

## Восстановление логической структуры таблиц из неструктурированных текстов на основе логического вывода\*

А. О. ШИГАРОВ

*Институт динамики систем и теории управления СО РАН, Иркутск, Россия*  
e-mail: shigarov@icc.ru

Обсуждаются вопросы автоматизации процессов структурирования табличной информации, изначально представленной в неструктурированном виде. Предлагается подход к восстановлению логической структуры таблиц (т.е. отношений её элементов) с использованием продукционной системы исполнения правил. Рассматриваются модель таблицы и алгоритмы обработки табличной информации, ориентированные на логический вывод. На основе данного подхода, модели и алгоритмов разработана система структурирования табличной информации, представленной в формате табличного процессора. Полученные с помощью предложенной системы экспериментальные результаты показывают эффективность применения предлагаемого подхода для широкого класса таблиц.

*Ключевые слова:* анализ и обработка табличной информации, извлечение информации из таблиц, обработка неструктурированной информации.

### Введение

В настоящее время многие исследователи в области управления данными (см., например, [1, 2]) отмечают важность проблем управления и интеграции неструктурированной информации. Под термином *неструктурированная информация*, или *неструктурированные данные* (англ. *unstructured information/data*), как правило, понимается любая информация, которая не имеет предопределённой формальной модели данных или не укладывается в таблицы реляционной базы данных [1–3]. При этом выделяют *неструктурированную текстовую* (файлы текстового или табличного процессора, сообщения электронной почты и др.) и *не текстовую* (изображения, аудио и видео файлы и др.) *информацию*.

Для обозначения неструктурированной текстовой информации иногда используются термины *слабоструктурированные документы* (англ. *weakly structured documents*) — включают типографские индикаторы структуры для идентификации заголовков, абзацев, списков, таблиц и т.д. (например, ASCII-текст, файлы печати PDF), и *полуструктурированные документы* (англ. *semi-structured documents*) — включают некоторые структурные и содержательные метаданные (например, документы Word, книги Excel, HTML страницы). Подробное обсуждение этих терминов дано в работе [4].

---

\*Работа выполнена при финансовой поддержке РФФИ (гранты № 12-07-31051 и 14-07-00166) и Совета по грантам Президента РФ СП-3387.2013.5.

Слабо- и полуструктурированные документы могут содержать таблицы. Поскольку в таких таблицах отсутствует схема данных, предназначенная для высокоуровневой машинной обработки (например, для выполнения запросов к данным по аналогии с SQL), то в указанном выше смысле они также являются примером неструктурированной текстовой информации. По аналогии с определением неструктурированной текстовой информации, данные, представленные в виде таких таблиц, можно называть *неструктурированной табличной информацией*. Следует отметить, что этот термин (англ. *unstructured tabular information/data*) в данном смысле также используется в некоторых работах, например в патентах [5–7].

Автоматизация процессов преобразования неструктурированной табличной информации к структурированному виду имеет важное прикладное значение в задачах управления данными, информационного поиска и извлечения информации, анализа и обработки документов. В литературе встречаются следующие задачи, которые можно рассматривать как приведение неструктурированной табличной информации к структурированному виду:

1) *каноникализация таблицы* (англ. *table canonicalization*) [8, 9] — приведение её к канонической форме, которая структурно соответствует таблице реляционной базы данных;

2) *понимание таблицы* (англ. *table understanding*) [10] — состоит в восстановлении отношений между метками (заголовками) и значениями данных, а также между метками и измерениями (доменами);

3) *извлечение информации из таблицы* (англ. *information extraction from table*) [10] — является аналогом задачи извлечения информации из текста и состоит в извлечении фактов, формирующих целевую базу данных.

Сложность анализа и обработки неструктурированной табличной информации обусловлена большим разнообразием форм представления и изображения таблиц. В зависимости от их исходного представления необходимо решать следующие задачи: 1 — *обнаружение таблицы* внутри документа; 2 — *восстановление физической структуры таблицы*, т. е. позиций и содержания её ячеек; 3 — *восстановление логической структуры таблицы*, т. е. отношений (смысловых значений) её элементов.

Известные методы и системы анализа и обработки таблиц базируются на различных подходах. Так, метод обнаружения таблиц в ASCII-тексте и изображениях [11] основан на динамическом программировании. В работе [12] рассматривается извлечение информации из таблиц с использованием машинного обучения. Пример использования вероятностных моделей для обнаружения таблиц и классификации их строк приводится в [13]. Однако для большинства из них при построении алгоритмов анализа характерно использование предположений (часто достаточно сильных) о структурах таблиц. Такие предположения (знания) оказываются встроенными в известные алгоритмы анализа и тем самым ограничивают классы эффективно (точно и полно) обрабатываемых ими таблиц. Современное состояние данной области исследований пока не позволяет говорить о полном решении проблем структурирования табличной информации. В работе [14] показано, что большинство исследований в этом направлении посвящено решению проблем низкоуровневой обработки табличной информации — обнаружению и сегментации (разделению на строки, столбцы и ячейки) таблиц из растровых изображений документов. При этом вопросы восстановления логической структуры таблиц остаются менее изученными.

