A Constraint-based Approach to Table Structure Derivation

Matthew Hurst, Intelliseek, Inc. Applied Research Center
mhurst@intelliseek.com

Abstract

This paper presents an approach to deriving an abstract geometric model of a table from a physical representation. The technique developed uses a graph of constraints between cells which must be satisfied in order to determine their relative horizontal and vertical position. The method is evaluated with a test set of tables drawn from US Securities and Exchange Commission (SEC) filings.

1 Problem Description

Given a flat, textual representation of a table, we wish to derive a abstract geometric model that identifies the relative location of cells and captures their textual content. For example, consider the ASCII table shown in Fig. 1, we want to derive the abstract geometric model represented by XML of the form <CELL X0="1" Y0="0" X1="3" Y1="0">YEAR ENDED DECEMBER 31,</CELL> This XML description may then be used to deliver the appropriate HTML version of the table, or as input to further high-level applications such as an information extraction systems.

The process we wish to implement takes as input an ASCII table and produces as output the abstract geometric model of the table. The co-ordinates of this model are composed of relative values from 0 to X_{max} and from 0 to Y_{max}. The co-ordinate system is a grid of spaces labeled by the co-ordinate values. Any cell in the table occupies one or more of these spaces to cover a rectilinear area in the abstract spatial table. See [6], [3], [5] for similar models.

2 Presenting Relationships in Tables

The only mechanism available to the table designer to indicate the logical relationship between the content of cells is some form of *spatial commonality*. Cells that are logically related have horizontal or vertical overlap (though, importantly, the converse is not true). Cells may be characterised in terms of their horizontal and vertical **extent** - i.e. the position and the width or height of the cell. There are a number of ways in which the extent of a cell may interact with that of another. We call this interaction **containment** and say that one cell's extent either partially or completely contains that of another.[1]

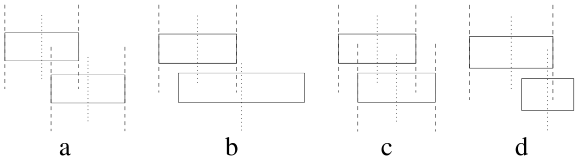
```
                                  YEAR ENDED DECEMBER 31,
                                 ------------------------
                             1991     1992     1993     1994
                           -------  -------  -------  ------
                             (UNAUDITED)
                                (DOLLARS IN THOUSANDS)

STATEMENT OF OPERATIONS
Investment income......   $ 8,806  $ 7,953  $ 8,333  $8,820
Interest expense.......     4,139    3,509    3,661   4,756
```

Figure 1. DOLLARS IN THOUSANDS **is associated with all values;** UNAUDITED **is only associated with the first two years (table fragment).**

Firstly, there may be no interaction. Secondly, the extent of one cell may partially contain that of another. Thirdly, the extent of one cell may completely contain that of another. When considering the interaction of extents, we label one cell the **source** and the other the sink. This gives a direction to the interaction and we make the general assumption that tables are read top to bottom and left to right.

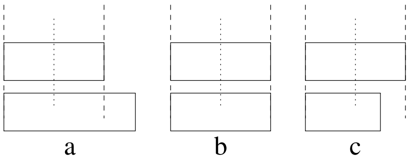
We can enumerate the possible forms of (non-empty) interactions of a pair of cells' extents.

class 1 There is partial containment between the sink and the source. There are variations on the location of the centre lines of the source and sink, as well as the edges of the source or sink.

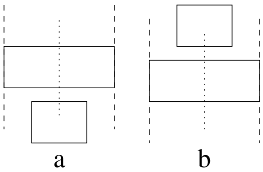


class 2 The source and sink are aligned on one or both sides.

[1] Here we describe only the horizontal extent, though these remarks may also be applied to the vertical extent.



class 3 The extent of the sink or source is wholly contained within the other extent.



