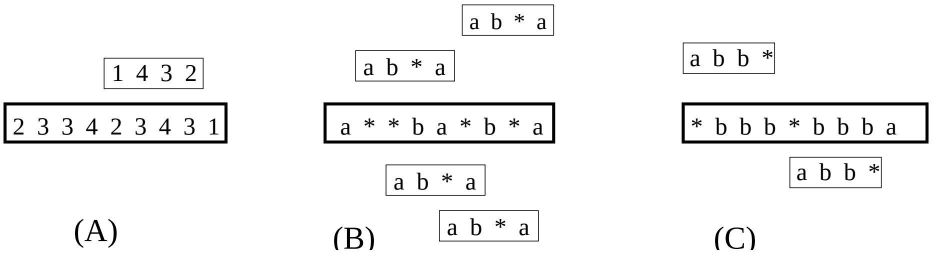

Fig. 1. (A) There is only one 1-match of $P = 1432$ in $T = 233423431$: we have $\mathcal{M}_\delta(P,\ T) \ = \ \{5\}$, (B) $\mathcal{D}(ab*a,\ a**ba*b*a) \ = \ \{2,3,5,6\}$, (C) $\mathcal{D}(abb*,\ *bbb*bbba) \ = \ \{1,5\}$.

P1: (intersecting property) $(\forall\ 1 \le p,q \le |\Sigma|)\ S_p(\mathcal{H}) \cap S_q(\mathcal{H}) \ne \emptyset$;

P2: Each segment of *'s in a fixed row has length δ, or at most δ in case when this is a prefix or suffix segment;

P3: if $h_i(p) \ne *$ and $h_i(q) \ne *$ and there is $*$-position between p and q in the morphism h_i then $h_i(p) \ne h_i(q)$, for each $h_i \in \mathcal{H}$;

P4: Assume that $h_i(p) \ne *$ and $h_i(q) \ne *$, and $p - q > \delta$. then there is a $*$-position between p and q.

Lemma 2.
Assume the family $\mathcal{H}$ satisfies properties P1-P4, then $\mathcal{H}$ is a δ-distinguishing family.

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

*	*	*	1	1	1	1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	4	4
1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	*	*	*	5	5	5	5	5
1	1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	*	*	*	5	5	5	5
1	1	1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	*	*	*	5	5	5
1	1	1	1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	*	*	*	5	5
*	1	1	1	1	*	*	*	2	2	2	2	*	*	*	3	3	3	3	*	*	*	4	4	4	4	*	*	*	5
*	*	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 together a 3-*distinguishing* family.

Theorem 3. $\alpha(\delta) \le 2 \cdot \delta + 1$

Proof. The structure of the morphisms, for $\delta = 3$, is illustrated in Figure 2. 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. It is easy to see that this family satisfies properties P1-P4. By Lemma 2 the constructed family is δ-distinguishing.

Example 2.
For $\delta = 3,\; s = 30$ 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 morphisms are presented by linear arrays, the i-th symbol is the morphic value of the input symbol i.

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

Lemma 3. $\alpha(3) \leq 6$.

Proof. The construction of 6 morphisms for $\delta = 3$ is illustrated in Figure 3. The sets of *stars* in each column have been selected from the family:

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

$$\{1,3,5\},\;\{2,4,5\},\;\{2,3,6\},\;\{1,4,6\},\;\{2,3,4\}\;]$$

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.

The morphism can be constructed in such a way that if there is a star between two positions in the same row then the symbols on these positions are different. The family obeys the properties P1-P3, but not P4. However 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 an arbitrarily large alphabet Σ.

Lemma 4. $\alpha(r \cdot \delta) \leq r \cdot \alpha(\delta)$.

Proof. (Sketch)
We use the linear representation of morphisms. Assume $\{h_1, h_2, \ldots, h_k\}$ is a δ-distinguishing

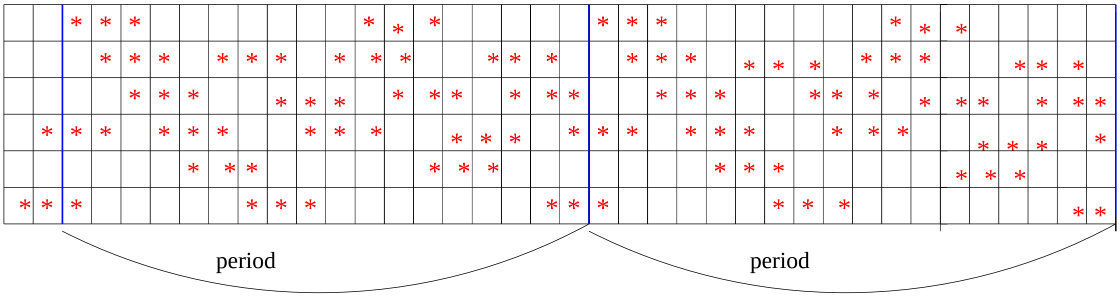

Fig. 3. The periodic structure of 6 morphisms (rows) representing together the 3-distinguishing family of morphisms. Only don't care symbols are shown. This family does not obey the property $P3$ but properties P1 and P2 are satisfied.