Наиболее близкими к представленному здесь исследованию являются работы [8, 9], где рассматривается преобразование (структурирование) табличной информации, называемое каноникализацией таблицы. В [8] предлагается метод интерпретации и каноникализации таблиц, которые содержатся в спецификациях, используемых в строительной промышленности. Для этого предлагается применять обработку естественного языка на основе онтологии предметной области (подъязыка спецификаций строительной промышленности). Способ каноникализации, предлагаемый в работе [9], основан на использовании библиотеки фреймов, содержащей знания о лексическом содержании таблиц. Каждый фрейм данных описывает один тип данных и применяется для отнесения выражений на естественном языке (табличных заголовков и значений) к этому типу. Для описания типов данных в [9] предлагается использовать регулярные выражения, словари и некоторые открытые ресурсы, например WordNet (<http://wordnet.princeton.edu>). Указанные исследования ориентированы на обработку только естественно-языкового содержания таблицы. На практике этого не всегда достаточно. Для более точного и полного извлечения информации из таблицы часто также требуется анализ пространственной и графической информации.

Настоящая работа посвящена вопросам восстановления логической структуры таблицы. Для структурирования табличной информации авторами предлагается подход, основанный на исполнении правил анализа структуры таблиц. При этом база фактов, используемая в процессе логического вывода, может включать информацию о пространственном, графическом и естественно-языковом содержании таблицы. В работе также описана система CELLS для структурирования табличной информации, основанная на предлагаемом подходе. В текущем состоянии эта система позволяет структурировать данные из таблиц, представленных в формате табличного процессора Excel. При этом предполагается, что исходные таблицы имеют аккуратную физическую структуру (композицию ячеек), совпадающую с их разграфкой, и дополнительную разметку для определения местоположения таблиц внутри листа, что позволяет избежать этапов обнаружения и сегментации исходных таблиц и сразу перейти к восстановлению их логической структуры. Представленные в работе экспериментальные результаты показывают возможности применения предлагаемой системы в задачах массового ввода неструктурированной табличной информации в базы данных.

Работа является продолжением исследований, посвящённых проблемам анализа и обработки таблиц, которые проводятся в ИДСТУ СО РАН на протяжении нескольких последних лет [15–18] и в которых, в частности, рассматриваются методы обнаружения таблиц внутри файлов печати [15, 16], анализа таблиц, представленных в виде ASCII-текста [17], а также алгоритмы формирования канонической формы таблицы по отношению её элементов [18].

## 1. Класс обрабатываемых таблиц

Большое количество неструктурированной табличной информации размещено в современных широко распространённых форматах данных, таких как Excel, Word, HTML и LaTeX. При этом возможности и ограничения представления таблиц в этих форматах во многом похожи, что позволяет привести следующую информацию о таблице:

- 1) позиции ячейки в координатах столбцов и строк;
- 2) объединения ячеек (например, атрибуты COLSPAN и ROWSPAN в HTML);

3) стили ячеек (оформление границ, фон, размещение содержания, шрифтовые метрики и др.);

4) содержание ячеек (текст, рисунки и др.).

Вместе с тем каждый из этих форматов имеет свои особенности. Так, в таблицах Word, HTML и LaTeX внутри ячеек могут содержаться другие таблицы, а в формате Excel это не допускается. В Excel для ячейки может быть определён один из простых типов данных (NUMERIC, DATE, STRING и др.). В HTML могут быть определены связи и роли ячеек с помощью атрибутов HEADERS, SCOPE тегов TD и TH.

По аналогии с терминологией, употребляемой в области анализа и распознавания документов [19, 20], в настоящей работе используются термины *физическая* и *логическая структура таблицы* применительно к табличной информации, имеющей высокоуровневое представление в форматах данных Excel, Word, HTML и LaTeX. При этом подразумевается, что физическая структура описывает читаемый (визуальный) состав таблицы, а логическая — её смысловой состав. Базовые ограничения обрабатываемых таблиц, которые основаны на перечисленных наблюдениях за представлением неструктурированной табличной информации в этих форматах данных, можно рассматривать отдельно для физической и логической структуры.

В настоящем исследовании используется следующий ряд базовых предположений о физической структуре таблицы.

1. Ячейка характеризуется позициями в координатах столбцов и строк, стилем оформления и содержанием.

2. Ячейка может располагаться на нескольких подряд идущих строках и столбцах, т. е. занимать несколько плиток сетки, которые всегда образуют прямоугольник, как показано на рис. 1, а.

3. Ячейка может содержать только текст, хотя на практике она может иметь и более богатое содержание, например, RTF (Rich Text Format), изображение, формулу (в Excel), что в данном исследовании не учитывается в целях упрощения реализации структур данных и алгоритмов развиваемой системы CELLS. Следует отметить, что ячейка не может содержать другие таблицы или ячейки.

Класс обрабатываемых таблиц также ограничен следующими базовыми предположениями о логической структуре таблицы.

