Recognising Tables Using Multiple Spatial Relationships Between Table Cells

Mohamed Alkalai
School of Computer Science
University of Birmingham
Birmingham, UK
M.A.Alkalai@cs.bham.ac.uk

ABSTRACT

While much work has been done on table recognition this research has been primarily concerned with tables in ordinary text. However, tables containing mathematical structures can differ quite significantly from ordinary text tables and therefore specialist treatment is often necessary. In fact, it is even difficult to clearly distinguish table recognition in mathematics from layout analysis of mathematical formulae. This blurring is often leading to a number of possible, equally valid interpretations. However, a reliable understanding of the layout of a formula is often a necessary prerequisite to further semantic interpretation. In this paper, a new construction of table representation method is introduced which, attempts to overcome the unsolved issues mentioned in several published works. This encompasses the lack of sufficient work that deals with tables with misaligned cells. I utilise multi spatial relationships between cells to recognise tabular components. Experimental evaluation on two different datasets shows a promising results.

Keywords

Multi spatial relationships between table cells, graph rewriting rules, multi possible table interpretations.

1 INTRODUCTION

Layout analysis of tables is a difficult problem in document analysis, mainly due to limitation of table segmentation techniques to deal with irregularities commonly found in tables such as cells spanning multiple columns or rows [ZBC04]. Although, tables which contains no spanning cells through columns or through rows and which the border of their cells are marked by the ruling lines would be easily recognised using simple techniques like projection profile cutting [EG95] or by using the graphic ruling lines. Due to the lack of standard convention of composing tables, this kind of tables structure are exceptionally existed in the literature. The usual distinction of tables, as physical layout, can often encompass the presence of cells that spread over several lines or several columns, and sometimes the borders of neighbouring cells are even misaligned. Even more, the borders of table cells are often not fully marked by the graphic lines.

Representing table structure for various domains of tables needs a framework which is flexible enough to express table layout structure usually differs from table domain to other. The information of both physical and semantic layouts must be expressed. While the former layout can be used for table re-composition, the latter layout contributes in extracting the table's content for re-use purposes.

In [RC03] the most two well-known table representation systems (which are introduced by the World Wide Web Consortium (W3C) and Advancement of Structured Information Standards (OASIS)) [OA99] which are used to represent tables are analysed. It is found that these two system have common Insufficient components which are: First, the representation of irregular physical layouts are difficult. The poorly aligned borders of cells are not allowed and improvised solutions are provided for the spanning cells. Finally, limited means are supplied for the description of the logical structure of a table.

To overcome the unsolved issues mentioned above, a new table representation technique is proposed which exploits the multi spatial relationships between table's cells that were found on wide range of tables in which were observed. I am working on a table domain that contains misaligned cells which, in turn, may have more than one possible interpretation of table structure [MA13]. As a consequence, a graph representation model as well as a new set of graph rewriting rules are proposed to deal with the requirements needed for this table interpretation process.

2 TYPES OF SPATIAL RELATIONSHIPS BETWEEN CELLS

In order to precisely define these multi spatial relationships, one first has to agree on how to express some concepts regarding cell's borders, vertical and horizontal overlaps between table cells.

Definition 1 (Cell's borders)**.** Let c be a cell, then the limits of its *bounding box* are defined by $l(c), r(c), t(c), b(c)$ representing left, right, top and bottom limit respectively. We also have $l < r$ and $t < b$.

Definition 2 (Vertical and horizontal overlap between cells)**.** Let c_1,c_2 be two cells. We say c_1 *overlaps vertically* with c_2 if we have $[t(c_1), b(c_1)] \cap [t(c_2), b(c_2)] \neq \emptyset$, where $[t(c), b(c)]$ is the interval defined by the top and bottom limit of the cell c. Similarly we define *horizontal overlap* of two cells c_1,c_2 by $[l(c_1), r(c_1)] \cap [l(c_2), r(c_2)] \neq \emptyset$.

We now formally define the multiple spatial relationships that can be found between cells. These relationships are ordered based on our observations from most significant to least significant. I found that there are eleven relationships between any two cells in a table which are:

1. Relationships between adjacent cells are observed between every cell c_1 and its neighbouring cell c_2 such that projecting the limits of one cell on the other must not cross any other cells of C which denotes all cells in a table.

Definition 3 (Adjacent cells)**.** For any cells $c_1, c_2 \in C$ with $c_1 \neq c_2$, we say c_1 is adjacent to c_2 if for all $c_3 \in C \setminus \{c_1, c_2\}$ it holds that:

(i) $[min(l(c_1), l(c_2)), max(r(c_1), r(c_2))] \cap [l(c_3), r(c_3)] = \emptyset$ in the case c_1 *overlaps horizontally* with c_2.

(ii) $[min(t(c_1), t(c_2)), max(b(c_1), b(c_2))] \cap [t(c_3), b(c_3)] = \emptyset$ in the case c_1 *overlaps vertically* with c_2.

2. Two cells that are vertically overlapped where the start and end y-axis borders of one cell is within the start and end y-axis borders of other cell.

Definition 4 (Interior vertical Overlap (IVO))**.** Let c_1,c_2 be two cells. We say c_1 *overlaps internally and vertically* with c_2 or c_2 *overlaps internally and vertically* with c_1 if we have $(t(c_1) > t(c_2)$ and $b(c_1) < b(c_2))$ OR $(t(c_2) > t(c_1)$ and $b(c_2) < b(c_1))$ respectively.

a. Two cells that are vertically overlapped and have the same start and end y-axis borders values.

