УДК 519.17, 510.5

© *А. Ю. Сапаров, И. Ю. Широбокова*

РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЛЯ УПРАВЛЕНИЯ ПРОЦЕССОМ РАСПОЗНАВАНИЯ РУКОПИСНЫХ МАТЕМАТИЧЕСКИХ ФОРМУЛ [1]

В работе рассматривается задача распознавания рукописных математических формул. Описываются основные проблемы, возникающие при решении данной задачи. Описывается метод предупреждения и исправления ошибок распознавания, основанный на ручном управлении процессом распознавания. Приводятся математические модели предложенного метода, основанные на использовании элементов теории графов. Для этого вводится понятие регулярного дерева изображения формулы, которое позволяет хранить все варианты распознавания исходного изображения формулы в наиболее компактном виде и упрощать процесс группового редактирования множества вариантов распознавания, связанный с изменением характера связи между вершинами графа. Приводится пример удобного интерфейса программы для управления процессом распознавания и исправления ошибок, который не требует от пользователя знания формата представления математических формул.

Ключевые слова: распознавание математических формул, промежуточный результат распознавания, правильный итоговый результат распознавания, регулярное дерево изображения формулы.

DOI: 10.20537/vm160111

Введение

При разработке системы автоматического распознавания текстов следует обращать внимание не только на повышение качества распознавания, но и на удобство использования этой системы. Зачастую разработчики не уделяют должного внимания такому важному пункту, как исправление ошибок, в результате чего этот процесс становится довольно утомительным занятием. При наличии большого числа однотипных ошибок становится неразумным ручное исправление всех ошибок по отдельности, и такое занятие по-разному действует на психику людей. В таком случае проблема решается при помощи некоторой классификации ошибок с целью добавления новых шаблонов распознавания и дальнейшей повторной обработки входных данных. Такой подход имеет сразу несколько недостатков. Во-первых, для добавления шаблона может потребоваться высококвалифицированный специалист. Во-вторых, выбор характерных признаков для классификации может оказаться достаточно трудной задачей, так как необходимо учитывать машинное представление исходных объектов. В-третьих, это может потребовать неоднократную повторную обработку всех входных данных, и, в зависимости от их объема и сложности, процесс может занять длительное время. Кроме того, большинство систем распознавания текстов не предусматривают возможность добавления шаблонов распознавания, поэтому единственным решением является ручное исправление ошибок.

Основная сложность при распознавании рукописной математической формулы, по сравнению с обычными текстами, заключается в сложности ее структуры и большом разнообразии вариантов записи отдельных символов. Из этого следует, что шаблоны распознавания в этом случае представляют собой более сложные объекты. Следовательно, и проблемы при исправлении ошибок распознавания более значительны. Кроме того, в данном случае пользователь должен обладать некоторыми профессиональными навыками, а именно: должен быть знаком

[1]Работа выполнена при финансовой поддержке РФФИ (грант 14–07–00934).

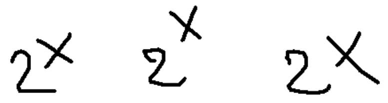

Рис. 1. Примеры изображений степенной формулы

с самим форматом представления математических формул на достаточном уровне. Для избавления от перечисленных проблем предлагается рассмотреть механизм динамического управления процессом распознавания рукописных математических формул. Идея заключается в представлении процесса распознавания в виде некоторой последовательности автоматизированных действий. Механизм подразумевает способность возврата к произвольному элементу из этой последовательности в зависимости от полученного результата. Таким образом, полный запуск всего процесса распознавания не требуется, но имеется возможность запускать только необходимые пункты. Реализация данного механизма заключается в создании системы автоматического выделения возможных ошибок на каждом этапе вычислений и представлении полученной информации в удобном виде. Особое внимание уделяется созданию интерфейса, который предоставляет возможность пользователю управлять всем процессом распознавания, а именно: просмотром промежуточных результатов, запуском процесса распознавания с произвольного этапа, добавлением шаблонов распознавания, ручным исправлением отдельных ошибок и групповым исправлением ошибок.

§ 1. Основные проблемы при распознавании рукописных математических формул

К основным проблемам при распознавании рукописных математических текстов можно отнести:

1. Невозможность определения точного взаимного расположения символов.
2. Неточность при написании отдельного символа.
3. Плохое качество исходного изображения.

1) В зависимости от почерка человека можно наблюдать различное взаимное расположение отдельных символов. Рассмотрим небольшой пример. Пусть простейшая степенная формула 2^x написана разными людьми (рис. 1), тогда при автоматическом анализе характера взаимного расположения двух символов может быть получено два различных результата. В итоге результатом распознавания может быть либо $2x$, либо 2^x. Очевидно, что при увеличении сложности формулы увеличивается число вариантов взаимного расположения отдельных символов формулы. В зависимости от почерка автоматический анализ может давать результаты с некоторыми ошибками, либо *неопределенностями.* Под неопределенностью понимается невозможность точного определения взаимного расположения. Например, с вероятностью 60% символ x находится выше и правее, чем символ 2, и с вероятностью 40% символ x находится строго справа от символа 2. В таких случаях невозможно установить правильность того или иного варианта распознавания без понимания контекста всего математического текста.