1. Содержание ячейки является либо вхождением, либо меткой. Вхождение представляет значение данных, а метка адресует (описывает) вхождение. Используемые

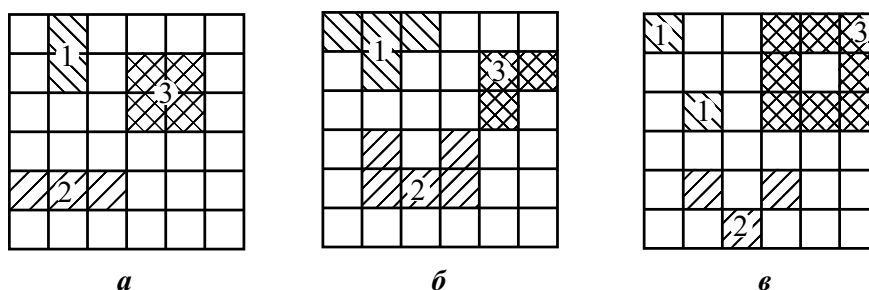


Рис. 1. Примеры объединения плиток сетки в ячейки таблицы, обозначенные как 1, 2 и 3: а — ячейка может объединять несколько плиток в Excel, Word, HTML и LaTeX; б — ячейка может визуально (для восприятия человеком) включать несколько плиток с помощью разграфки; в — скорее всего, таким образом ячейки никто не представляет

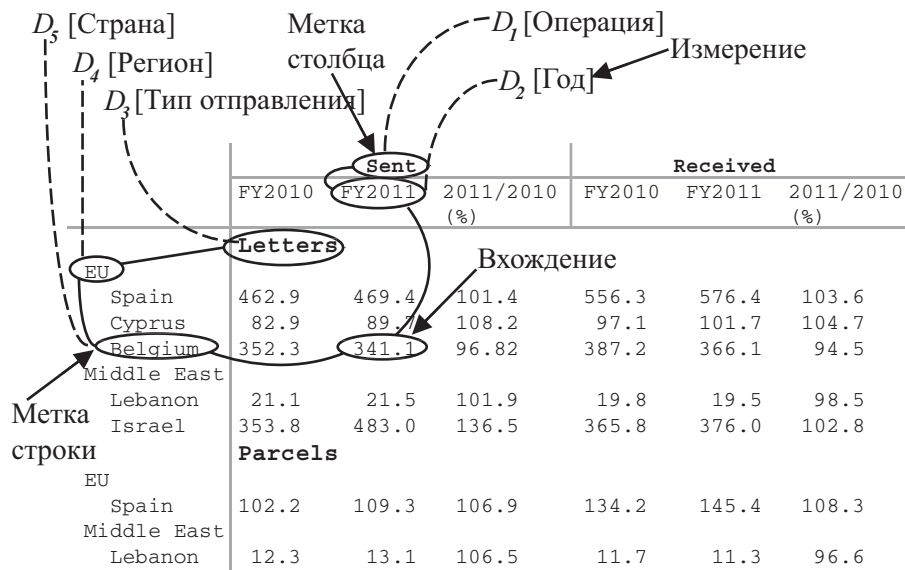


Рис. 2. Логическая структура таблицы

в настоящей работе термины *вхождение* и *метка* соответствуют смыслу терминов *entry* и *label* из работы [21].

2. Метка может адресовать (описывать) вхождения и другие метки либо только в строках, либо только в столбцах.

3. Метки могут образовывать иерархические отношения между собой.

4. Метки могут являться значениями измерений.

Пример с фрагментом логической структуры таблицы представлен на рис. 2.

Необходимо отметить, что таблица может иметь контекст, а её вхождения и метки могут характеризоваться представленной в нем информацией. Например, время, место и другие измерения, которые адресуют табличные данные, могут быть представлены в контексте. Однако в данном исследовании контекст как часть табличной информации не рассматривается.

## 2. Модель таблицы CELLS

На основе принятых ограничений табличной структуры предлагается достаточно общая модель таблицы CELLS, ориентированная на представление фактов о табличной информации в процессе логического вывода. Модель включает два уровня: физической и логической структуры, которые в упрощённом виде можно описать следующим образом.

1. Уровень физической структуры  $T_p = (S_r, S_c, C)$  состоит из

- пространства строк  $S_r$  и столбцов  $S_c$ ;
- набора ячеек  $C$ , в котором каждая ячейка  $c = (p, c', G)$  включает
  - координаты в пространстве строк  $S_r$  и столбцов  $S_c$  ( $c_l$  — левая,  $r_t$  — верхняя,  $c_r$  — правая и  $r_b$  — нижняя границы соответственно)  $p = (c_l, r_t, c_r, r_b)$ ,
  - содержание  $c'$ ,
  - графическое форматирование (цветовые схемы, шрифтовые метрики, выравнивание, стили оформления границ и др.)  $G$ .