下面说的是使用单元格内的文本内容长宽来预估单元格大小，以此定位单元格，这跟我之前项目的做法（首先ocr再后处理）类似；第二种是针对图片中原有的线条直接定位单元格

We now consider how the extent of a cell is expressed in the graphical representation of a table. There are two features that indicate the spatial characteristics of the cells on the page. The first is the position of the textual content of the cell. The characters are positioned approximately within the vertical and horizontal extents of the cell. However, we cannot assume a direct relationship between the extents and the apparent size of the text - text may underfill or overfill the extents of its cell. The second is the use of lines placed on the page to indicate in whole or in part the extent of some set of cells.

When dealing with real-world text files we notice the following problem areas:

Implicit indication of extent : This is due to the lack of explicit line-art and the lack of correlation between the size of the textual content and the extent of the cell. In addition, text may be stretched or otherwise spaced out to approximate extent.[2]

Mis-alignment of cells : As a consequence of editing environments, cells are often not aligned and may in fact be aligned with the wrong cells.

To summarise: the arrangement of cells on the page is a reflection of the logical structure of the table expressed in terms of the containment of the cells' extents. Containment is ambiguous (effectively a reduction of dimensions), and the constraints of the layout task and the manner in which tables are produced may add further noise into the overall presentation.

3 The Constraint-based Approach

The approach proposed here uses constraints between cells to determine their relative position. Intuitively, we make statements such as "cell A is to the left of cell B." The motivation for this approach is the assumption that the table as a whole contains enough information to overcome any local noise at the per cell level.

In order to reduce the search space, we limit these constraints to those which express *local* spatial relationships. This may be carried out by, for each cell, inspecting its immediate neighbourhood and determining those cells with which it has some spatial interaction. The immediate neighbourhood of a cell can not be defined in absolute geometric terms, so it is appropriate to consider the neighbourhood as being the set of cells that are in some way 'visible' to the source cell via a line-of-sight metaphor horizontally or vertically - looking between text blocks.

`ASCII` tables will often contain under-specified spatial interaction due to the lack of explicit extent of cells (for example the content `(DOLLARS IN THOUSANDS)` in Fig. 1 has an extent that only spans a sub-set of the values that its interpretation must be applied to) and over specification due to the mis-alignment of cells. Consequently the discovery of local spatial relationships is not a simple matter of computing the containment of cell extents as proxied by their contents, but requires further and more complex analysis procedures.

We define a representation for local spatial relationships, our constraints - the **proto-link**: a tuple $\langle c_i, c_j \rangle$ where c_i is the source of the link and c_j is the sink of the link. We regard proto-links as hypotheses indicating potential spatial interaction between two cells. Fig. 2 a) shows an example of proto-links discovered below a source cell (the unmarked cell). Notice that the links ordered left to right based on the portion of the sink cell that is visible to the source cell.

Proto-links capture spatial commonality as a mixed (i.e. ambiguous) indication of: parent and child cells in the logical table (e.g. an access and data cell), siblings cells, arrangements of cells that are a consequence of concatenating what are essentially independent tables together or the point where a column or row of cells is interrupted by a cut-in or other complex structure.

3.1 Partially Ordered Unification Variables

We interpret proto-links as representing ordering in the relative co-ordinate system between the source and the sink (as described in Section 1, cells have two pairs of co-ordinates). In the case of vertical proto-links, the proto-link indicates that the source's Y_1 is less than the sink's Y_0 and so on.

These inequalities can be represented by a **system of partially ordered unification variables** (POU variables): a tuple $\langle V, O \rangle$, where V is a set of variables and O is a function mapping the set of all pairs of variables to true or false (true indicates that the relation $\leq$ holds).

These variables are unifiable so that the methods that operate on them may state that two cells share a particular spa-

[2] Sometimes it is only the understanding of the domain that allows us to determine the extent of a cell.