Definition 5 (Fully match vertical Overlap (FVO))**.** Let c_1,c_2 be two cells. We say c_1 *overlaps fully and vertically* with c_2 if we have $t(c_1) = t(c_2)$ and $b(c_1) = b(c_2)$

b. Two cells that are vertically overlapped where the interval of start and end y-axis borders of one cell is in the middle of the start and end y-axis borders of other cell.

Definition 6 (Central vertical Overlap (CVO))**.** Let c_1,c_2 be two cells. We say c_1 *overlaps centrally and vertically* with c_2 or c_2 *overlaps centrally and vertically* with c_1 if we have $(t(c_1) - t(c_2) = b(c_2) - b(c_1))$ OR $(t(c_2) - t(c_1) = b(c_1) - b(c_2))$ respectively.

3. Two cells that are vertically overlapped where the end y-axis border of one cell is greater than the start y-axis border and less than the end y-axis border of other cell.

Definition 7 (Partial vertical Overlap (PVO))**.** Let c_1,c_2 be two cells. We say c_1 *overlaps partially and vertically* with c_2 if we have $(b(c_1) > t(c_2)$ and $b(c_1) < b(c_2))$ and $t(c_1) < t(c_2)$ OR $(b(c_2) > t(c_1)$ and $b(c_2) < b(c_1)$ and $t(c_2) < t(c_1))$

4. Two cells that are vertically overlapped and have the same start or end y-axis border values but not both.

Definition 8 (Sided vertical Overlap (SVO))**.** Let c_1,c_2 be two cells. We say c_1 *overlaps one-sidedly and vertically* with c_2 if we have $t(c_1) = t(c_2)$ and $b(c_1) \neq b(c_2)$ OR $t(c_1) \neq t(c_2)$ and $b(c_1) = b(c_2)$

5. Two cells that are horizontally overlapped where the start and end x-axis borders of one cell is within the start and end x-axis borders of other cell.

Definition 9 (Interior horizontal Overlap (IHO))**.** Let c_1,c_2 be two cells. We say c_1 *overlaps internally and horizontally* with c_2 or c_2 *overlaps internally and horizontally* with c_1 if we have $(l(c_1) > l(c_2)$ and $r(c_1) < r(c_2))$ OR $(l(c_2) > l(c_1)$ and $r(c_2) < r(c_1))$ respectively.

a. Two cells that are horizontally overlapped and have the same start and end x-axis borders values.

Definition 10 (Fully match horizontal Overlap (FHO))**.** Let c_1,c_2 be two cells. We say c_1 *overlaps fully and horizontally* with c_2 if we have $l(c_1) = l(c_2)$ and $r(c_1) = r(c_2)$

b. Two cells that are horizontally overlapped where the interval of start and end x-axis borders of one cell is in the middle of the start and end x-axis borders of other cell.

Definition 11 (Central horizontal Overlap (CHO))**.** Let c_1,c_2 be two cells. We say c_1 *overlaps centrally and horizontally* with c_2 or c_2 *overlaps centrally and horizontally* with c_1 if we have $(l(c_1) - l(c_2) = r(c_2) - r(c_1))$ OR $(l(c_2) - l(c_1) = r(c_1) - r(c_2))$ respectively.

6. Two cells that are horizontally overlapped where the end x-axis border of one cell is greater than the start x-axis border and less than the end x-axis border of other cell.

Definition 12 (Partial horizontal Overlap (PHO))**.** Let c_1,c_2 be two cells. We say c_1 *overlaps partially and horizontally* with c_2 if we have $(r(c_1) > l(c_2)$ and $r(c_1) < r(c_2)$ and $l(c_1) < l(c_2))$ OR $(r(c_2) > l(c_1)$ and $r(c_2) < r(c_1)$ and $l(c_2) < l(c_1))$

7. Two cells that are horizontally overlapped and have the same start or end x-axis border values but not both.

Definition 13 (Sided horizontal Overlap (SHO))**.** Let c_1,c_2 be two cells. We say c_1 *one-sidedly and horizontally overlaps* with c_2 if we have $l(c_1) = l(c_2)$ and $r(c_1) \neq r(c_2)$ OR $l(c_1) \neq l(c_2)$ and $r(c_1) = r(c_2)$

2.1 Experiments

The following tables 1 and 2 show statistical numbers that correspond the total occurrence of every relationship between cells mentioned above. A dataset of 110 tables that are taken from [AD07] is used for testing and obtaining these numbers. One can infer from these tables that, although some types of relationship between cells appear a small number of times, all of these relationships do occur between table cells and therefore we have to consider them when we attempt to correctly recompose table cells. Next, putting into account these spatial relationships illustrated in section 2, I introduce an approach that uses a graph model to represent table structure. Then, I produce graph rewriting rules that contribute to automatically interpreting the possible table structure.

No of Tables	Total of Cells	FHO	CHO	IHO	PHO	SHO
110	3107	6069	18	2275	3976	10586

Table 1: Statistical numbers of horizontal overlap relationships

2.1.1 Arrow representation

We define some arrow drawings in figure 1 that represent the different relationships between nodes to use them later in expressing the production rules. Each of these arrow drawings illustrates one of the relationships formally defined in section 2.