2. Уровень логической структуры  $T_l = (D, L_r, L_c, E)$  состоит из

- набора представленных в обрабатываемой таблице измерений  $D = \{D_i\}$ , каждое из которых содержит значения  $D_i = \{d_j\}$ ;
- дерева меток строк  $L_r$  и столбцов  $L_c$ , отражающих связи между метками, не являющимися значениями измерений  $D_i$  из набора  $D - l = (l')$ , где  $l'$  — содержание метки;
- набора вхождений  $E$ , в котором каждое вхождение  $e = (e', D', L')$  включает
  - содержание  $e'$ ,
  - набор связанных с ним значений измерений  $D_i$  из набора  $D - D'$ ,
  - набор связанных с ним меток из деревьев  $L_r$  и  $L_c - L'$ .

### 3. Представление и исполнение правил анализа табличной структуры

Основная идея предлагаемого подхода состоит в следующем. Обычно внутри тематической коллекции документов от одного поставщика таблицы компонуются и форматируются однообразно. Для такой коллекции документов можно определить набор формализованных правил анализа табличной структуры, который удовлетворяет всем или почти всем её таблицам. Эти правила можно представить в виде базы знаний, а процесс восстановления логической структуры таблицы — реализовать как логический вывод. Схема структурирования неструктурированной табличной информации показана на рис. 3.

Уровень физической структуры в модели CELLS формируется в результате обнаружения внутри источника и восстановления физической структуры исходной таблицы, представленной в неструктурированном виде. В настоящей работе эти этапы обработки не обсуждаются. Предполагается, что они выполняются в сторонних системах. Полученная в результате физическая структура таблицы составляет входные данные, которые формируют базу фактов для логического вывода. Кроме того, факты могут быть дополнены внешней информацией об измерениях.

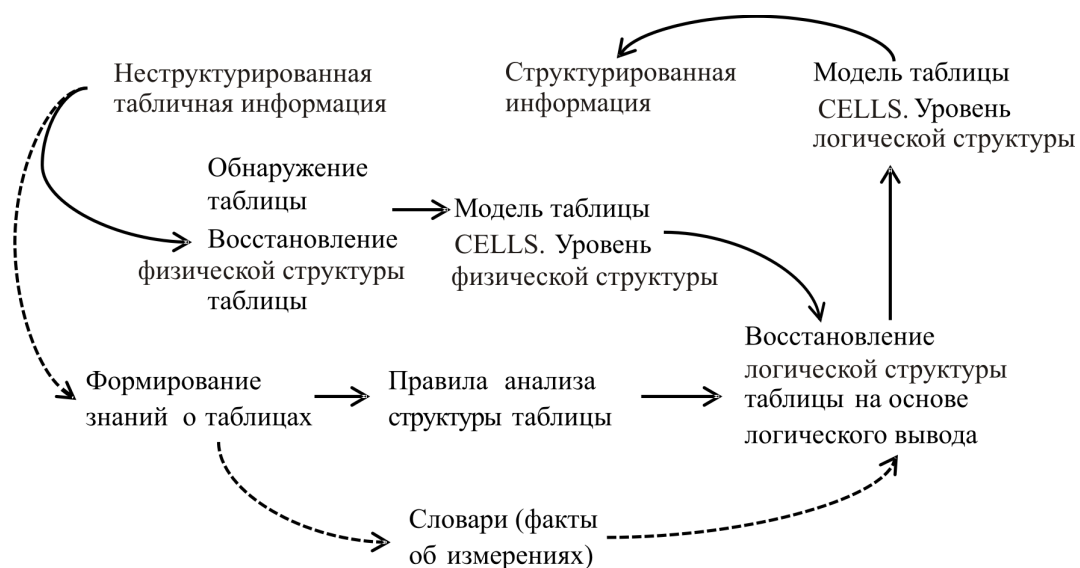


Рис. 3. Схема структурирования табличной информации

Продукционные правила анализа табличной структуры записываются на языке выражений MVEL [22]. Они отображают доступную информацию — позиции (координаты), графическое форматирование и естественно-языковое содержание ячеек — в отсутствующие изначально отношения между метками, вхождениями и измерениями. Логический вывод для таких правил может выполняться в свободной системе исполнения правил Drools Expert [23]. Полученные в процессе вывода новые факты о логической структуре таблицы должны быть достаточными для её каноникализации.

Далее приводится ряд примеров возможных правил анализа структуры на языке MVEL.

**Пример 1.** Если ячейка `$c` находится в первом столбце `c1 == 1`, то она выполняет роль метки строки `modify ( $c ) { setRole( Role.ROWLABEL ) }`.

```
...
when
    $c : CCell( c1 == 1 )
then
    modify ( $c ) { setRole( Role.ROWLABEL ) }
...

```

**Пример 2.** Если ячейка `$c1` расположена непосредственно над ячейкой `$c2` и при этом полностью охватывает её по столбцам, то ячейки связаны `$c1.addConnectedCell( $c2 )`.

```
...
when
    $c1 : CCell()
    $c2 : CCell( rt == $c1.rb + 1,
                ( $c1.c1 <= c1 && cr < $c1.cr ) ||
                ( $c1.c1 < c1 && cr <= $c1.cr ) )
then
    $c1.addConnectedCell( $c2 )
...