2) Неточность при написании отдельного символа также связана с использованием различных почерков. Кроме того, важную роль играет качество записывающего устройства (если запись не выполняется на экране планшета) и качество сканирования документа. В зависимости от расхождения с некоторым эталонным способом записи отдельного символа (наклонов, поворотов, сдвигов, разрывов) при анализе отдельного символа могут быть получены различные варианты распознавания. Рис. 2 показывает, как в некоторых случаях могут быть получены неопределенности при распознавании символа c и цифры 0. Аналогично с первой проблемой, в данном случае невозможно со 100% уверенностью сказать, какой вариант распознавания правильный.

3) Данная проблема является более общим случаем по сравнению с предыдущими двумя: она включает в себя главным образом наличие в исходном изображении каких-то посторонних

Рис. 2. Примеры изображений отдельных символов

шумов, которые при распознавании могут быть восприняты как объекты математической формулы, что в результате приводит к получению неправильного результата. Кроме того, в исходном изображении может быть потеряна значимая часть из-за сканирования низкого качества. Существуют некоторые алгоритмы предварительной обработки изображения для подготовки к дальнейшему анализу, поэтому будем считать, что изображение не содержит шумов и отсканировано с высокой точностью. Далее описываются подходы, позволяющие решать только первые две проблемы из рассмотренных.

§ 2. Общее представление результата распознавания математической формулы

Промежуточный результат распознавания представляется в виде множества графов изображений формулы. Каждый элемент множества — это возможный вариант взаимного расположения всех компонент формулы. Например, для изображения формулы из рис. 1 результат может быть представлен в виде множества из двух графов: $\{F_1, F_2\}$, где $F_1 = (\{2, x\}, 2x(1, 0))$, $F_1 = (\{2, x\}, 2x(1, 1))$. Вершинам графа соответствуют отдельные символы и математические знаки, а ребра описывают их взаимное расположение. Ребро $2x(1, 1)$ содержит информацию о том, что символ x находится правее и выше, чем символ 2. Ребро $2x(1, 0)$ содержит информацию о том, что символ x находится строго справа от символа 2. В свою очередь, вершины 2 и x представляются графами символов. Граф символа 2 содержит 2 вершины и 1 сегментированное ребро, соединяющее эти вершины; граф символа x содержит 5 вершин и 4 ребра, соединяющие центральную вершину с остальными. Ребра графа символа состоят из 8 сегментов и дают приблизительное описание траектории движения записывающего устройства. Вершины характеризуют точки, в которых запись либо начинается (крайняя точка), либо заканчивается (крайняя точка), либо выполняется более одного раза (точка пересечения).

Каждый символ может иметь несколько вариантов распознавания, поэтому промежуточный результат при распознавании отдельного символа представляется целым множеством шаблонных символов и указанием некоторого числового значения для каждого элемента, которое называется весом. Это число характеризует, на сколько тот или иной символ похож на шаблонный. Для промежуточного представления целой строки используется сразу несколько множеств. Если строка содержит 3 элемента, то результат представляется в виде 3 множеств, каждое из которых содержит возможные варианты распознавания соответствующего символа. Таким образом, общее число вариантов распознавания всей строки равно произведению числа элементов во всех множествах. Для длинных строк это число может быть достаточно большим даже для компьютера, поэтому для удобства результат представляется в виде взвешенного регулярного выражения:

$$S = (p_1 \cdot a_1 | \ldots | p_n \cdot a_n)(q_1 \cdot b_1 | \ldots | q_m \cdot b_m) \ldots (r_1 \cdot d_1 | \ldots | r_k \cdot d_k),$$

где p_i, q_i, r_i — веса, $a_{1,\ldots,n}$, $b_{1,\ldots,m}$, $d_{1,\ldots,k}$ — варианты распознавания отдельных символов.

В итоге общий вид промежуточного результата распознавания с t вариантами взаимного расположения символов имеет следующую структуру:

$$F = \{(S, U_1), (S, U_2), \ldots, (S, U_t)\},$$
$$S = \{((p_{1,1} \cdot a_{1,1} | \ldots | p_{1,n_1} \cdot a_{1,n_1}), (s_1, v_1)), \ldots, ((p_{l,1} \cdot a_{l,1} | \ldots | p_{l,n_l} \cdot a_{l,n_l}), (s_l, v_l))\},$$
$$U_i = \{(s_{b_j}, s_{e_j}, v_{i,j})\},$$

где l — количество символов в формуле, n_j — количество вариантов распознавания j-го символа, (s_j, v_j) — граф j-го символа, $a_{i,j}$ — j-й вариант распознавания i-го символа формулы, $p_{i,j}$ — вес j-го варианта распознавания i-го символа формулы, $s_{b_j} \in S$ — начало j-го ребра, $s_{e_j} \in S$ — конец j-го ребра, $v_{i,j} \in \{\{-1,0,1\} \times \{-1,0,1\} \backslash \{(0,0)\}\}$ — направление j-го ребра.