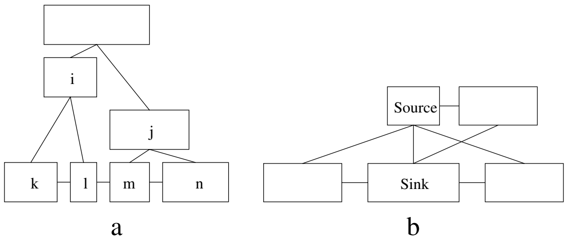


Figure 2. a) i and j are connected as the right frontier of i (l) is connected to the left frontier of j (m). So they are left to right ordered wrt the source cell. b) An illegal proto-link configuration.

tial characteristic (e.g. are aligned along their relative X_0 values) as the result of some constraint resolution.

Such a system may conveniently be implemented by a graph in which the nodes are the variables and the arcs between them represent the ordering. We require that non-maximal paths in the graph are eliminated. If $A \leq B \leq C$ then there is no arc between C and A as this relationship may be inferred.

As these variables are unifiable, a procedure is required to carry out the unification operation. This procedure does some simple house keeping updating the graph as appropriate. A consequence of the unifiability of the variables is that the arrangement $A \leq B, B \leq A$ results in the unification of A and B by implication.

Intuitively, POU variables represent (possible) column and row co-oridnates. Cells which share variables (explicitly or through unification) share column or row variables and are thus aligned.

3.2 Proto-link Configuration

We define the configuration of a proto-link as being the number of proto-links exiting the source on the side from which this proto-link originates, the number of proto-links entering the sink on side which this proto-link enters, whether the source is connected and whether the sink is connected. For the link j-m in Fig. 2 a, the configuration is `2, 1, connected, connected`.

When there are multiple sinks or sources, we can inspect the set of cells to see if they are connected. The cells are trivially connected if they are linked left to right or top to bottom via proto-links. Otherwise, if two cells that are ordered via the order of their proto-links connecting them with the source (or sink) then we may look to see if the order is implied below or to the right by other links. This may be done by inspecting the frontiers of the cells according to the tree of proto-links extending from it. A frontier is the set of cells on the left, right, up, or down edge of the tree (Fig. 2 a). Connectivity provides information about the ordering of sibling sink cells (cells linked to the same source).

As the proto-links are used to distribute constraints between the variables representing the abstract spatial position of the cells, they may also be used to identify cases which should not occur in a table. Such conflicts may then be used to reassess the segmentation of the text into cells as well as correct the assignment of proto-links.

Some arrangements of links are not permitted, they represent constraint conflicts (Fig. 2 b)). To resolve these illegal instances there are two possible operations. The first is to merge some subset of the cells in the table so that the offending proto-link configuration is removed. The other is to remove one or more of the proto-links in the configuration surrounding and including the offending proto-link.[3]

Merging cells means that we accept that the original analysis of the textual input segmenting it into a set of text areas forming the content of cells was incorrect. Removing one or more of the proto-links means that we accept that the assignment of proto-links was incorrect.[4]

In summary, configurations are used to provide extra constraints (through the detection of connectivity) and to detect conflicts.

4 Implementation and Evaluation

In this section, we describe a system which takes as input SEC filings and delivers an XML representation of the tables in the document.

4.1 Implementation

The table recognition process described below takes as input flat `ASCII` files and delivers XML or HTML output. The main steps can be summarised as follows (assuming the discovery of tables and the initial recognition of cells as described in [4]):

Discover Proto-links The same operation is carried out on the bottom and right sides of the bounding box of each cell. We will consider the bottom side here.

A mask is used to implement the line of site search. The mask represents those x co-ordinate positions that have been filled by a sink cell. If a link is to be made to a sink, its horizontal extent must be visible ‘through’ the mask.

The search for sink cells is carried out starting at the first line in the document below the source’s bounding box. All the cells whose bounding box’s y_0 is on that line and whose

[3] Here, we don’t consider the more complex solution of splitting up text blocks.

[4] Note that a merge means that we think the text in the two cells should be aggregated to form a single coherent piece of text.

horizontal extent intersects with the source are retrieved. Each sink is linked to the source and the mask is set for all the x positions within the extent of the sink.

Then the next line is considered. If a cell is found, a check is made with the mask to see if there is a gap which permits access to the cell. If there is then a link is made and the mask is updated. If not then no link is made. The search continues until the mask is full or the edge of the table is encountered.