No of Tables	Total of Cells	FVO	CVO	IVO	PVO	SVO
110	3107	47	1589	3470	63	69

Table 2: Statistical numbers of vertical overlap relationships

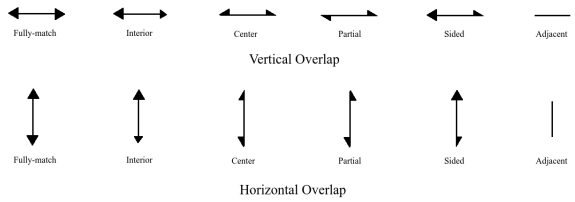

Figure 1: Different arrows represent different relationships between cells

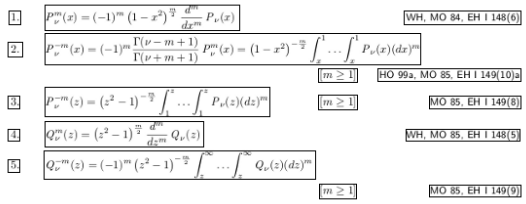

Figure 2: Example for table representation model

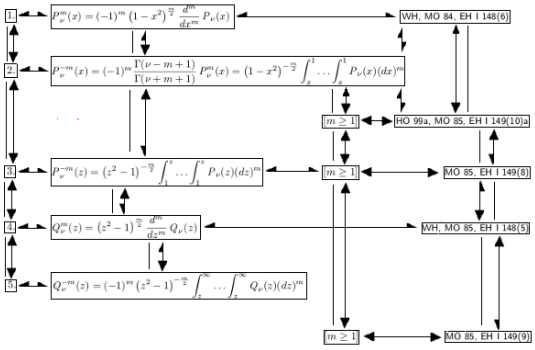

Figure 3: The graph representation of the table in figure 2

3 GRAPH REPRESENTATION MODEL

We use a graph model *G* to represent table layout structure where the geometric relationships that occur between table cells are represented by the edges and the nodes represent the cells themselves. For example, figure 3 is the graph representation of the table in figure 2. Due to the narrow spaces between nodes, one can notice that not all relationships found between nodes (cells) are represented in the figure 3. This representation model opens the gate for expressing table layouts using grammars by representing all layout relationships that any two cells can have.

3.1 Graph rewriting rules

After determining the possible relationships between table cells and also graphically representing table struc-

ture, several rewriting rules are composed to assist in interpreting the layout structure of tables. The purpose of constructing these rules is to rewrite the graph that represents a table so that we have a possible interpretation of the table layout structure. Before representing these rules, I first state what the graph rewriting approach consists of. In [ARC96] graph rewriting rules are encompassed of three components which are:

Production Rules: These rules have a form of $g_l \rightarrow g_r$ where g_l denotes the subgraph that might replace g_r which denotes the subgraph of an initial graph of table G. Later, several production rules are illustrated which are helped in coming up with a set of possible table structure form interpretations.

Embedded Notations: The role of this component is to monitor and save the integrity of the graph G by showing the required conversions on the edges within G when a production rule $g_l \rightarrow g_r$ is applied. A four tuple (n_1, e_1, n_2, e_2) is used to expressed this notation where n_1,e_1 represent a node and an edge from g_r respectively which can be replaced by n_2,e_2 which represent a node and an edge from g_l respectively.

Application Conditions: These conditions are associated with each production rule. They determine when a production rule might be applied. They are typically expressed as constraints or predicates on the node and edge attributes. The conditions on a rule must be satisfied before the rule is applied for rewriting a graph that represents table.

3.1.1 A set of production rules

Based on the fact that table must have at least two rows and two columns, these rules are built to occasionally interpret the table layout structure (in case of having the smallest table) and more often to direct the recognition processing to a possible table interpretation. Each rule here is consist of several possible cell combinations g_l that can replace number of cells in $g_r \in G$. The application of these rules are controlled using the application conditions. This would prevent the possibility of a collision of two or more table form interpretations. Due to the paper page limits, I illustrate only a part of these rules in more details. Figures under this section show the production rules described by the graph illustrated in section 3 where ($\rightarrow$, $|$ and $\emptyset$ denote derivation, selection and null element, respectively).

3.1.2 Production Rule One:

In this case, a node combination of spanning node that has horizontal overlap with more than one node is located in the graph G. There are eight possible interpretations g_l that can replace this combination of cells in $g_r \in G$. The following figure 4 illustrates this rule in detail.

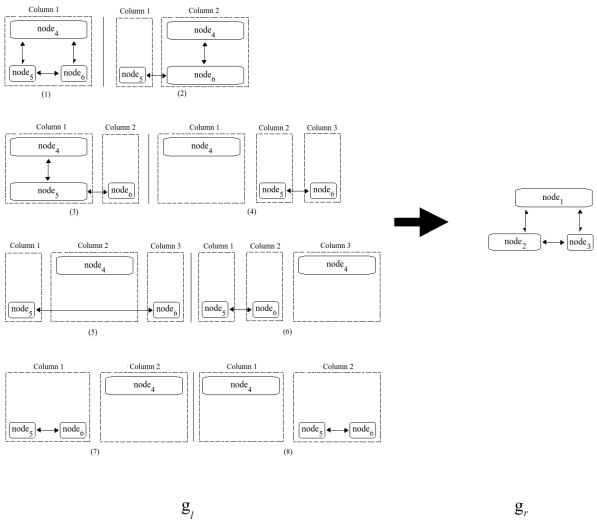

Figure 4: Rule One

Definition 14 (Spanning Node)**.** Let $N = \{n_1, ...n_m\}$ be a set of nodes vertically overlapping each other where $m >= 2$. We say n_h is a horizontal spanning node, if n_h horizontally overlaps with N.

Definition 15 (Production Rule One)**.** Let $g_r \in G$ be a combination that contains a horizontal spanning node n_h horizontally overlaps with N nodes. We call the following nodes re-arranging of this combination in g_l as possible interpretations:

(i) The same combination g_r is remained but clustered in one column col.

(ii) The combination splits into two columns col where the first col contains a set of nodes $N' \in N$ and the second col encompasses n_h horizontally overlaps with a set of nodes $N'' = N \setminus N'$

(iii) Similar to (ii), the combination splits into two columns col. However, the first col encompasses n_h horizontally overlaps with a set of nodes $N' \in N$ and the second col contains a set of nodes $N'' = N \setminus N'$

(iv) In this possible interpretation, three columns col are constructed where the first, second and third columns contain n_h, N' and N'' respectively.