Итак, в общем виде результат распознавания — это множество графов изображений формулы, в которых вершины, помимо информации о скелете отдельных символов, содержат список вариантов распознавания этих символов в виде взвешенного регулярного выражения.

Замечание 1. В общем виде результата распознавания у всех элементов множества графов список вершин один и тот же, т. е. они отличаются только списком ребер.

§ 3. Пути решения выявленных проблем

Для решения выявленных проблем предлагается рассмотреть следующие подходы:

1. Сопоставление с шаблонным графом формулы.
2. Поиск по регулярному выражению.
3. «Регулярное дерево» изображения формулы.
4. Использование истории распознавания.

1) Данный подход был подробно рассмотрен в работе [2], в которой описывается способ моделирования изображения рукописной математической формулы с использованием элементов теории графов [3]. В работе рассматриваются такие понятия, как граф изображения формулы и отображение класса формулы. На основе графа изображения формулы строится модель рукописной математической формулы. Отображение класса формулы строит граф этой формулы. Метод заключается в построении графа изображения формулы и решении задачи нахождении обратного отображения, что позволяет по построенному графу определить, к какому классу относится формула. Таким образом, метод позволяет выбрать тот вариант распознавания, который по структуре подходит под рассматриваемые классы формул (строчная формула, нижний индекс, верхний индекс, дробь, интеграл, извлечение корня и т. д.). Предложенный метод позволяет анализировать формулы с неограниченным уровнем вложенности при построении формул более высоких порядков.

2) Для уточнения результатов распознавания используются регулярные выражения [4]. Метод применения для решения данной задачи подробно описан в работе [1]. Способ заключается в задании множества всевозможных вариантов распознавания в виде взвешенного регулярного выражения, которое отличается от обычного регулярного выражения тем, что каждому элементу порождаемого этим выражением множества ставится в соответствие некоторый числовой вес, отражающий превосходство одних элементов над другими. Веса вычисляются по принципу схожести рассматриваемого символа изображения с шаблонным символом. Подход позволяет задавать шаблоны слов в виде регулярных выражений. Каждый шаблон может содержать бесконечное число элементов, что важно при работе с такими объектами, как математические формулы, так как их разнообразие может быть бесконечно большим. Важным достоинством предложенного метода является то, что он позволяет использовать бесконечные множества на практике и избавляет от необходимости перебора всех вариантов распознавания при поиске правильного варианта.

3) В зависимости от качества исходных данных, итоговый результат распознавания может содержать различное число ошибок. При обработке рукописных формул, написанных неразборчивым почерком, получить правильный итоговый результат практически невозможно. В этом случае приходится исправлять ошибки вручную, что может оказаться довольно трудозатратным занятием. Предлагается рассмотреть способ исправления результатов, заключающийся во вмешательстве человека в процесс автоматического распознавания. Таким образом, пользователь сможет на более ранних этапах предупредить возникновение ошибки и тем самым исключить возникновение других ошибок, связанных с данной ошибкой. Для реализации метода предлагается использовать понятие промежуточного результата распознавания. Так как он может состоять из множества вариантов, то требуется его представление в удобном виде с

точки зрения как пользователя, так и программы. После каждого вмешательства пользователя должна выполняться автоматическая обработка результатов с уже уточненными данными, что приведет к получению правильных результатов. С точки зрения пользователя, метод подразумевает наличие удобного интерфейса для редактирования промежуточных результатов распознавания. Более подробно механизм будет описан в следующих параграфах.

4) Под историей распознавания подразумевается список результатов обработки предыдущих входных данных, для которых подтверждена правильность распознавания. При наличии неопределенностей в итоговом результате выполняется поиск соответствия в истории. К основным критериям поиска относятся структура построенного графа изображения формулы и значение отдельных символов. Если найдено полное соответствие, то выбирается этот элемент. Если полного соответствия нет, то выполняется нечеткий поиск, т. е. некоторые вершины построенного графа могут не соответствовать полностью, а являться рекурсивной заменой графом другой формулы (формула следующих уровней вложенности). Такой поиск позволяет выполнить сортировку промежуточного результата распознавания и быстро найти правильный вариант среди предложенных.

§ 4. Операции с графом изображения формулы

Промежуточный результат распознавания математической формулы содержит несколько вариантов построения итогового графа изображения формулы. Дальнейшая задача заключается в выборе правильного варианта среди предложенных. Если список вариантов распознавания небольшой (2–5), то целесообразным является просмотр всех вариантов и ручной выбор правильного. При наличии большого числа вариантов простой перебор становится неэффективным в связи с необходимостью просмотра и концентрации внимания на большом списке вариантов. В таком случае оказывается быстрее исправить неверный результат вручную, а не искать среди предложенных. Однако требуется знание формата хранения математической формулы, что может привести к некоторым проблемам у большинства пользователей.