The full algorithm for discovery uses a number of further heuristic steps to propose additional proto-links based on spanning cells and so on. These steps are not described here for brevity.

Assign POU Variables to Cells For each cell four variables are required (X_0, Y_0, X_1, Y_1). We can take a slightly pragmatic approach simplify processing in later stages by recognizing when certain cells share a variable. Cells with a y_0 value equal to the y_0 value for the bounding box of the table can be assumed to have the same Y_0 value (which will eventually be set to 0). An equivalent process is carried out for the X values.

Reduce POU Variables We can also recognize cases where the width of the cell in terms of the relative co-ordinate system is unary (actually represented as zero width due to the nature of the implementation of the indexing system).

We can define this context in the following manner.

1. A cell that has no proto-links linking it to cells below we term **unary** below.
2. A cell that has a single proto-link linking it to a sink below is unary below iff the sink cell is unary below.
3. A cell that has no proto-links linking it to cells above we term **unary** above.
4. A cell that is linked to a spanning cell above is unary above.
5. A cell that is linked to a single cell above that is itself unary above is unary above.

A similar set of definition can be made for horizontal proto-links. Any cell that is unary below and unary above can unify its X_0 and X_1 POU variables. Any cell that is unary right and unary left can unify its Y_0 and Y_1 POU variables. [5]

This process effectively locates columns of unit width.

Dealing with illegal proto-link configurations When we encounter illegal proto-link configurations we can elect to either remove one or more links or to merge two or more text blocks that we may prefer to consider as one cell.[6]

Apply Constraints to POU Variables Now that a set of proto-links describe the spatial relationships between a set of cells we can consider constraining the POU variables used to describe the relative location of the cells based on our understanding of what the proto-links denote. This is done simply by following the constraint sets for each proto-link configuration (a set of heuristics). Additional trivial constraints state that $X_0 \leq X_1$ and that $Y_0 \leq Y_1$ for each cell.

Disambiguate POU Variables Now that we have a complete system of POU variables associated with the cells in the table we must check for partial order that require further clarification. Strategies for disambiguation are based largely on the inspection of features of the physical space of the table.

There are two cases that need to be handled.

1. Multiple superior variables. The POU variable graph states that $v_i \leq \{S\}$ where $\{S\}$ is a set of 2 or more POU variables.
2. Multiple inferior variables. The POU variable graph states that $v_i \geq \{I\}$ where $\{I\}$ is a set of 2 or more POU variables.

There are two types of resolution to these conflicts: variables may be ordered or variables may be unified.

Propogate Relative Co-ordinate Values Once the POU variables have been disambiguated we may begin to assign value to them. Given that we can assign the value 0 to the X and Y values of the cells at left and top borders of the table we can propogate these values through the POU variable graph incrementing at each step.

4.2 Evaluation

We took annual reports for 5 companies (22 documents) and randomly selected 10 tables from each company (50 tables from a total of 662). All the text blocks were marked by hand in each table. An example of the proto-link analysis is shown in Fig. 3 which contains a screen capture of the systems main interface used both for markup and result presentation.

[5] In addition, we look for spanning or cut-in cells that are balanced. If the sets of cells linked to the spanning or cut-in cell are connected then the appropriate X or Y variables are unified. This creates continuity across spanning or cut-in cells for the columns or rows on either side.

[6] A number of heuristics have been implemented which are not presented due to space restrictions.

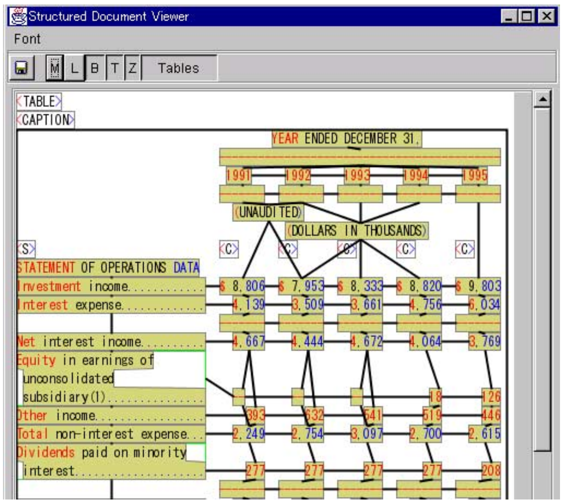


Figure 3. Analysis of a table with a complex header and some mis-aliged columns.