```

**Пример 3.** Если ячейка `$c` расположена полностью в первом столбце и содержит текст, удовлетворяющий регулярному выражению `"(?i).*(total)"`, то при формировании выходных данных её необходимо игнорировать.

```
...
when
    $c : CCell( c1 == 1, c1 == cr, text matches "(?i).*(total)" )
then
    modify ( $c ) { setIgnored( true ) }
...

```

**Пример 4.** Если есть измерение `$d` с названием "Religion", а ячейка `$c` содержит некоторый текст и его шрифт имеет красный (`"#ff0000"`) цвет, при этом все остальные ячейки в той же самой строке (`rt == $c.rt`) не содержат текст, то ячейка `$c` связана с измерением `$d`.

```
...
when
    $d : CDimension( name == "Religion" )

```

```

    $c : CCell ( text != null, style.getFont().getColor() == "#ff0000" )
    not ( exists CCell ( rt == $c.rt, text != null ) )
then
    $c.setDimension( $d )
...

```

**Пример 5.** Если ячейка `$e` содержит вхождение `role == Role.ENTRY` и находится в одном столбце с ячейкой `$l`, содержащей метку столбца `role == Role.COLLABEL`, то ячейки связаны `$e.addConnectedCell( $l )`.

```

...
when
    $l : CCell( role == Role.COLLABEL )
    $e : CCell( role == Role.ENTRY, cl == $l.cl, cr == $l.cr )
then
    $e.addConnectedCell( $l )
...

```

Примеры правил, которые применялись при тестировании системы CELLS, можно найти по адресу <http://cells.icc.ru/test>.

#### 4. Дополнительные алгоритмы структурирования табличной информации

Кроме логического вывода, рассматриваемое структурирование табличной информации опирается на ряд дополнительных алгоритмов. По порядку выполнения относительно логического вывода их можно разделить на алгоритмы предобработки и постобработки.

В предобработку входят опционально: 1) удаление лишних пробельных и служебных символов из текстового содержания, 2) исключение из таблицы пустых строк и столбцов и восстановление отсутствующих настроек стилей границ ячеек. Последнее необходимо, поскольку видимые и физические границы ячейки не всегда совпадают. Визуально они могут быть образованы границами соседних ячеек. Приведение стилей физических границ ячеек в соответствие с её видимыми границами позволяет упростить правила анализа структуры таблицы.

В процессе логического вывода накапливается информация о логической структуре таблицы. Для этой информации выполняется постобработка, которая включает приведение текстового содержания ячеек к эталонным написаниям, сопоставление меток с измерениями и формирование канонической формы таблицы. Далее приводится краткое описание этих алгоритмов.

Метки внутри одной или нескольких таблиц могут различаться по написанию, но иметь одно лексическое значение, т.е. являться синонимами. Например, метки «2010», «FY 2010», «Year 2010», «Previous Year», «2010 г.» и «Текущий год» могут быть синонимами, означающими 2010 год. В качестве их эталона может использоваться лексическое значение «2010». Такие метки заменяются соответствующими эталонами путём сопоставления со словарем, который содержит набор отношений вида  $(S, R)$ , где  $S$  — регулярное выражение для идентификации синонимов,  $R$  — соответствующий эталон, заданный также в виде регулярного выражения. Например, если в словаре задать отношения вида («FY[2][0][0-1][0-3]», «[2][0][0-1][0-3]»), то все заголовки, соответствующие



регулярному выражению «FY[2][0][0-1][0-3]», т.е. «FY2000»,..., «FY2013», будут заменены на соответствующие эталоны «2000»,..., «2013».

Для сопоставления меток с измерениями используется словарь, содержащий набор отношений вида  $(S, D_i)$ , где  $S$  — регулярное выражение для идентификации значений измерения,  $D_i$  — соответствующее измерение. Следует отметить, что словари эталонов и измерений могут также использоваться в процессе логического вывода.

В процессе такого сопоставления из деревьев меток  $L_r$  и  $L_c$  исключаются метки, которые являются значениями измерений  $D_i$  из набора  $D$ , как показано на рис. 4. При этом место, занимаемое исключаемой меткой в дереве  $L_r$  или  $L_c$ , переходит поддереву вложенных в неё меток. Связи вхождения с исключаемой меткой заменяются связя-