Рассмотрим способ исправления ошибок распознавания, не требующий полного перебора всех возможных вариантов и не требующий знания формата.

Определение 1. $F_t \in F$ будем называть *правильным итоговым результатом распознавания*, если пользователь подтверждает, что структура графа F_t соответствует изображению формулы на входных данных.

Определение 2. Пусть F — промежуточный результат распознавания. Функцию $FC : F \longrightarrow \{0,1\}$ будем называть *функцией проверки итогового результата*, если выполнено условие:

$$FC(F_t) = \begin{cases} 1, & \text{если } F_t \text{ — правильный итоговый результат распознавания,} \\ 0, & \text{в противном случае.} \end{cases}$$

Условие 1. Пусть M — множество подмножеств графов изображений формулы. Будем считать, что $\forall F \in M\ \exists F_t \in F :\ FC(F_t) = 1$. Иными словами, в промежуточном результате всегда существует правильный вариант распознавания.

Условие 2. Пусть F — промежуточный результат распознавания, тогда $\forall F_t, F_l \in F$: $\Big(FC(F_t) = 1 \text{ и } FC(F_l) = 1\Big) \Rightarrow (F_t = F_l)$. Иными словами, в промежуточном результате всегда единственный правильный вариант распознавания.

Определение 3. Пусть M — множество подмножеств графов изображений формулы. Функцию $FR : M \longrightarrow M$ будем называть функцией *уточнения* результата распознавания, если $\forall F \in M :\ FR(F) \subset F$. Иными словами, функция *уточнения* исключает из промежуточного результата распознавания некоторые результаты согласно некоторому правилу.

Определение 4. Пусть $F_t \in F$ — правильный итоговый результат распознавания. Функцию уточнения FR будем называть *позитивной*, если $F_t \in FR(F)$. Функцию уточнения FR будем называть *негативной*, если $F_t \notin FR(F)$. Иными словами, *позитивная* функция уточнения исключает из промежуточного результата распознавания только неверные варианты, а *негативная* — также и верный вариант. Согласно условию 1, результатом выполнения *негативной* функции уже не является промежуточный вариант распознавания, так как в нем не существует правильных вариантов.

Определение 5. Пусть $G = (S, U)$ — граф изображения формулы. Через $(s_1, s_2) \in U$ будем обозначать ребро графа, если не имеет значения, какое направление имеет ребро. Будем считать, что выполнено равенство $(s_1, s_2) = (s_1, s_2, v)$, где v — направление ребра, если $(s_1, s_2, v) \in U$.

Определение 6. Пусть $F_1 = (S, U_1)$ и $F_2 = (S, U_2)$ — графы изображений формулы. Будем говорить, что граф F_1 *соответствует по структуре* графу F_1 ($F_1 \sim F_2$), если выполнено условие:

$$\forall s_1, s_2 \in S \; \forall v \in \Big\{\{-1,0,1\} \times \{-1,0,1\} \backslash \{(0,0)\}\Big\} : \\ \Big((s_1, s_2, v) \in U_1 \Rightarrow \exists w \in \Big\{\{-1,0,1\} \times \{-1,0,1\} \backslash \{(0,0)\}\Big\} \; (s_1, s_2, v) \in U_2\Big).$$

Иными словами, два графа изображения формулы *соответствуют по структуре друг другу*, если они имеют одинаковые вершины, а ребра отличаются только направлениями.

Определение 7. Пусть F — множество графов изображений формул. F будем называть *моноструктурированным*, если $\forall F_1, F_2 \in F : \; (F_1 \sim F_2)$. Иными словами, множество графов изображений формул *моноструктурировано*, если все графы из этого множества соответствуют друг другу.

Определение 8. Пусть F — моноструктурированное множество графов изображений формулы. Запись вида

$$F = \Big(S, \Big\{(s_{b_1}, s_{e_1}, v_{1,1}|v_{1,2}|\ldots|v_{1,m_1}), \ldots, (s_{b_n}, s_{e_n}, v_{n,1}|v_{n,2}|\ldots|v_{n,m_n})\Big\}\Big),$$

где S — множество вершин графов из F, $s_{b_i} \in S$ — начало i-го ребра, $s_{e_i} \in S$ — конец i-го ребра, $v_{i,1}, v_{i,2}, \ldots, v_{i,m_i} \in \Big\{\{-1,0,1\} \times \{-1,0,1\} \backslash \{(0,0)\}\Big\}$ — все варианты направлений соответствующих ребер с началом в s_{b_i} и концом в s_{e_i}, будем называть *регулярным деревом изображения формулы* для множества графов изображений формул F.