Table 1. Precision and recall statistics.

	Precision	Recall
Proto-link Evaluation	99.20 %	98.85 %
Table Recognition Evaluation	99.59 %	95.37 %

Proto-link Evaluation The precision of a proto-link analysis for a table indicates how many of the hypothesised proto-links were correct. The recall indicates how many of the correct proto-links were proposed. These values are shown in Table 1.

Two major error types were observed. The first we term **reduced spanning**. This occurs when the extent of a spanning cell is significantly reduced from its ideal extent due to the lack of any device (e.g. underline) to compensate. The result is that a subset of the cells that the source spans is missed. The second we term **sibling mis-alignment**. This occurs generally in columns of aligned cells when the extent of the cell's immediate neighbours are not visible via the methods described above.

Table Recognition Evaluation Evaluating the recognition of a table is, in itself, an interesting research topic, and certainly a non-trivial problem. One ground truth against which the table should be judged is the ideal spatial table. An inspection of the output of a table recognizing system might test to see if a cell is in the correct space in the abstract table. However, this is not a reliable indication of the algorithm unless we were only interested in the ability to get all of the table correct. If a cell or set of cells above or to the left is incorrect, then there may be a knock on effect to the cell in question.

Additionally, it is hard to give a meaningful evaluation to a spatial recognition program as it would need to provide different rewards and penalties to results depending on the function of the cells - access cells in the stub and head, and data cells in the body of the table. This is perhaps the motivation behind the approach presented in [2] to an evaluation based on queries asked of the table rather than an inspection of the relative positions in the table. These queries are answered via navigation of a logical structural model. The evaluation in [1] uses a similar approach based on an inspection of how many of the 'line items' (the access cells in the stub) were correctly extracted.

In this paper we evaluate the spatial representation in terms of the ideal table as defined above. The evaluation indicates the *potential* for the derived spatial table to produce the correct logical table. Errors of omission are those cases where there is logical relationship between two cells but they are not aligned. Errors of addition are those where the spanning condition doesn't hold. The precision and recall results are shown in Table 1.

The errors observed in the table recognition evaluation have an obvious relationship to the errors reported for the proto-link evaluation. Additional errors were due to the systems inability to understand `ASCII` line art.

There are two approaches to evaluation: the functional approach ([2]) in which queries of the resulting model are made and compared to ground-truth, and the absolute approach discussed here. Full comparisons are not yet possible without further standardisation and data set creation.

5 Conclusion

This paper has presented an approach to table recognition which uses local spatial constraints to overcome impoverished table presentation and errors in ASCII tables.

References

[1] D. Ferguson. Parsing financial statements efficiently and accurately using c and prolog. In *PAP '97*, April 1997.

[2] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. A system for understanding and reformulating tables. In *Fourth ICPR Workshop on Document Analysis Systems*, Rio De Janeiro, Brazil, December 2000.

[3] M. Hurst. *The Interpretation of Tables in Texts*. PhD thesis, University of Edinburgh, School of Cognitive Science, Informatics, University of Edinburgh, 2000.

[4] M. Hurst. Layout and language: An efficient algorithm for text block detection based on spatial and linguistic evidence. In *Document Recognition and Retrieval VIII*. SPIE, 2001.

[5] T. Kieninger and A. Dengel. A paper-to-html table converting system. In *Proceedings of Document Analysis Systems (DAS) 98*, Nagano, Japan, November 1998.

[6] A. Laurentini and P. Viada. Identifying and understanding tabular material in compound documents. In *International Conference on Pattern Recognition*, 1992.