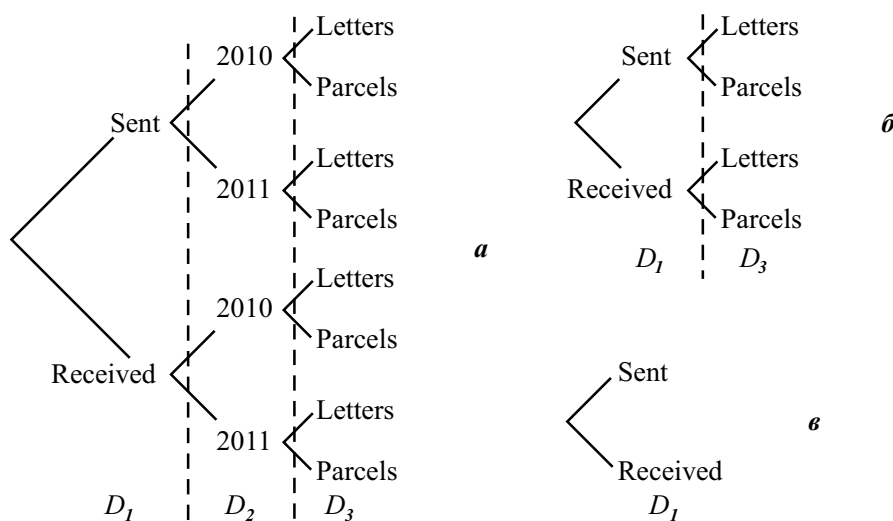


Рис. 4. Редуцирование дерева меток при их сопоставлении со значениями измерений: а — без восстановления измерений; б — восстановлено измерение  $D_2 = \{2010, 2011\}$  «Год»; в — восстановлены измерения  $D_2$  «Год» и  $D_3 = \{Letters, Parcels\}$  «Тип отправления»

Данные	Операция	Год	Тип отправления	Регион	Страна
462.9	Sent	2010	Letters	EU	Spain
82.9	Sent	2010	Letters	EU	Cyprus
...	...	...	...	...	...
12.3	Sent	2010	Parcels	Middle East	Lebanon
469.4	Sent	2011	Letters	EU	Spain
89.7	Sent	2011	Letters	EU	Cyprus
341.1	Sent	2011	Letters	EU	Belgium
21.5	Sent	2011	Letters	Middle East	Lebanon
483.0	Sent	2011	Letters	Middle East	Israel
109.3	Sent	2011	Parcels	EU	Spain
13.1	Sent	2011	Parcels	Middle East	Lebanon
556.3	Received	2010	Letters	EU	Spain
11.3	Received	2011	Parcels	Middle East	Lebanon

Рис. 5. Каноническая форма таблицы из рис. 2: все метки сопоставлены измерениям, поэтому поля COL\_LABEL и ROW\_LABEL отсутствуют

ми этих вхождений с соответствующим значением измерения  $D_i$ . В идеальном случае, когда каждая метка соотнесена с некоторым измерением, деревья меток вырождаются.

Из восстановленной информации модели таблицы CELLS формируется таблица в канонической форме, которая включает следующие поля:

- DATA — данные (вхождения);
- ROW\_LABEL — пути меток от листьев до корней из невырожденного дерева  $L_r$ ;
- COL\_LABEL — пути меток от листьев до корней из невырожденного дерева  $L_c$ ;
- $D_1, \dots, D_N$  — поля значений измерений  $D_i$  из набора  $D$ .

Каждый кортеж в такой канонической форме представляет связь между вхождением, путями в деревьях меток и значениями восстановленных измерений. Дополнительно поле ROW\_LABEL/COL\_LABEL может быть разделено на несколько отдельных полей, каждое из которых будет соответствовать одному уровню вложенности в дереве меток строк/столбцов. Пример канонической формы обработанной таблицы приводится на рис. 5 (см. выше). Сформированная каноническая таблица может экспортироваться в реляционную базу данных с помощью стандартных средств известных систем управления базами данных.

## 5. Экспериментальные результаты

Экспериментальная оценка рассмотренного подхода выполнена с помощью системы CELLS, в которой реализованы структуры данных, представляющие модель таблицы CELLS, и алгоритмы 1) загрузки исходной табличной информации в формате Excel (тестовых данных со специальной разметкой), 2) структурирования табличной информации, восстановленной в процессе логического вывода, 3) экспорта результатов в формате Excel. Все предложенные структуры данных и алгоритмы реализованы для платформы исполнения Java.

Для экспериментальной оценки сформирована коллекция тестовых данных, включающая 97 таблиц в формате Excel, собранных из семи различных источников. Коллекция доступна по адресу <http://cells.icc.ru/test>. Её краткое описание приводится в таблице “Тестовые данные и экспериментальные результаты”.

Источниками тестовых данных были слабоструктурированные документы в низкочастотном формате файлов печати PDF — государственные и финансовые статистические отчёты со значительным табличным содержанием. Для формирования коллекции исходная табличная информация преобразована из формата PDF в Excel. При этом, насколько это возможно, в полученных тестовых таблицах было сохранено графическое форматирование, представленное в соответствующих им PDF-источниках.

Тестовые данные имеют дополнительную разметку для определения местоположения таблицы внутри листа Excel. Верхний левый угол тестовой таблицы обозначен маркером «\$START», а нижний правый — маркером «\$END». Кроме того, тестовые таблицы имеют аккуратную декомпозицию на ячейки, т. е. там, где это возможно, их физическая структура и разграфка совпадают, что позволяет избежать этапы обнаружения и сегментации таблицы.

