Survey on Methods for Mathematical Expression Analysis in Arabic Handwriting

Elena Smirnova **Stephen M. Watt**
Ontario Research Centre for Computer Algebra,
The University of Western Ontario,
London, ON, N6A5B7, Canada
{elena, watt}orcca.on.ca

Abstract

We address the question of recognizing handwritten mathematics in Arabic and related languages. After presenting an overview of the major styles used to express mathematics in these settings we outline potential problems specific to the representations. Finally we discuss strategies for on-line analysis of handwritten mathematical content in this context.

1 Introduction

On-line analysis of handwritten mathematics in western European notations [1][2][3] and recognition of natural language text written in Arabic scripts [4][5] present two considerably complex problems, both of which have been studied over the past many years. We explore the question of what difficulties arise in handwriting analysis and recognition when mathematics is used in Arabic and similar languages. It is important to understand these issues in order to treat technical, scientific and engineering materials as well as to support pen-based devices in instructional settings. We explore the extent to which this treatment is feasible, what approaches we can adapt from existing techniques in both areas, and what areas need to be re-addressed *ab initio* to deal with the overall problem.

1.1 Problem Overview

To begin with, we identify areas of difficulty that Arabic notations introduce to the analysis of handwritten mathematics:

The first challenging issue is the problem of *stroke segmentation*. While this problem exists already in natural language text, it becomes more complex when dealing with mathematics.

The second major problem comes from *direction inconsistency* in handwriting flow. In most of the cases, mathematics is written in the direction opposite to the surrounding text. One may ask why direction in mathematical content should be such an important issue? Does the meaning of the expressions "$A < B$", "$i \in \mathbf{N}$", or "$z = x + y$" change when we read them from right to left or from left to right? To answer this question, let us consider the following simple formula fragment, $p \mid q$. If this were written from left to right, the mathematical meaning would be p divides q, but if entered from right to left, the same expression would mean q divides p. Obviously, this example as well the formula of Figure 1, demonstrate that interpretation depends on the direction in which the mathematics is written.

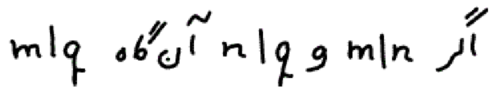


Figure 1. An example of mathematical content sensitive to writing direction

Beyond these two major problems, Arabic notations raise additional issues. These include dealing with a wide collection of new glyphs, including basic and extended Arabic alphabets, considering both dotted or dotless forms of letters and two additional sets of numerals for Arabic-Indic and Eastern Arabic-Indic (see Table 1). In addition, we must recognize a whole range of supplementary symbols and ligatures for certain functions and operators.

Table 1. Various set of numerals used in different Arabic notations

Western Arabic (Europe)	0	1	2	3	4	5	6	7	8	9
Arabic - Indic	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
Eastern Arabic-Indic(Iran, Urdu)	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹

Another pitfall is less obvious and is not taken into consideration in off-line recognition, but does arise in on-line analysis. This is the order and direction of strokes used to enter Latin script based glyphs. For example, a native Arabic writer will most likely draw a fraction bar starting from right to left, symbol of integral and summation from bottom to top and so on. A large number of on-line character recognizers, however, are sensitive to stoke order and direction. For example, recognizers based on elastic matching or hidden Markov Model algorithms will fail to unify any of two digital ink models shown on Figure 2. Besides causing problem with single character recognition, stroke order may also affect structure recognition in terms of building local contexts, later used for candidate prediction (as will be discussed in Section 0).

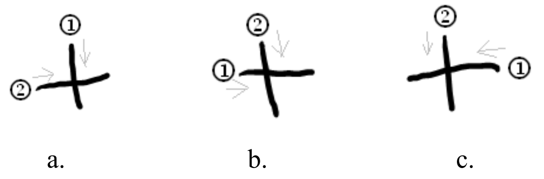


Figure 2. Three different ways of entering character

1.2 State of the Art and Objectives

We have been studying mathematical handwriting analysis as a part of a project on pen-based interfaces for mathematics. So far we have created a framework for on-line digital ink analysis and recognition of handwritten mathematical expressions [3]. This framework architecture provides connections to mathematical software packages such as computer algebra systems, such as Maple, and document processors, such Microsoft Word.