family, where each h_i is identified with $h_i(123\ldots|\Sigma|)$. We construct an $r\delta$-distinguishing family by first replacing each symbol x in each h_i by r copies of x. Then each of the resulting linear representations of morphisms is cyclically shifted by j, $1 \leq j < r$. In this way we obtain $k \cdot r$ 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 of length exactly $|\Sigma|$. We omit the correctness proof in this version.

The last two lemmas together imply directly the following fact.

Theorem 4. *If k is divisible by 3, then $\alpha(\delta) \leq 2 \cdot \delta$.*

Theorem 5. [Lower bounds]
(a) $\alpha(\delta) \geq \delta + 2$, *for* $\delta \in \{1, 2\}$ *we have* $\alpha(\delta) = 2 \cdot \delta + 1$.
(b) $\alpha(\delta) = \Theta(\delta)$.

3 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 said to be *binary*.

Theorem 6. *For each $\delta \geq 1$ there are two families $\mathcal{H}1$ and $\mathcal{H}2$ of morphisms 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. $|\mathcal{H}1| = 3$; $|\mathcal{H}2| = 2\delta + 1$.

Proof. We write morphism in a linear form: $h = h(123 \ldots |\Sigma|)$. Take $\mathcal{H}1 = \{h_1,\ h_2,\ h_3\}$, where

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

Additionally we take $2\delta + 1$ morphisms whose structure is the same as the one used in the proof of Theorem 3, see Figure 2. The only difference is that we replace groups of non-star entries alternately by a and b. In this way we get a family of binary morphism, which for positions at distance at most 2δ "behaves" in the same way as the family from Theorem 3. We omit, in this extended abstract version, technical details related to correctness.

As a corollary of Theorem 6 we have the following fact (this asymptotic bound was shown previously in [4]):

Theorem 7. *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 $\delta = 3,\ s = 30$ we take the family $\mathcal{H}1$ given by:

$$\begin{aligned} h_1 &= 111111 * * * 222222 * * * 333333 * * * 444 \\ h_2 &= * * *111111 * * * 222222 * * * 333333 * ** \\ h_3 &= 111 * * * 222222 * * * 333333 * * * 444444 \end{aligned}$$

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

$$\begin{aligned} h_1 &= * * *1111 * * * 2222 * * * 1111 * * * 2222 * * \\ h_2 &= 1 * * * 2222 * * * 1111 * * * 2222 * * * 1111* \\ h_3 &= 11 * * * 2222 * * * 1111 * * * 2222 * * * 1111 \\ h_4 &= 111 * * * 2222 * * * 1111 * * * 2222 * * * 111 \\ h_5 &= 1111 * * * 2222 * * * 1111 * * * 2222 * * * 11 \\ h_6 &= *1111 * * * 2222 * * * 1111 * * * 2222 * * * 1 \\ h_7 &= * * 1111 * * * 2222 * * * 1111 * * * 2222 * ** \end{aligned}$$

4 Final Remarks

The δ-matching problem is a very special instance of approximate string-matching, and its algorithmic complexity is still not well understood. In this paper 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 are 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 $\alpha(r \cdot 3) \leq 6r$. 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, editor, *Proceedings of the Artificial Intelligence and Simulation of Behaviour Symposium*, The Society for the Study of Artificial Intelligence and Simulation of Behaviour, Edinburgh, UK, pp 42-47, 1999.
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, editors, *Proceedings of the 10th Australasian Workshop On Combinatorial Algorithms*, Perth, WA, Australia, pp 129–144, 1999.
3. M. Crochemore, W.Rytter, *Text algorithms*, Oxford Press 1994.
4. R. Cole, R. Hariharan, P. Indyk, Tree pattern-matching and subset matching in $O(n \log^3 n)$ time, in *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 245-254.
5. T. Crawford, C.S. Iliopoulos, R. Raman, String matching techniques for musical similarity and melodic recognition, *Computing in Musicology*, Vol. 11 (1998) pp. 73-100.
6. M.J. Fischer, M.S. Paterson, String-matching and other products, in R. Karp editor, *Complexity of Computation: Proceedings of a Symposium in Applied Mathematics of the AMS and the SIAM*, (1974) pp.113-125