Пример 1. Рассмотрим пример. Пусть есть изображение формулы xyz (рис. 3). Так как взаимное расположение символов неоднозначное, то имеется сразу несколько вариантов распознавания, а именно: xyz, x^{yz}, xy^z, x^{y^z}. Очевидно, что при построении вариантов распознавания все графы будут содержать 3 вершины: x, y, z. Множество графов будет моноструктурированным, так как не сложно точно определить порядок следования символов. Соответственно, все графы будут иметь по 2 ребра: (x, y) и (y, z). Графы изображений формулы примут вид:

$$F_1 = (\{x, y, z\}, \{(x, y, (1,0)), (x, y, (1,0))\}),$$
$$F_2 = (\{x, y, z\}, \{(x, y, (1,1)), (x, y, (1,0))\}),$$
$$F_3 = (\{x, y, z\}, \{(x, y, (1,0)), (x, y, (1,1))\}),$$
$$F_4 = (\{x, y, z\}, \{(x, y, (1,1)), (x, y, (1,1))\}).$$

В виде регулярного дерева результат будет выглядеть следующим образом:

$$F = (\{x, y, z\}, \{(x, y, (1,0)|(1,1)), (y, z, (1,0)|(1,1))\}).$$

Такая запись имеет более компактный вид, что важно при наложении ограничений на такие параметры, как объем требуемой памяти и количество операций в единицу времени.

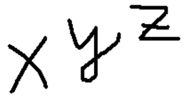

Рис. 3. Пример изображения формулы

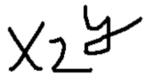

Рис. 4. Пример изображения формулы с неоднозначным взаимным расположением символов

Замечание 2. Заметим, что регулярное дерево изображения формулы может содержать большее количество элементов, чем исходное множество графов изображений формул. В примере 1, если известно, что x^{y^z} не может быть правильным вариантом распознавания, то множество содержит только 3 элемента, тогда как в регулярном дереве сохраняются все варианты направлений, а, значит, и все варианты распознавания. Отметим, что при решении таких задач, как распознавание рукописных математических формул, важно на протяжении всех вычислений хранить все предполагаемые варианты. Данный подход позволяет как минимум уменьшить требуемую память для вычислений и за один проход обрабатывать сразу несколько различных графов изображений формул.

Определение 9. Пусть F — регулярное дерево изображения формулы. Множество графов изображений формулы, порождаемое регулярным деревом F, будем называть *регулярным множеством графа изображения формулы*.

Определение 10. Будем говорить, что промежуточный результат распознавания F имеет *нормальную форму*, если он имеет вид:

$$F = \bigcup_{i=1}^{n} F_i,$$

где каждое множество F_i представлено регулярным деревом. Иными словами, *нормальная форма* — это объединение моноструктурированных подмножеств множества F.

Пример 2. Рассмотрим пример. Для изображения формулы на рис. 4 может быть несколько вариантов распознавания, а именно: $x_2{}^y$, x_2y, x_{2^y}, $x2^y$, $x2y$. Для данного примера связи между символами неоднозначны, т. е. точно не известно, с каким символом должен быть связан символ y. Он относится либо к символу x, либо к цифре 2. В первых двух вариантах распознавания два ребра: $(x, 2)$, (x, y). В остальных случаях ребра соединяют другие вершины: $(x, 2)$, $(2, y)$. Для такого промежуточного результата распознавания может быть построено следующее множество в нормальной форме:

$$F = \Big\{ \Big(\{x, 2, y\}, \{(x, 2, (1, -1)), (x, y, (1, 1)|(1, 0))\} \Big) \Big\} \bigcup \\ \Big\{ \Big(\{x, 2, y\}, \{(x, 2, (1, -1)|(1, 0)), (2, y, (1, 1)|(1, 0))\} \Big) \Big\}.$$

Определение 11. Пусть FR — позитивная функция уточнения для F. FR будем называть *идеальной функцией уточнения*, если выполнено условие: $FR(F) = \{F_t\}$, где F_t — правильный итоговый результат распознавания.

Определение 12. Пусть $G = (S, U)$ — обыкновенный граф, тогда через $S(G)$ будем обозначать множество вершин графа G, а через $U(G)$ — множество его ребер.