To conduct experiments we have implemented a number of techniques for on-line recognition of handwritten mathematics within European-style notations. The system currently handles more than 240 mathematical characters and glyphs. It also allows uploading different model datasets, including custom-defined glyphs. A large Mathematical Context database has been created to assist the character and structure recognizers of our system [10].

We are presently exploring the possibility of adapting our existing framework and theoretical approaches to mathematical expression analysis in Arabic notations.

2 Variety of Arabic Notations for Mathematics

The authors of [6] present a classification of Arabic mathematical notations used in practice. According to this classification, there are currently four major categories of mathematical Arabic and related notations. We can distinguish them by (1) direction in which mathematics is written and (2) by usage of local alphabets to denote numbers, variables and operators.

2.1 Mathematical Directionality

There are two classes in which mathematics is written from left to right. These are the *Moroccan* and *Persian* styles. In two other classes mathematics flows from right to left. These are called *Maghreb* (meaning "West") and *Machrek* ("East") styles. We will discuss each of these four cases in more detail.

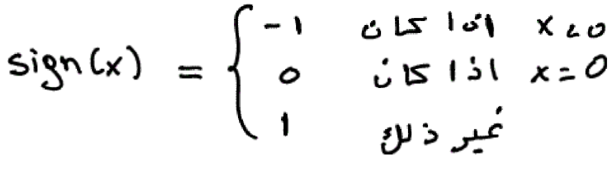


Figure 3. Moroccan style: Arabic text inside a formula

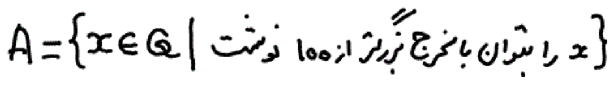


Figure 4. Persian style: Farsi text inside formula

2.2 Origin of Alphanumerics

Each of the four notational styles has a different way of using Arabic and/or Western scripts. In the Moroccan style (Figure 3), mathematical content is entirely written without using Arabic characters, *i.e.* all variables and numerals use European-style notations. Persian style (Figure 4) is similar to Moroccan, except it uses only Eastern Arabic-Indian numerals for numbers. The third case is presented by Maghreb notation, where the variables are written using local alphabets, but numbers are given in Western-Arabic format (Figure 7.a and Figure 8.a). The fourth case is the most interesting: Machrek notations use local alphanumerics not only for numbers and variables, but also for the names of functions and mathematical operators (see Figure 7.b, Figure 8.b and Figure 9.b)

To enable character recognition for all of the above cases, we need to update the model dataset to include all new symbols and then train the recognizer system on the new set. It is also important to present an option for profile selection, where each profile corresponds to one of the notational styles. This will save the recognizer from loading a wide range of glyph models not used in the current notation (and consequently affecting recognition rates).

The following sections discuss the question of how these four Arabic notations affect expression analysis.

2.3 Combination of Text and Mathematics

In both Moroccan and Persian style, plain text is naturally written from right to left, while mathematical context flows in the opposite direction (Figure 3 and Figure 4). This phenomenon is known as *bidirectionality* and also occurs in multi-lingual texts, where, for example, French and Arabic are mixed.

In general, bidirectionality introduces a set of additional difficulties to the document analysis. In particular off-line recognition becomes more difficult. Fortunately, in case of on-line digital ink processing, we are able to detect writing direction through timestamps on strokes order and the relative positioning of glyphs entered. This information can then be used to assist the structure analyzer in distinguishing mathematical content from runs of plain text.

In the case of Maghreb and Machrek notations, mathematical expressions will be written in the same direction as the surrounding text (right to left), except for the case of numbers[1]. Even looking somehow less recognizer-friendly, these notational styles should be easier to process, since they will be a "mirrored" case of already studied mono-directional Latin-based languages [1] [9].

2.4 Mirrored Glyphs

In the cases where mathematics is written from right to left, certain Latin-based characters should be mirrored when they appear inside of Arabic notations (see Figure 5). The most common cases of symbols requiring mirroring are opening and closing parentheses, symbols of asymmetric relations and operations, such as $<$, $\leq$, $\Rightarrow$, $\in$, $\subset$, $\subseteq$, $\rightarrow$, $\backslash$. This does not introduces much of a problem to recognition of the characters themselves, as long as their mirrored images are present in a model database and are supported by rendering tools (such as the Unicode standard). However mirrored glyphs may cause errors on the stage of expression recognition, as will be discussed in 3.2.