(v) Similar to (iv), three columns col are constructed where the first, second and third columns contain N', n_h and N'' respectively.

(vi) Likewise (iv) and (v), three columns col are constructed where the first, second and third columns contain N', N'' and n_h respectively.

(vii) The combination splits into two columns col. The first col encompasses N and the second col contains n_h

(viii) In this possible interpretation, two columns *col* are constructed where the first and second columns contain n_h and N respectively.

Associated embedded notation: As can be seen in definition 15, each different cell combination in g_r can be replaced by more that one possible interpretation in g_l. This involves some of edge conversions. Definition 16 formally expresses the edge conversions that are needed for each replacing of the combination in g_r with each possible interpretation g_l in definition 15.

Definition 16 (Notation One)**.** Let $g_r \in G$ be the combination mentioned in Def. 15. We call the following notations as the corresponding edge conversions needed to replace g_r with the possible interpretations g_l that also defined in Def. 15 respectively.

1. $(\text{node}_2, \downarrow, \text{node}_5, \uparrow)$
2. $(\text{node}_2, \downarrow, \text{node}_5, \emptyset), (\text{node}_3, \updownarrow, \text{node}_6, \updownarrow)$
3. $(\text{node}_2, \downarrow, \text{node}_5, \updownarrow), (\text{node}_3, \updownarrow, \text{node}_6, \emptyset)$
4. $(\text{node}_2, \downarrow, \text{node}_5, \emptyset), (\text{node}_3, \updownarrow, \text{node}_6, \emptyset)$
5. $(\text{node}_2, \downarrow, \text{node}_5, \emptyset), (\text{node}_3, \updownarrow, \text{node}_6, \emptyset)$
6. $(\text{node}_2, \downarrow, \text{node}_5, \emptyset), (\text{node}_3, \updownarrow, \text{node}_6, \emptyset)$
7. $(\text{node}_1, \downarrow, \text{node}_4, \emptyset), (\text{node}_1, \updownarrow, \text{node}_4, \emptyset)$
8. $(\text{node}_2, \downarrow, \text{node}_5, \emptyset), (\text{node}_3, \updownarrow, \text{node}_6, \emptyset)$

3.1.3 Production Rule Two:

A node combination of two nodes that have horizontal overlap with two nodes is located in the graph G. In this case, there are thirteen possible interpretations g_l that can replace this combination of cells in $g_r \in G$. The following figure 5 illustrates this rule in detail.

Definition 17 (Rule Two)**.** Let $g_r \in G$ be a combination that contains n_1, n_2 as two nodes vertically overlaps with each others and n_3, n_4 as two nodes vertically overlaps with each others such that n_1 horizontally overlaps with n_3 and n_2 horizontally overlaps with n_4. We call the following nodes re-arranging of this combination in g_l as possible interpretations:

(i) The same combination g_r is remained but clustered in one column *col*.

(ii) The combination splits into two columns *col* where the first *col* contains a node n_1 horizontally overlaps with n_3 and the second *col* encompasses n_2 horizontally overlaps with n_4.

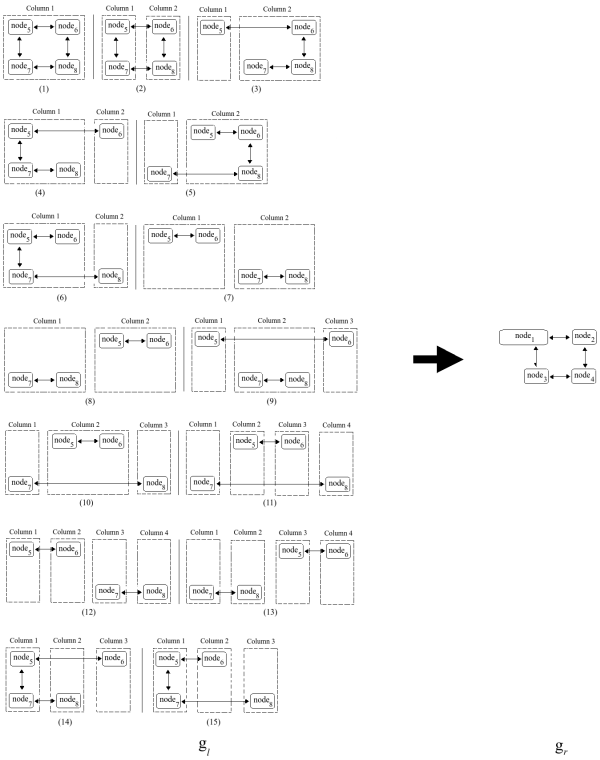

Figure 5: Rule Two

(iii) Similar to (ii), the combination splits into two columns *col*. However, the first *col* encompasses a node n_1 and the second *col* contains a nodes n_2 horizontally overlaps with n_4 which vertically overlaps with n_3.

(iv) Likewise (iii), the combination splits into two columns *col*. However, the first *col* encompasses a node n_1 horizontally overlaps with n_3 which vertically overlaps n_4 and the second *col* contains a node n_2

(v) Again, the combination splits into two columns *col*. However, this time, the first *col* encompasses a node n_3 and the second *col* contains a node n_1 vertically overlaps with n_2 which horizontally overlaps with n_4

(vi) In this possible interpretation, the combination splits into two columns *col*. The first *col* encompasses a node n_1 horizontally overlaps with n_3 and vertically overlaps with n_2 and the second *col* contains a node n_4

(vii) This time, the combination splits into two columns *col*. The first *col* encompasses a node n_1 vertically overlaps with n_2 and the second *col* contains a node n_3 vertically overlaps with n_4

(viii) Similar to (vii), the combination splits into two columns *col*. The first *col* encompasses a node

n_3 vertically overlaps with n_4 and the second *col* contains a node n_1 vertically overlaps with n_2

(ix) Three columns *col* are constructed where the first, second and third columns contain n_1, n_3 vertically overlaps with n_4 and n_2 respectively.