В эксперименте оценивается восстановление меток, вхождений и внутренних связей между метками. Оценка восстановления внешних связей между метками и измерениями не производится. Следует отметить, что в некоторых случаях интерпретация ролей и связей таблицы не всегда очевидна даже для специалиста. Отвечая на вопросы: есть ли связь между двумя ячейками или нет, является ли содержимое ячейки вхождением

Тестовые данные и экспериментальные результаты

Источник	Количество						Время исполнения правил, мс
	таблиц	ячеек	вхож- дений	меток	связей между метками <sup>8</sup>	правил	
JAPAN_STAT <sup>1</sup>	15	1088	734	257	102	10	417
AEROFLOT <sup>2</sup>	13	2047	727	321	167	16	526
BOEING <sup>3</sup>	21	2156	964	470	196	14	663
CHINA_STAT <sup>4</sup>	18	7216	4180	862	551	12	964
CHEVRON <sup>5</sup>	7	812	268	141	89	12	283
USDA_NASS <sup>6</sup>	7	1553	1175	313	174	16	638
TOBACCO <sup>7</sup>	16	2844	2195	508	335	10	730

<sup>1</sup> Statistical Handbook of Japan 2007. Statistics Bureau of Japan. Chapter 5, 8.

<sup>2</sup> OJSC «Aeroflot – Russian Airlines» Consolidated Financial Statements For the Year Ended December 31, 2006. P. 4–10, 25–26.

<sup>3</sup> Boeing Co, Annual Report 2010. P. 50–55, 83–85.

<sup>4</sup> China statistical yearbook 2003. National Bureau of Statistics of China. P. 23–48, 555, 559, 571, 584, 590, 664, 708, 774, 765.

<sup>5</sup> Chevron Corp. News Release November 2, 2012. Chevron Corp. P. 1, 5–9.

<sup>6</sup> USDA NASS. 2003 Agricultural Statistics Annual. USDA (U.S. Department of Agriculture). National Agricultural Statistics Service. Chapter VI. P. 5–7, 12.

<sup>7</sup> Tobacco: World Markets and Trade 2005. USDA (U.S. Department of Agriculture). Foreign Agricultural Service.

<sup>8</sup> Исключая связи корней деревьев меток.

или меткой, два специалиста могут прийти к разным решениям. В тестовых данных в некоторых случаях авторы также не нашли однозначного решения: каким образом интерпретировать их. Проведённая экспериментальная оценка основана на предположении о том, что использованная в процессе тестирования интерпретация таких случаев является правильной. С учётом этого все метки, вхождения и внутренние связи восстановлены (обнаружены) 1 — *полно*, т. е. нет таких меток, вхождений и внутренних связей, которые присутствуют в тестовых таблицах, но отсутствуют в результатах обработки, и 2 — *точно*, т. е. нет таких меток, вхождений и внутренних связей, которые отсутствуют в тестовых таблицах, но присутствуют в результатах обработки.

Полученные экспериментальные результаты приводятся в таблице. Логический вывод выполнялся в системе Drools Expert (5.4.0.Final). При этом использовался процессор Intel Core 2 Quad, 2,66 ГГц. Экспериментальные результаты показывают эффективность применения предлагаемого подхода для широкого класса таблиц.

## Заключение

В работе изложен оригинальный подход к восстановлению структуры таблиц на основе логического вывода, базирующийся на предположении о том, что для одного или нескольких схожих источников можно разработать непротиворечивый набор правил анализа структур содержащихся в них таблиц. Например, для обработки таблиц из одного источника «Statistical Handbook of Japan 2007. Statistics Bureau of Japan. Chapter 5, 8» потребовалось разработать 10 правил, составленных из 93 строк кода на

языке MVEL. Предполагается, что этот набор правил подходит и для многих других таблиц из похожих источников (других глав или изданий за другие годы этого статистического справочника).

Рассмотренный подход позволяет создавать отдельные наборы правил для различных структур таблиц. Однако разработка достаточно универсальных баз знаний для многих разнородных источников имеет слишком высокую цену и не всегда возможна из-за противоречий, содержащихся в самих источниках. Поэтому данный подход предназначен в основном для задач управления данными для массового структурирования табличной информации из наборов похожих источников.

Подход взят в основу развиваемой авторами системы структурирования табличной информации CELLS. Полученные экспериментальные результаты показали эффективность её применения для широкого класса таблиц, представленных в формате Excel. Кроме того, система была успешно применена на практике. В рамках совместного российско-монгольского проекта РФФИ № 11-07-92204 с её помощью наполнена база данных по социально-экономическому положению территорий Монголии из неструктурированных текстовых источников, предоставленных Институтом национального развития Монголии (<http://mdi.gov.mn>). Для анализа этой информации потребовалось описать измерения «Время», «Территории» и «Отрасли производства». Два дополнительных измерения «Показатели» и «Меры» были сформированы из названий таблиц в процессе обработки их контекстов. Потребовалось также составить девять правил, занимающих 81 строку кода на языке MVEL. Информация извлекалась из более 40 таблиц в формате Excel. Всего в процессе обработки этих таблиц извлечено более 15 000 значений данных (вхождений).

В то же время авторам представляется перспективным дальнейшее исследование возможностей для упрощения правил анализа структуры таблицы за счёт развития структур данных представления табличной информации и дополнительных алгоритмов её предобработки и постобработки. Кроме того, поскольку при тестировании часто обнаруживаются ранее неизвестные детали, связанные с исходным представлением табличной информации, реализация структур данных и алгоритмов системы CELLS может потребовать уточнения при обработке других форматов: Word, HTML, PDF.