Определение 13. Пусть $F = \{(S, U_1), \ldots, (S, U_n)\}$ — промежуточный результат распознавания, $s_1, s_2 \in S$ — произвольные вершины. Определим для промежуточного результата распознавания несколько *операций*:

Добавление ребра:

$$F + (s_1, s_2) = \Bigg\{ \bigg(S, \Big(U_1 \setminus \bigcup_i \{(s_i, s_2)\} \Big) \bigcup \{(s_1, s_2)\} \bigg), \ldots, \bigg(S, \Big(U_n \setminus \bigcup_i \{(s_i, s_2)\} \Big) \bigcup \{(s_1, s_2)\} \bigg) \Bigg\}.$$

В нормальной форме операция имеет вид:

$$\bigcup_{i=1}^{n} F_i + (s_1, s_2) = \bigcup_{i=1}^{n} F_i \setminus \bigcup_{i:(s_1,s_2) \notin U(F_i)} F_i.$$

Добавление направления ребра:

$$F + (s_1, s_2, v) = \Bigg\{ \bigg(S, \Big(U_1 \setminus \bigcup_{i:v_i \neq v} \{(s_1, s_2, v_i)\} \Big) \bigcup \{(s_1, s_2, v)\} \bigg), \ldots, \bigg(S, \Big(U_n \setminus \bigcup_{i:v_i \neq v} \{(s_i, s_2, v_i)\} \Big) \bigcup \{(s_1, s_2, v)\} \bigg) \Bigg\}.$$

В нормальной форме операция имеет вид:

$$\bigcup_{i=1}^{n} F_i + (s_1, s_2, v) = \Bigg(\bigcup_{i:(s_1,s_2) \notin U(F_i)} F_i \Bigg) \bigcup \Bigg(\bigcup_{i:(s_1,s_2) \in U(F_i)} \Big(S, U(F_i) \setminus \{(s_1, s_2)\} \bigcup \{(s_1, s_2, v)\} \Big) \Bigg). \quad (4.1)$$

Отметим, что в (4.1) F_i — это целое множество графов, но запись вида $U(F_i)$ допустима, так как F_i представлено в виде одного регулярного дерева.

Операция *добавление ребра* добавляет во все графы промежуточного результата распознавания указанное ребро, предварительно удалив все ребра, у которых конец совпадает с указанным. Операция *добавление направления ребра* добавляет в указанное ребро во всех графах промежуточного результата распознавания указанное направление, предварительно удалив все остальные направления.

Пример 3. Рассмотрим пример для операции добавления ребра. Для промежуточного результата распознавания формулы на рис. 4 добавление ребра (x, y) принимает вид:

$$F + (x, y) = \Big\{ \Big(\{x, 2, y\}, \{(x, 2, (1, -1)), (x, y, (1, 1)|(1, 0))\} \Big) \Big\} \bigcup \Big\{ \Big(\{x, 2, y\}, \{(x, 2, (1, -1)|(1, 0)), (x, y, (1, 1)|(1, 0))\} \Big) \Big\} = \Big\{ \Big(\{x, 2, y\}, \{(x, 2, (1, -1)|(1, 0)), (x, y, (1, 1)|(1, 0))\} \Big) \Big\} = F'.$$

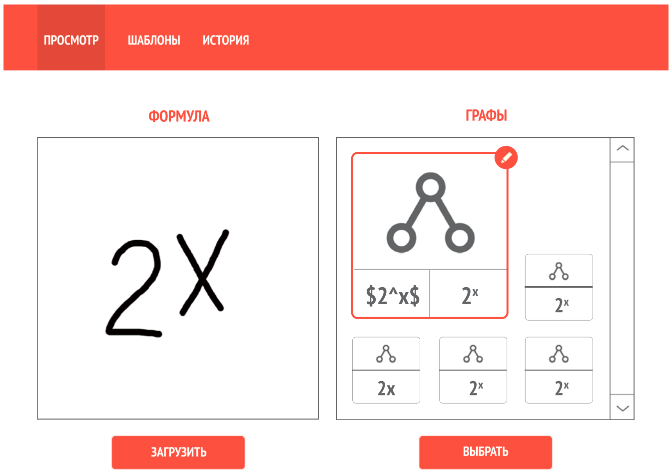

Рис. 5. Страница просмотра формулы и множества графов изображения формулы

Пример 4. Рассмотрим пример для операции добавления направления ребра. Для промежуточного результата распознавания формулы в пример 3 добавление направления $(1,1)$ для ребра (x,y) принимает вид:

$$F' + (x, y, (1,1)) = \Big\{ \Big(\{x, 2, y\}, \{(x, 2, (1,-1)|(1,0)), (x, y, (1,1))\} \Big) \Big\} = F''.$$

Если еще раз применить операцию добавления направления, то получим следующий результат:

$$F'' + (x, 2, (1,-1)) = \Big\{ \Big(\{x, 2, y\}, \{(x, 2, (1,-1)), (x, y, (1,1))\} \Big) \Big\} = F'''.$$

Итоговый результат F''' содержит единственный вариант распознавания, который является верным вариантом. Таким образом, функция, использующая одну операцию добавления ребра и две операции добавления направления ребра, является идеальной функцией уточнения для промежуточного результата распознавания F.

Замечание 3. Любое конечное множество графов изображений формул после нескольких (конечного числа) применений операций добавления ребра можно сделать моноструктурированным. Это следует из того, что операция удаляет разные ребра из графов и добавляет всегда одинаковое ребро во все графы.