(x) Likewise (ix), three columns *col* are constructed. However, the first, second and third columns contain n_3, n_1 vertically overlaps with n_2 and n_4 respectively.

(xi) In this possible interpretation, four columns *col* are constructed where the first, second, third and four columns contain n_3, n_1, n_2 and n_4 respectively.

(xii) Similar to (xi), four columns *col* are constructed. However, the first, second, third and four columns contain n_1, n_2, n_3 and n_4 respectively.

(xiii) Likewise (xii), four columns *col* are constructed. However, the first, second, third and four columns contain n_3, n_4, n_1 and n_2 respectively.

(xiv) Three columns *col* are constructed where the first, second and third columns contain n_1 horizontally overlaps with n_3, n_4 and n_2 respectively.

(xv) Likewise (xiv), three columns *col* are constructed. However, the first, second and third columns contain n_1 horizontally overlaps with n_3, n_2 and n_4 respectively.

Associated embedded notation: Each different cell combination in g_r can be replaced by more that one possible interpretation in g_l. This involves some of edge conversions. Definition 18 formally expresses the edge conversions that are needed for each replacing of the combination in g_r with each possible interpretation g_l in definition 17.

Definition 18 (Notation Two)**.** Let $g_r \in G$ be the combination mentioned in Def. 17. We call the following notations as the corresponding edge conversions needed to replace g_r with one of the possible interpretations g_l that also defined in Def. 17 respectively.

1. $(\text{node}_1, \updownarrow, \text{node}_5, \updownarrow)$
2. $(\text{node}_1, \updownarrow, \text{node}_5, \updownarrow)$
3. $(\text{node}_1, \updownarrow, \text{node}_5, \emptyset)$
4. $(\text{node}_1, \updownarrow, \text{node}_5, \updownarrow), (\text{node}_2, \updownarrow, \text{node}_6, \emptyset)$
5. $(\text{node}_3, \updownarrow, \text{node}_7, \emptyset)$
6. $(\text{node}_1, \updownarrow, \text{node}_5, \updownarrow), (\text{node}_4, \updownarrow, \text{node}_8, \emptyset)$

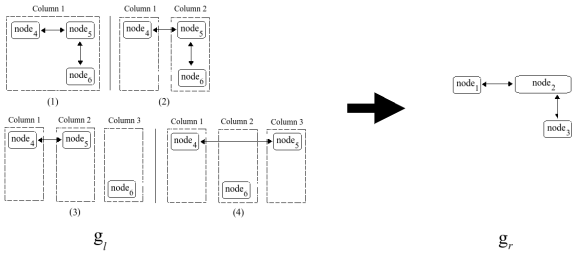

Figure 6: Rule Three

7. $(\text{node}_1, \updownarrow, \text{node}_5, \emptyset), (\text{node}_2, \updownarrow, \text{node}_6, \emptyset)$
8. $(\text{node}_3, \updownarrow, \text{node}_7, \emptyset), (\text{node}_4, \updownarrow, \text{node}_8, \emptyset)$
9. $(\text{node}_1, \updownarrow, \text{node}_5, \emptyset), (\text{node}_4, \updownarrow, \text{node}_8, \emptyset)$
10. $(\text{node}_3, \updownarrow, \text{node}_7, \emptyset), (\text{node}_2, \updownarrow, \text{node}_6, \emptyset)$
11. $(\text{node}_3, \updownarrow, \text{node}_7, \emptyset), (\text{node}_2, \updownarrow, \text{node}_6, \emptyset)$
12. $(\text{node}_1, \updownarrow, \text{node}_5, \emptyset), (\text{node}_2, \updownarrow, \text{node}_6, \emptyset)$
13. $(\text{node}_3, \updownarrow, \text{node}_7, \emptyset), (\text{node}_4, \updownarrow, \text{node}_8, \emptyset)$
14. $(\text{node}_3, \updownarrow, \text{node}_7, \updownarrow), (\text{node}_4, \updownarrow, \text{node}_8, \emptyset)$
15. $(\text{node}_1, \updownarrow, \text{node}_5, \updownarrow), (\text{node}_3, \updownarrow, \text{node}_6, \emptyset)$

3.1.4 Production Rule Three:

A combination of three nodes which are $node_1$,$node_2$,$node_3$ where $node_1$ has vertical overlapping with $node_2$ and in the same time $node_2$ has horizontal overlapping with $node_3$ is located in G. In this case, there are four possible cell interpretations g_l that can replace this combination of cells in $g_r \in G$. The following figure 6 illustrates this rule in detail.

Definition 19 (Rule Three)**.** Let $g_r \in G$ be a combination that contains n_1, n_2 and n_3 as nodes such that n_1,n_2 vertically overlap with each others and n_2 horizontally overlaps with n_3. We call the following nodes re-arranging of this combination in g_l as possible interpretations:

(i) The same combination g_r is remained but clustered in one column *col*.

(ii) The combination splits into two columns *col* where the first *col* contains node n_1 and the second *col* encompasses n_2 horizontally overlaps with n_3.

(iii) In this possible interpretations, the combination splits into three columns *col*. The first, second and third columns encompass n_1, n_2 and n_3 respectively.