## Список литературы

- [1] INMON W. H., NESAVICH A. Tapping into Unstructured Data: Integrating Unstructured Data and Textual Analytics into Business Intelligence. 1st Edit. Prentice Hall. NJ, USA, 2007. 264 p.
- [2] INMON W. H. Matching unstructured data and structured data // The Data Administration Newsletter. 2006. <http://www.tdan.com/view-articles/5009>
- [3] BLUMBERG R., ATRE S. The problem with unstructured data // DM Review. 2003. [http://soquelgroup.com/Articles/dmreview\\_0203\\_problem.pdf](http://soquelgroup.com/Articles/dmreview_0203_problem.pdf)
- [4] FELDMAN R., SANGER J. The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge Univ. Press, 2006. 422 p.
- [5] KLEIN E., LACOMB C., LAYMON M. Automated Understanding and Decomposition of Table-Structured Electronic Documents. US Patent Appl. Publ. US US 2004/0193520 A1. 2004. <http://www.google.com/patents/US20040193520>
- [6] KLEIN E., LACOMB C., LAYMON M. ET AL. Automated Understanding, Extraction and Structured Reformatting of Information in Electronic Files. US Patent Appl. Publ. US 2004/0194009 A1. 2004. <https://www.google.com/patents/US20040194009>

- [7] SRINIVASAN V., KOTHIWALE M., ALAM R., BHARADWAJ S. Method for Extracting, Interpreting and Standardizing Tabular Data From Unstructured Documents. US Patent Appl. Publ. US 2006/0288268 A1. 2006. <http://www.google.com/patents/US20060288268>
- [8] DOUGLAS S., HURST M., QUINN D. Using natural language processing for identifying and interpreting tables in plain text // Proc. of the 4th Annual Symp. on Document Analysis and Information Retrieval. Las Vegas, 1995. P. 535–546.
- [9] TIJERINO Y., EMBLEY D., LONSDALE D., NAGY G. Towards ontology generation from tables // World Wide Web: Internet and Web Information Systems. 2005. Vol. 8, No. 3. P. 261–285.
- [10] EMBLEY D.W., HURST M., LOPRESTI D., NAGY G. Table-processing paradigms: a research survey // Intern. J. on Document Analysis and Recognition. 2006. Vol. 8, No. 2. P. 66–86.
- [11] HU J., KASHI R., LOPRESTI D., WILFONG G. Medium-independent table detection // Proc. of the Document Recognition and Retrieval VII. San Jose CA, USA, 2000. P. 291–302.
- [12] TENGLI A., YANG Y., MA N.L. Learning table extraction from examples // Proc. of the 20th Intern. Conf. on Comput. Linguistics. Stroudsburg, USA, 2004. Article 987.
- [13] PINTO D., MCCALLUM A., WEI X., CROFT B. Table extraction using conditional random fields // Proc. of the 26th Int. ACM SIGIR Conf. on Research and Development in Informaion Retrieval. New York, USA, 2003. P. 235–242.
- [14] E SILVA A.C., JORGE A.M., TORGO L. Design of an end-to-end method to extract information from tables // Intern. J. on Document Analysis and Recognition. 2006. Vol. 8, No. 2. P. 144–171.
- [15] БЫЧКОВ И.В., РУЖНИКОВ Г.М., ХМЕЛЬНОВ А.Е., ШИГАРОВ А.О. Эвристический метод обнаружения таблиц в разноформатных документах // Вычисл. технологии. 2009. Т. 14, № 2. С. 58–73.
- [16] SHIGAROV A.O., BUCHKOV I.V., KHMEI'NOV A.E., RUZHNIKOV G.M. A method for table detection in metafiles // Pattern Recognition and Image Analysis. 2009. Vol. 19, No. 4. P. 693–697.
- [17] ХМЕЛЬНОВ А.Е., ШИГАРОВ А.О. Метод извлечения таблиц из неформатированного текста // Вычисл. технологии. 2008. Т. 13. Спец. выпуск 1. С. 93–101.
- [18] ШИГАРОВ А.О., БЫЧКОВ И.В., РУЖНИКОВ Г.М. и др. Система трансформации таблиц // Информ. технологии и вычисл. системы. 2013. № 3. С. 15–26.
- [19] MACHINE Learning in Document Analysis and Recognition. Series: Studies in Computational Intelligence. Vol. 90 / Eds. S. Marinai, H. Fujisawa. Springer, 2008. 434 p.
- [20] ZANIBBI R., BLOSTEIN D., CORDY J.R. A survey of table recognition: models, observations, transformations, and inferences // Intern. J. on Document Analysis and Recognition. 2004. Vol. 7, No. 1. P. 1–16.
- [21] WANG X. Tabular Abstraction, Editing, and Formatting. PhD Thesis. Waterloo, Ontario, Canada, 1996.
- [22] MVEL. <http://mvel.codehaus.org>
- [23] DROOLS Expert (JBoss Community). <http://www.jboss.org/drools/drools-expert.html>

*Поступила в редакцию 2 июля 2013 г.,  
с доработки — 6 декабря 2013 г.*