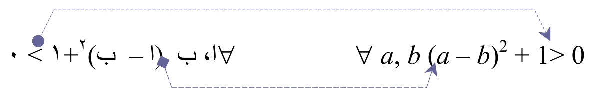


Figure 5. Example with mirrored glyphs

Aside from the previous case, a number of symbols used in Maghreb and Machrek notations do not have native mirrored images in the set of European-style mathematical characters. The formula in Figure 6 presents such a case. These special symbols will then require adding proper recognition models through additional glyphs. The authors of [7] raised the question of including reversed characters for integrals, radicals, summation etc. in the Unicode standard dataset. Until then, rendering of reversed symbols and ligatures as preset glyph images can be a possible solution, similar to one we had to develop in our system for rendering of the OpenFace alphabets [8]. This is not suitable, however, for high-quality typesetting.

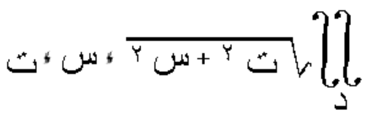


Figure 6. Special cases of mirrored glyphs

2.5 Delimiters and Special Operators

Large operators used for subexpression grouping have always been a special issue in rendering and writing mathematical content, even for most common European notations. The Arabic and Persian styles introduce six additional glyphs used for the operators of summation (Figure 7), product (Figure 8) and limit (Figure 9).

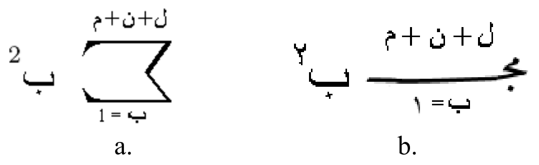


Figure 7. Notations for the sum operator in Maghreb (left) and Machrek (right) styles

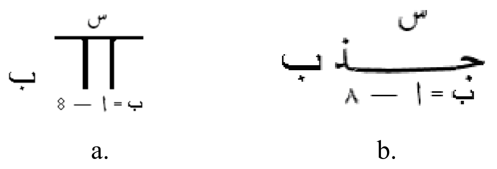


Figure 8. Notations for the product operator in Maghreb (left) and Machrek (right) styles

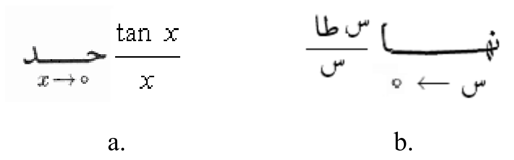


Figure 9. Notations for the limit operator in Persian(left) and Machrek (right) styles

As shown in the above formulas, large operators in Arabic notation are usually stretched to the same width as their lower and upper limits. Since the ligatures are stretched in one direction only, without preserving the ratio, this introduces a difficulty for character recognizers to match stretched forms with fixed-width models. This problem, however, fits the already studied case for matching symbols of long radicals to their normalized models [8].

In addition to new notations for large operators, factorial and binomial coefficients may appear in their own special form, using the symbol ل (LAM), see Figure 10.

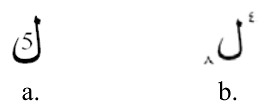


Figure 10. Arabic notation for the factorial(left) and binomial coefficient (right)

[1] Numbers are always written from left to right for all Arabic notations, independently of which system of numerals is used.

3 Influence of Notations on Expression Analysis

In this section we discuss how an overall expression analysis will be affected by dealing with various Arabic mathematical notations. This not only addresses the question of structure recognition, but also the question of *how to interpret* the recognized mathematical context.

3.1 Implicit directionality

As we saw in sections 1.1 and in 2.4, it is important to determine the direction in which mathematical context was entered before proceeding with expression analysis. Furthermore, we must be aware of the situation where mathematical directionality is changed implicitly.

Consider a statement in English "A^2>0 **if** A>0". If we ask a native Persian speaker to write this down in Farsi, we will get something looking like the expression in Figure 11. Suppose the recognizer determined all the glyphs correctly {A,>, ٠, اگر, A,٢,>,٠}. Given the information that the notation used is Persian and mathematical context there flows from left to right, the structure analyzer may directly translate this to "A>0 **if** A^2>0" (if ↔اگر), which will be a surprisingly wrong interpretation of the original mathematical content.