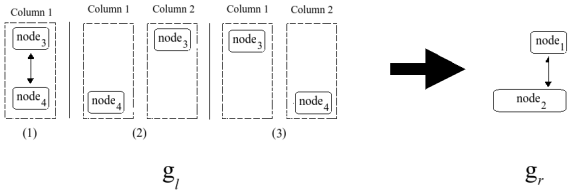

Figure 7: Rule Four

(iv) Likewise (iii), the combination splits into three columns *col*. However, the first, second and third columns *col* encompass n_1, n_3 and n_2 respectively.

Associated embedded notation: Each different cell combination in g_r can be replaced by more that one possible interpretation in g_l. This involves some of edge conversions. Definition 20 formally expresses the edge conversions that are needed for each replacing of the combination in g_r with each possible interpretation g_l in definition 19.

Definition 20 (Notation Three)**.** Let $g_r \in G$ be the combination mentioned in Def. 19. We call the following notations as the corresponding edge conversions needed to replace g_r with one of the possible interpretations g_l that also defined in Def. 19 respectively.

1. $(\text{node}_2, \updownarrow, \text{node}_5, \updownarrow)$ 2. $(\text{node}_2, \updownarrow, \text{node}_5, \updownarrow)$
3. $(\text{node}_2, \updownarrow, \text{node}_5, \emptyset)$ 4. $(\text{node}_3, \updownarrow, \text{node}_6, \emptyset)$

3.1.5 Production Rule Four:

A combination of two nodes which are $node_1$,$node_2$ where $node_1$ has horizontal overlapping with $node_2$ is located in G. In this case, there are three possible cell interpretations g_l that can replace this combination of cells in $g_r \in G$. The following figure 7 illustrates this rule in detail.

Definition 21 (Rule Four)**.** Let $g_r \in G$ be a combination that contains n_1, n_2 as nodes such that n_1 horizontally overlaps with n_2. We call the following nodes rearranging of this combination in g_l as possible interpretations:

(i) The same combination g_r is remained but clustered in one column *col*.

(ii) The combination splits into two columns *col* where the first *col* contains node n_2 and the second *col* encompasses n_1

(iii) In this possible interpretations, the combination splits into two columns *col*. The first and second columns *col* encompass n_1 and n_2 respectively.

Associated embedded notation: Each different cell combination in g_r can be replaced by more that one possible interpretation in g_l. This involves some of edge conversions. Definition 22 formally expresses the edge conversions that are needed for each replacing of the combination in g_r with each possible interpretation g_l in definition 21.

Definition 22 (Notation Four)**.** Let $g_r \in G$ be the combination mentioned in Def. 21. We call the following notations as the corresponding edge conversions needed to replace g_r with one of the possible interpretations g_l that also defined in Def. 21 respectively.

1. $(\text{node}_1, \updownarrow, \text{node}_3, \updownarrow)$ 2. $(\text{node}_2, \updownarrow, \text{node}_4, \emptyset)$
3. $(\text{node}_1, \updownarrow, \text{node}_3, \emptyset)$

4 DISCUSSION AND EXPERIMENTAL EVALUATION

4.1 Table structure analysis

Using the graph rewriting rules, an analysing of the table structure is performed. Since the information of the table structure is fully described in the graph that can be re-written by applying the rewriting rules, one can utilise a general graph parser for table structure analysis. As these rewriting rules have a form which is equivalent to a context sensitive grammar, it is not easy to parse the tables.

To overcome this problem, a constraint is associated and performed for each rule, prior to applying it, to help with recognising table structure. As observed and also mentioned in [MA13], the tables in our dataset have more than one possible interpretation of their structures. Therefore, one has to use a suitable constraint on the rewriting rules each time one attempts to produce a particular desirable output. Next, an example of constraints that are imposed on the rewriting rules is shown to clarify how one can select one possible interpretation g_l of cells combination $g_r \in G$ or more from the proposed rewriting rules to use them in obtaining a specific possible table interpretation. Parsing a table, that is taken from our table dataset, using the rewriting rule presented in section 3.1.1 which passes specific constraints, is illustrated in the next example.

4.2 An example of constraints associate with each rule

To have the possible interpretation of table which is shown in figure 11, I use two constraints where the first one states that *For any* $node_1$*,* $node_2$*,* $noda_3$*, if* $node_1$ *and* $node_2$ *vertically overlap within a line, and* $node_2$ *and* $node_3$ *horizontally overlap, then the three*

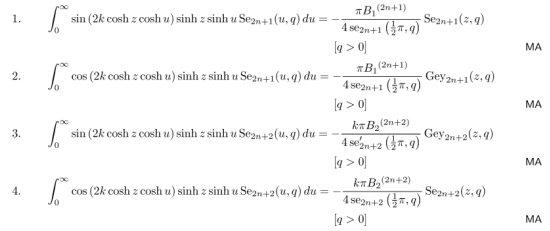

Figure 8: table which is taken from [AD07]: example

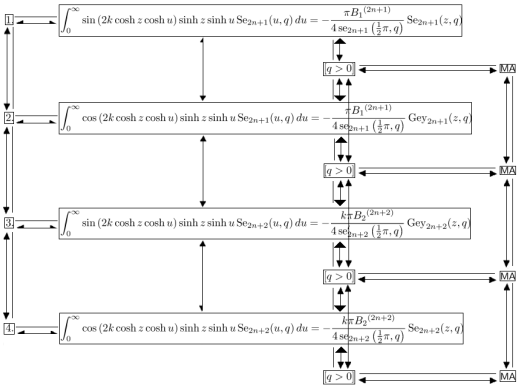

Figure 9: A graph represents table which is taken from [AD07]

nodes must be placed in separate columns. Rule three presents this combination of nodes g_r and its possible interpretation g_l labelled (3) is applied on this combination to rewriting a sub of graph G, if $(b(node_2) - t(node_2))$>$(b(node_3) - t(node_3)) * e$ where e is a fixed value. In my experiment, $e = 2$ (which is determined empirically). Otherwise, possible interpretation g_l labelled (2) in rule three is applied. The second states *If $node_1$ and $node_2$ are in different lines and are horizontally overlapping, then they must be placed in the same column*. Rule four presents this combination of nodes g_r and its possible interpretation g_l labelled (3) is applied on this combination to rewriting a sub of graph G, if $(b(node_1) - t(node_1))$>$(b(node_2) - t(node_2)) * e$ where e is a fixed value. In my experiment, $e = 2$ (which is determined empirically). Otherwise, possible interpretation g_l labelled (1) in rule four is applied.