Рассмотренных операций достаточно для того, чтобы построить идеальную функцию уточнения. Задача исправления результатов распознавания рукописных математических формул сводится к задаче построения наименьшей функции, т. е. такой функции, которая использует наименьшее количество операций. Для удобства работы пользователя интерфейс программы должен предоставлять возможность быстрого построения функции, которая за наименьшее число шагов из большого количества (возможно, порядка 1000) вариантов распознавания позволяет выбрать правильный итоговый результат.

§ 5. Описание интерфейса

Разработка пользовательских интерфейсов строится таким образом, чтобы создать его максимально привлекательным и удобным для оптимизации его взаимодействия с пользователем. Так, был спроектирован интерфейс, максимально упрощающий процесс проверки и исправления результатов распознавания рукописной формулы. Функциональное назначение интерфейса состоит в следующем:

1. Просмотр графа изображения формулы и текстового представления формулы.

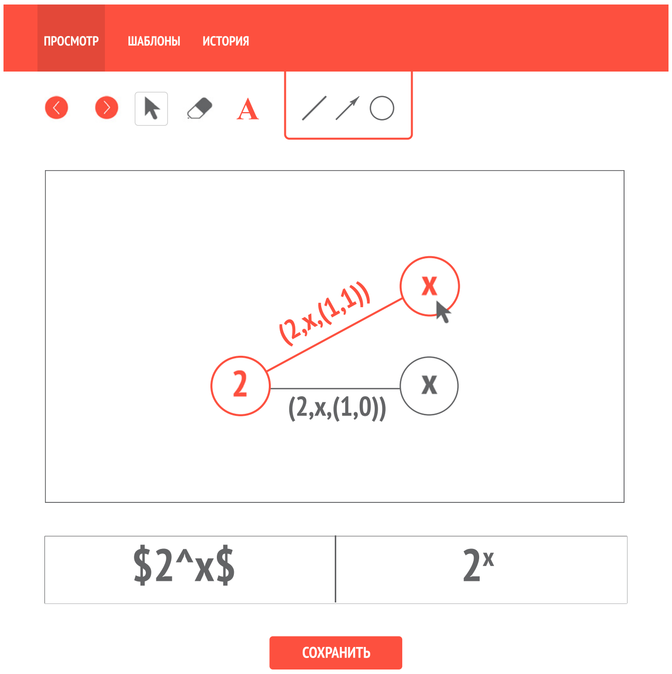

Рис. 6. Страница редактирования графа изображения формулы

2. Выбор правильного варианта распознавания из предложенных графов.

3. Редактирование графа изображения формулы.

4. Просмотр и редактирование шаблонов.

5. Просмотр истории распознавания, предполагающий поиск близких по структуре формул из прошлых результатов.

Пользователь загружает изображение с рукописной формулой, а программа выполняет ее распознавание. На выходе строится множество графов и их текстовое представление. Страница содержит два окна, одно с загруженной пользователем формулой, а другое — со списком из миниатюр. Каждая миниатюра состоит из графа изображения формулы, текстового представления и скомпилированного результата формулы. При выборе соответствующего варианта из предложенных появляется возможность редактирования графа вручную, что предполагает изменение характера связей между вершинами. Если из множества графов был найден правильный вариант распознавания, следует выделить его и нажать на кнопку «Выбрать» (рис. 5). Окно редактирования представляет собой поля для редактирования, конечного результата в виде текстового представления и панель инструментов, состоящую из кнопок «Отменить», «Повторить», указателя, ластика, инструмента «Текст» и фигур, необходимых для построения графа. Стоит отметить, что имеется возможность автоматического выбора инструмента: в зависимости от того, где именно находится указатель, выполняются разные операции. Например, если указатель мыши расположен на ребре графа, то при помощи перетаскивания изменяется конец графа, а если на вершине — то изменяется вектор направления ребра. По достижению нужного результата редактирования графа следует сохранить его, нажав на соответствующую кнопку (рис. 6). Кроме того, существует страница для шаблонов, которые представлены в виде графа изображения формулы и текстового представления. Здесь можно добавить и отредактировать шаблон, после чего «Сохранить» (рис. 7). На странице «История» можно осуществить поиск близких по структуре формул из прошлых результатов с помощью различных вариантов сортировки истории (рис. 8).