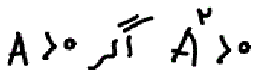
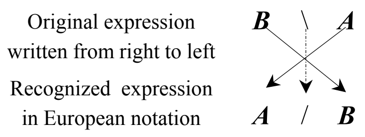


Figure 11. Statement "A^2>0 **if** A>0" written in Farsi

3.2 Careful Mirroring

Every asymmetric symbol of mathematical relation or operation used in right to left notation has to be carefully assigned to its mirrored couple. And the writer must be aware that only mirrored symbols will be expected in the context. Failing to do so may not only end up in having an inversion of the original content, as we saw in 3.1, but also can lead to misinterpretation of the mathematical operators.

Suppose user writes from right to left "$B \setminus A$". At the stage of expression analysis, the order of the characters is reversed and the symbol "\" must be substituted by its mirrored couple "/":


Original expression written from right to left
B \ A
Recognized expression in European notation
A / B

This gives an expression which carries the same meaning as the European style "A/B", that is "A divided by B". If the original character "/" is not properly replaced by "/", the recognizer will return an expression "A\B", meaning "set A minus set B".

3.3 Additional Container Symbols

The notation for factorial introduces one more case of a *container symbol*, in addition to the symbols for radical and long division. This requires the addition of a new set of rules to the structural analyzer so, for example, the layout of the expression in Figure 10.a will be detected as nested (Figure 12.a) rather than linear (Figure 12.b).

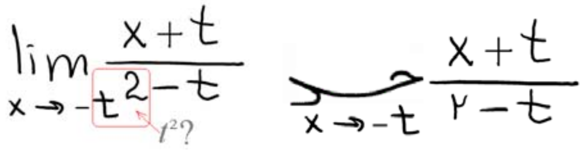
```
[Row] — ل
       \ [Contains]          [Row] — ل — 5
        5
```

a. (right) b. (wrong)

Figure 12. Possible layout tree for Figure 10.a.

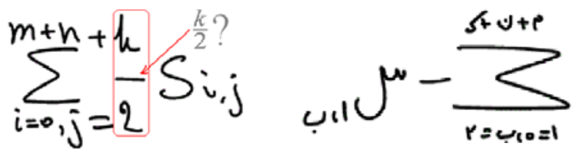
3.4 Stretched Large Operators

The stretched delimiter operators, presented in Section 2.5, are written long enough to allow lower and upper limits to fit directly above or beneath the bounding box of the operator glyph. This will actually assist the structural analysis of handwritten formulae to detect characters that belong to under- and superscript areas of large operators more accurately than in the non-Arabic case. To see this, compare the pairs of expressions in Figure 13, Figure 14 and Figure 15.



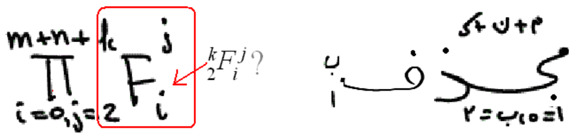
a. (ambiguous) b. (clear)

Figure 13. Ordinary and stretched notations for the limit operator



a. (ambiguous) b. (clear)

Figure 14. Ordinary and stretched notations for the N-ary summation operator



a. (ambiguous) b. (clear)

Figure 15. Ordinary and stretched notations for the N-ary product operator

3.5 Use of Mathematical Context

In our framework for pen-based computing, a mathematical context database is used to assist an isolated character recognizer to disambiguate between similar candidates [10]. For example, this allows the lower and upper limits of integration in the formula of Figure 16.a to be correctly recognized as "0" (number) and "∞"(infinity) correspondingly, instead of their possible interpretations as "o" (letter) and "∝" (proportional) or "○" (circle) and "α" (alpha).

A similar approach could be applied to recognizing handwritten mathematics in Arabic settings. For example, ا ("ALEF") can easily be confused with Eastern-Arabic numeral ١ ("1"); ٠("0") can be taken for a dot; numeral ٥("5") can be interpreted as ه("HEH") or the symbol for degree "°". Having a mathematical context database for Arabic notations would help to disambiguate the case shown in Figure 16.b, where the lower limit should be detected as ٠("0") and the upper limit as ١ ("1").