In the next steps, a description of how to apply the possible interpretations, that are selected above, to the graph in figure 9, that represents the table in figure 8, is given, to obtain a possible interpretation of this table structure. Figure 11 shows the output of this process. To visually show this output, I border every column's cells in this table with dash-graphic lines.

4.2.1 *Steps toward interpretation of table structure: example*

For our example, we order the application of these possible interpretations as follows:

1. possible interpretation labelled (3) in Rule Three (thereafter called PI_3).

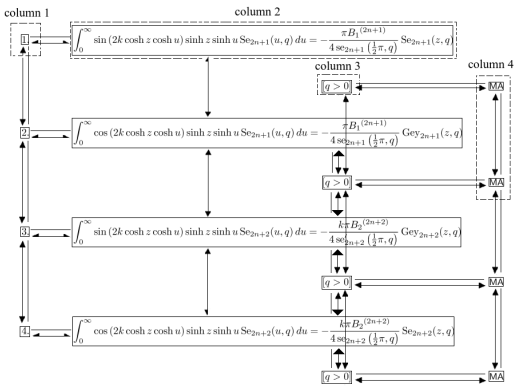

Figure 10: First rewriting of the graph that represents a table which is taken from [AD07]

2. possible interpretation labelled (1) in Rule Four (thereafter called PI_4).

The rewriting procedure begins by searching in the graph in figure 9, starting from the top-left node, for a combination of nodes that can be replaced by PI_3. Once a candidate combination of nodes is found, the replacement process is accomplished. In our case, the first two nodes in the first row on the graph as well as the first node in the second row are marked as a candidate combination that can be replaced by PI_3. Figure 10 shows the first rewriting of the graph in figure 9.

As it can be seen in the figure 10, the combination of three nodes were split to three different columns by applying the PI_3. As a consequence, some edges are removed. The same process is repeatedly preformed on the rest of graph nodes whenever the same combination of nodes as the one on g_r in figure 6 is found.

Similar to the way of applying PI_3, the possible interpretation PI_4 is implemented. Figure 10 shows the result of performing this PI_4 for the first time on two nodes horizontally overlapped. The two nodes remain horizontally overlapped and clustered in one column. The same process is repeatedly preformed on the rest of graph nodes whenever the same combination of nodes as the one on g_r in figure 7 is found.

Figure 11 illustrates the final result of rewriting the graph in figure 9. It is clear that applying the selected possible interpretations PI_3 and PI_4 has successfully contributed to obtain one of the possible interpretations of table structure that is shown in figure 8.

In addition to the example above, which shows a case of using my framework, and for more robust evaluation, I have run the implementation of this framework over 110 tables that are taken from [AD07]. This is accomplished by using the possible interpretations PI_3, PI_4 with another possible interpretation PI_2 from rule two which states:

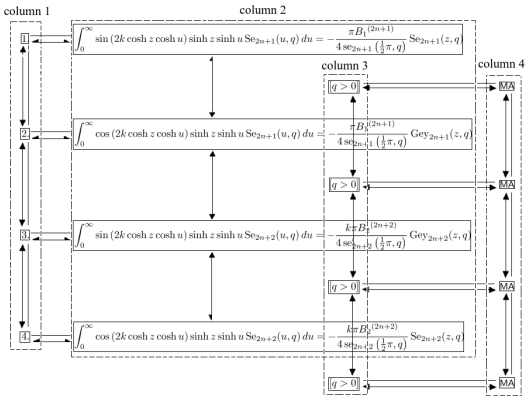

Figure 11: Final result of rewriting the graph that represents a table which is taken from [AD07]

1. *Each ($node_1$,$node_2$) and ($node_3$,$mode_4$) which are in same lines l_1 and l_2 respectively and vertically overlapped as well as $node_1$ is horizontally overlapped with $node_3$ and $node_2$ is horizontally overlapped with $node_4$ must be split into three different columns such that first, second and third columns contain $node_1$ is horizontally overlapped $node_3$,$node_2$ and $node_4$ respectively*. Rule two is represented this combination g_r and its possible interpretation g_l labelled (15) is applied on this combination to rewriting a sub of graph G, if $(b(node_2) - t(node_2))$>$(b(node_4) - t(node_4)) * e$ where e is a fixed value. In my experiment, $e = 2$ (which is determined empirically). Otherwise, possible interpretation g_l labelled (2) in rule two is applied.

Note: the goal of this experiment is to find the misaligned columns and split them from other columns. This would itself form one possible interpretation of table structure.

These constraints that were used to select these possible interpretations are inferred and constructed based on observing common features of the tables structure that we have in the dataset.

Table 3 shows the results of running my technique over 110 tables in concise manner. The table contains in its first column a classification of tables with different number of columns that I used in the experiment. The second one have the corresponding number of tables in the dataset that fall into every type of table in first column. The rest of columns in this table contain number of tables that have all their columns correctly extracted, number of tables that their columns were partially extracted (75% to 95%) and number of tables that the technique was unable to correctly extract their columns.