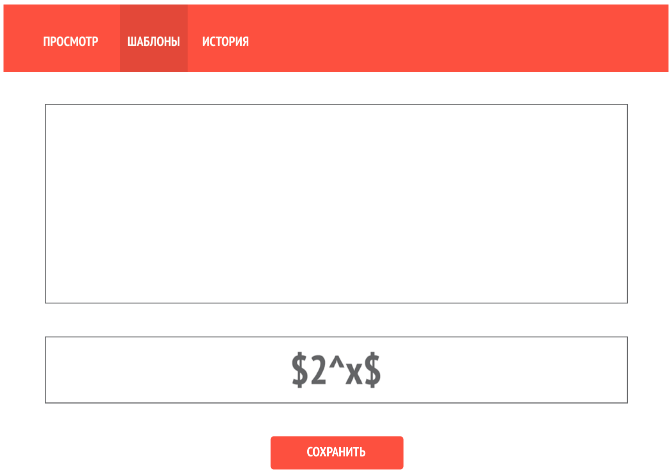

Рис. 7. Страница шаблонов

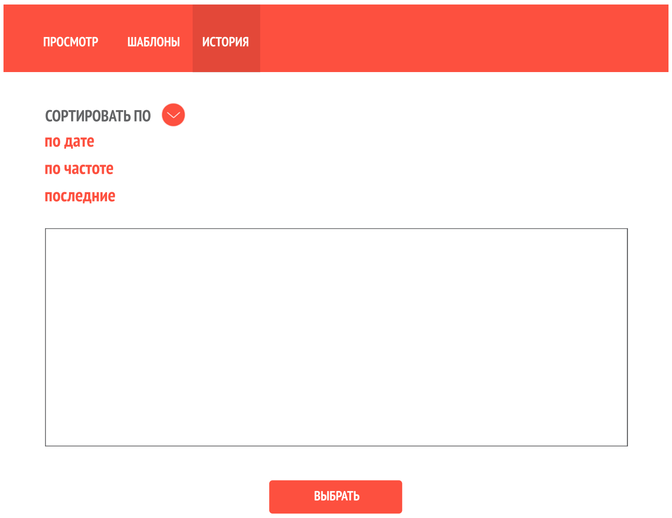

Рис. 8. Страница истории распознавания

СПИСОК ЛИТЕРАТУРЫ

1. Сапаров А.Ю., Бельтюков А.П. Математическое моделирование изображений формул с целью их распознавания // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2013. Вып. 1. С. 153–167.
2. Сапаров А.Ю., Бельтюков А.П. Применение регулярных выражений в распознавании математических текстов // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2012. Вып. 2. С. 63–73.
3. Зыков А.А. Основы теории графов. М.: Наука, 1986. 381 с.
4. Карпов Ю.Г. Теория автоматов. СПб.: Питер, 2003. 208 с.

Поступила в редакцию 01.12.2015

Сапаров Алексей Юрьевич, к. т. н., старший преподаватель, кафедра теоретических основ информатики, Удмуртский государственный университет, 426034, Россия, г. Ижевск, ул. Университетская, 1.
E-mail: say.saplh@gmail.com

Широбокова Ирина Юрьевна, магистрант, кафедра теоретических основ информатики, Удмуртский государственный университет, 426034, Россия, г. Ижевск, ул. Университетская, 1.
E-mail: semplice1993@gmail.com

A. Yu. Saparov, I. Yu. Shirobokova

User interface development to manage the process of handwritten mathematical formula recognition

Keywords: mathematical formula recognition, preliminary result of recognition, correct final result of recognition, regular tree of mathematical expression image.

MSC: 05C20, 68R10

The article deals with the problem of recognition of handwritten mathematical formulae. It is reported about the main problems that arise in solving this task. Also it is specially noted the method for preventing and correcting recognition errors based on hand process of recognition. Much attention is given to mathematical models of the proposed method. They are based on graph theory. For this purpose, the definition of a regular tree for the image of formula is given it allows to keep all options for recognition of the original image of the formula in the most compact form and to simplify the process of editing a variety of options of the group recognition linked to the changing nature of the relationship between the vertices of graph. An example of user-friendly interface to manage the recognition and correction of errors is given that doesn't require user knowledge of the format of mathematical formulae.

REFERENCES

1. Saparov A.Yu., Beltyukov A.P. Mathematical modeling of formula images for their recognition, *Vestn. Udmurt. Univ. Mat. Mekh. Komp'yut. Nauki*, 2013, no. 1, pp. 153–167 (in Russian).
2. Saparov A.Yu., Beltyukov A.P. Regular expressions in the mathematical text recognition problem, *Vestn. Udmurt. Univ. Mat. Mekh. Komp'yut. Nauki*, 2012, no. 2, pp. 63–73 (in Russian).
3. Zykov A.A. *Osnovy teorii grafov* (Foundations of the graphs theory), Moscow: Nauka, 1986, 381 p.
4. Karpov Y.G. *Teoriya avtomatov* (Automata theory), St. Petersburg: Piter, 2003, 208 p.

Received 01.12.2015

Saparov Aleksei Yur'evich, Candidate of Engineering, Senior Lecturer, Department of Theoretical Foundations of Computer Science, Udmurt State University, ul. Universitetskaya, 1, Izhevsk, 426034, Russia.
E-mail: say.saplh@gmail.com

Shirobokova Irina Yur'evna, Master Student, Department of Theoretical Foundations of Computer Science, Udmurt State University, ul. Universitetskaya, 1, Izhevsk, 426034, Russia.
E-mail: semplice1993@gmail.com