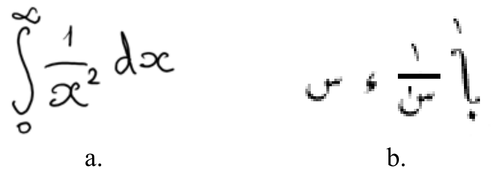


Figure 16. Context used for character disambiguation in European (left) and Arabic(right) notations

In practical terms, developing such a database may be difficult to achieve: Our current database for European expression frequencies has been derived from the analysis of tens of thousands of mathematical documents in TeX form, with millions of mathematical character sequences. We have not identified a suitable source of Arabic training material.

4 Conclusions and Future Work

We have reviewed various notational styles for mathematics used in Arabic script based languages, and we have identified a set of problems these notations introduce to recognition of handwritten mathematics.

As an outcome of this review, we anticipate that a wide range of methods already developed for mathematical handwriting analysis for European languages to be applicable in the case of Arabic notations. Beyond this, we realize that Arabic customs for handwritten mathematics introduce new classes of problems, mainly dealing with stroke segmentation and structure analysis in bidirectional notations. These are intensified by different ways and orders for entering both individual mathematical symbols and whole expressions.

Aside from the new challenges, we have discovered certain things that are easier with Arabic notations. Among them are clearer structure organization for expressions with large delimiter operators and more explicit distinction between mathematical and text fragments (in bidirectional notations).

Future work on this subject includes training our recognition framework [3] for Arabic scripts, and developing tools for automated notational profile detection [11]. In the long term, we hope to bring as much as possible of the already understood approaches for mathematical handwriting analysis to Arabic notation, including "mirroring" of existing methods for structure recognition.

Acknowledgements

We would like to thank Hanan Lutfiyya and Azar Shakoori for their patient explanations and high-quality examples of handwritten mathematical expressions they provided for this paper.

References

[1] D. Blostein and A. Grbavec. Recognition of Mathematical Notation. in *H. Bunke and P.S.P. Wang (editors) Handbook of Character Recognition and Document Image Analysis*, pp. 557-582, World Scientific, Singapore, 1996.

[2] Bo Wan and Stephen Watt, An Interactive Mathematical Handwriting Recognizer for the Pocket PC", In Proc. *International Conference on MathML and Technologies for Math on the Web*, 2002, http://www.mathmlconference.org/2002/ /presentations/wan/

[3] Elena Smirnova and Stephen M. Watt, A Context for Pen-Based Computing, pp. 409-422*, Proc. Maple Conference 2005*, July 2005, Waterloo Canada, Maplesoft.

[4] T. Sari and M. Sellami, Cursive Arabic Script Segmentation and Recognition System. *International Journal of Computers and Applications*, Vol. 27, 2005.

[5] Al-Emami, S., and Usher, M., On-Line Recognition of Handwritten Arabic Characters. Pattern Analysis and Machine Intelligence, IEEE Transactions (PAMI) Vol. 12, No. 7, 1990, pp. 704-710.

[6] Azzeddine Lazrek, Mustapha Eddahibi, Khalid Sami, Cadi Ayyad, Bruce R. Miller. Arabic mathematical notation. *W3C Interest Group Note*, January 2006. http://www.w3.org/TR/arabic-math/

[7] Mohamed Jamal Eddine Benatia, Azzeddine Lazrek, Khalid Sami, Arabic mathematical symbols in Unicode, *IUC 27*, Berlin, Germany, April 2005. http://www.ucam.ac.ma/ fssm/ rydarab/doc/communic/unicodem.pdf.

[8] Stephen M. Watt and Xiaofang Xie, Recognition for Large Sets of Handwritten Mathematical Symbols, *Proc. IEEE International Conference on Document Analysis and Recognition*, 2005, Seoul Korea.

[9] Ian Ruterford, Structural analysis for pen-based math input, *MMath Thesis*, University of Waterloo, 2005.

[10] Elena Smirnova and Stephen M. Watt, Combining Prediction and Recognition in Mathematical Handwriting Analysis, *Ontario Research Centre for Computer Algebra, , Research Report TR-06-03*, June 2005, London, Canada. http://www.orcca.on.ca/ /TechReports/2006/TR-06-03.html

[11] Elena Smirnova and Stephen M. Watt, Notation Selection in Mathematical Computing Environments, pp. 339-355, *Proc. Transgressive Computing 2006* (TC 2006), April 2006, Granada Spain.