The table 3 shows very promising results of running the implementation of the proposed framework on 110 tables. However, one should still do more experiments on other datasets to ensure the technique performance consistency. Also, it is essential to test using tables from different domains and with various structures. This involves observing the target tables and coming up with general constraints to contribute on selecting suitable possible interpretations from the production rules which in turn are used to interpret the tables structure.

4.3 More experiments

For testing the robustness of my technique, the implementation of my method was run over a dataset which was used for a competition at ICDAR 2013 conference. The dataset is available online which is freely downloadable at http://www.tamirhassan.com/competition.html, contains 40 excerpts as individual PDF files, with a total of 157 tables.

4.3.1 No need of constraints

The following points state the selected possible interpretations g_l of combination of nodes g_r, appear in the rewriting rules, which are obtained by observing the tables structure of this dataset. Since, there is no misaligned cells within this dataset tables, no constraints are imposed on applying these possible interpretations.

1. *Each $node_1$ that is in l_1 and horizontally overlapped with $node_2$ and $node_3$ which are in same line l_2 and vertically overlapped must be clustered into one column*. Rule one represents this combination g_r and its possible interpretation g_l labelled (1) is applied on this combination to rewriting a sub of graph G.

2. *Each ($node_1$,$node_2$) and ($node_3$,$mode_4$) which are in same lines l_1 and l_2 respectively and vertically overlapped as well as $node_1$ is horizontally overlapped with $node_3$ and $node_2$ is horizontally overlapped with $node_4$ must be split into two different columns such that first and second columns contain $node_1$ is horizontally overlapped $node_3$ and $node_2$ is horizontally $node_4$ respectively*. Rule two represents this combination g_r and its possible interpretation g_l labelled (2) is applied on this combination to rewriting a sub of graph G.

3. *Each $node_1$ and $node_2$ which are in different lines l_1 and l_2 respectively as well as horizontally overlapped are remained and form a column*. Rule four is represented this combination g_r and its possible interpretation g_l labelled (1) is applied on this combination to rewriting a sub of graph G.

4.3.2 Experimental results

After running the described technique using these possible interpretations, over the target dataset, the following results that are concisely expressed in table 4 is obtained.

Table with different number of columns	No. of tables	All columns are correctly extracted	Columns are partially extracted	Failure to extract columns
Table with 3 columns	25	21	3	1
Table with 4 columns	65	62	1	2
Table with 5 columns	20	16	2	2

Table 3: Result in numbers of running my technique over 110 tables

No. of tables	All columns are correctly extracted	Columns are partially extracted	Failure to extract columns
157	111	14	32

Table 4: The results of running my technique over tables of the target dataset

Having observed the tables that their columns either partially extracted or not correctly extracted, I found that the most common error in these tables occurs when the spanning cell does not horizontally overlap all cells that it should do, due to the fact that the cells were extracted based on their contents. Manual intervention, as it is described in [MA13], would be one of the solutions to this problem. Another solution is to extend the cell segmentation technique in [MA12] so that, it extracts the real borders of cells.

A comparison of the performance of my technique on this dataset with other techniques performance on the same dataset is not possible due to the absence of any available published results. In addition to this, current table recognition methods are informally presented [ZR05]. Details of how these techniques work is usually not fully described. This makes it difficult if not impossible to compare different techniques performance.

5 CONCLUSION

The framework represented in this paper was built on the observation of a wide range of tabular forms which occur in many documents from different domains. The abstract components of this framework can be used as basis of wide range of other applications of document recognition. The technique is also able to produce several interpretations of a table. Unlike other table representation techniques, the proposed approach has the capability to deal with misaligned columns that sometimes appear in tabular mathematical components. To achieve this, I first give a formal definitions to all possible relationships that can be found between table cells. Then, a graph model is described for representing table layout structure. A set of rewriting rules are given to contribute to rewrite the graph. Two examples of the rewriting rules and how some of possible interpretations g_l in these rules are selected, using specific constraints, are also described. An application of the selected possible interpretations on a table, which is taken from our dataset, is demonstrated. Finally, experiments on two different-domains datasets shows promising results.

6 REFERENCES

[AD07] A. Jeffrey and D. Zwillinger. Table of Integrals, Series, and Products. Elsevier Inc. 2007.

[EG95] Edward A. Green and Mukkai S. Krishnamoorthy. Model-based analysis of printed tables. ICDAR. pp.214-217, 1995.

[OA99] OASIS, xml exchange table model document type definition. Organization for the advancement of structured information standards. 1999.

[MA12] Mohamed Ali Ibrahim Alkalai and Volker Sorge. Issues in Mathematical Table Recognition. CICM '12, MIR Workshop. 2012.

[MA13] Mohamed Alkalai. Recognising Tabular Mathematical Expressions using Graph Rewriting. CIAPR. Volume 8259, Springer-Verlag, pp.124-133, Havana, Cuba, November 2013.

[RC03] J. Ramel, M. Crucianu, N. Vincent and C. Faure. Detection, Extraction and Representation of Tables. IEEE Computer Society. pp.374-378, Washington, DC, USA, 2003.

[ARC96] Armon Rahgozar and Robert Cooperman. A Graph-based Table Recognition System. SPIE Proc. pp.192-203, 1996.

[ZBC04] Zanibbi, Richard and Blostein, Dorothea and Cordy, R. A survey of table recognition: Models, observations, transformations, and inferences. Int. J. Doc. Anal. Recognit. Volume 7, Springer-Verlag, pp.1-16, March 2004.

[ZR05] Zanibbi, Richard, **advisor:** Blostein, Dorothea and Cordy, James R. A Language for Specifying and Comparing Table Recognition Strategies, School of Computing, Queen's University, 2005, Kingston, Ont., Canada, Canada.