Complex Table Form Analysis Using Graph Grammar

Akira Amano[1] and Naoki Asada[2]

[1] Kyoto University, Kyoto 606-8501, Japan,
amano@i.kyoto-u.ac.jp
[2] Hiroshima City University, Hiroshima 731-3194, Japan,
asada@its.hiroshima-cu.ac.jp

Abstract. Various kinds of complex table forms are used for many purposes, e.g. application forms. This paper presents a graph grammar based approach to the complex table form structure analysis. In our study, field types are classified into four, i.e. blank, insertion, indication, explanation, and four kinds of indication patterns are defined between indication and blank or insertion. Then, two dimensional relations between horizontally and vertically adjacent fields are described by graph representation and those reduction procedures are defined as production rules. We have designed 56 meta rules from which 6745 rules are generated for a complex table form analysis. Experimental results have shown that 31 kinds of different table forms are successfully analyzed using two types of meta grammar.

1 Introduction

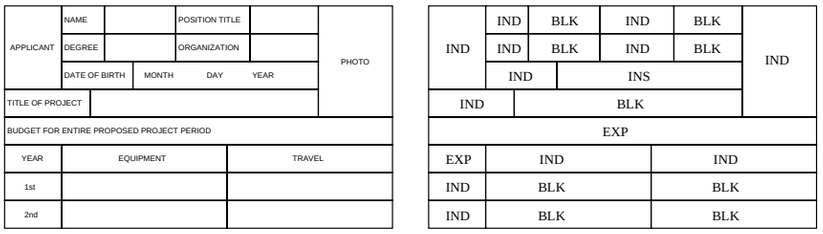
Various kinds of form documents are in circulation around us such as research grant application sheets to which we need to fill in appropriate data to send some information to others.

One popular type of form document is table form document which are widely used in Japanese public documents. Although many researches have been done for automated table processing[1], there are few researches which extracts semantic (structural) information. Among them, production system based systems[2][3][4] have been proposed, yet, they have drawback that modification of structural knowledge is annoying. Practically, it is very important to adapt structural knowledge to each document type, as there exist large variety of table form documents.

For this problem, system using grammatical representation for the structural knowledge have been proposed. In the system proposed by Rahgozar et. al.[5], graph grammar is used for the representation of the knowledge. However, document structure considered in this system is quite simple compared to prior ones such as [2].

We have proposed table form structure analysis system using ordinary one dimensional grammar[6]. In this system, simple grammar is used for the analysis of complex document structure. As the grammar is very simple, it is easy to

D. Lopresti, J. Hu, and R. Kashi (Eds.): DAS 2002, LNCS 2423, pp. 283–286, 2002.



(a) original image (b) box classification result

Fig. 1. An example of table form document.

modify and maintain consistency of them. However, as the table form documents have two dimensional structure, the structure analysis part of the system handles two dimensional information which are not described in the grammar. Thus, some part of the structural knowledge were embedded in the analysis part of the system, that leads to difficulty in modifying structural knowledge in some case.

In this paper, we propose table form structure analysis system based on graph grammar which can handle complex table structure. As the structural knowledge is fully expressed in the grammar, we can easily modify it to suit various kinds of documents.

2 Document Structure

The system deals with documents that consist of rectangular fields formed by horizontal and vertical rules as shown in Fig.1(a). In this paper, each field is called *box* which is considered as a primitive element of document structure. Boxes are classified into four types, `BLK`(blank box to be filled-in), `INS`(insertion box to be inserted or pasted between preprinted letters), `IND`(indication box that indicates other boxes) and `EXP`(general explanation box) according to the database. Figure 1(b) shows the box types of Fig.1(a).

Box indication patterns considered in our system is same as those in [2]. The indication box plays an important role in the document structure analysis; that is, the function of the blank and insertion boxes are determined by the left or upper adjacent indication box, and such a horizontal or vertical relation is always established when both boxes have the same height or width, respectively. This means that the unification of an indication box and its associated blank or insertion one forms a rectangular block like a box, so we call it a *compound box*. This unification also takes place in the situation that a compound box is associated with adjacent indication box. Note that, there are two types of unification; one is one dimensional unification in which one indication box and one associated box is unified, and the other is two dimensional unification in which two indication boxes placed above and left of their associated box are unified.

(a)same height (b)included (c)included(reverse) (d)don't connect

Fig. 2. Symbols for adjacent box connectivity. (edge going to the right only)

3 Document Structure Grammar

Two dimensional document structure can naturally be denoted by graph grammar. Graph grammar is a four-tuple $\{\Sigma, \Delta, \mathrm{S}, \mathrm{P}\}$ where Σ is node labels of 4 box types and 9 compound box types, and Δ is edge labels shown in Fig.2. Note that each edge has direction attribute, therefore, edge label becomes combination of four edge types(a,b,c,d) and four edge directions(l,r,u,d). S is starting symbol `document` which represents whole document. P denotes a set of productions that are of the form $p = (L, R, E)$ where L and R are lhs and rhs graphs of the production rule p respectively, and E is a set of embedding rules. Embedding rules are of the form $\{v_1, e_1, n_1, v_2, e_2\}$ where edge labeled e_1 from node v_1 to node of the label n_1 in rhs is replaced with the edge label e_2 from node v_2.

For example, adjacent `IND` and `BLK` boxes in upper left part of Fig.1 becomes `hicb` (horizontal indication compound box) as shown in upper part of Fig.3 and corresponding production rule is shown in lower part of the figure. Note that, a set of production rules for producing `hicb` from `IND` and `BLK` are used according to the variety of combination of edge label. Afterwards, `Hicb`s are converted to `gcb`s (general compound boxes) by another set of rules, and they are combined to one `gcb`. Finally, leftmost `IND` and adjacent `gcb` become `hicb` and it is converted into a `gcb`.

For the two dimensional part, first, two `IND` and one `BLK` boxes in left top corner are converted into `vci` (vertical cell indication), `hci` (horizontal cell indication) and `cel` (cell box) as shown in upper part of Fig.4 and corresponding production rule is shown in lower part of the figure. Similary, `IND` boxes are converted into `vci` and `hci` boxes, and `BLK` or `INS` boxes are converted into `cel` boxes by another set of rules. Afterwards, every `cel`s are combined into one `cel`,

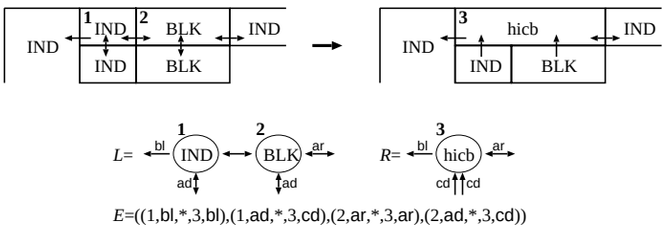


Fig. 3. A production rule for compound box.

and adjacent `vcis` and `hcis` are combined with it. Finally, together with left top `EXP`, they are combined into one `table`.

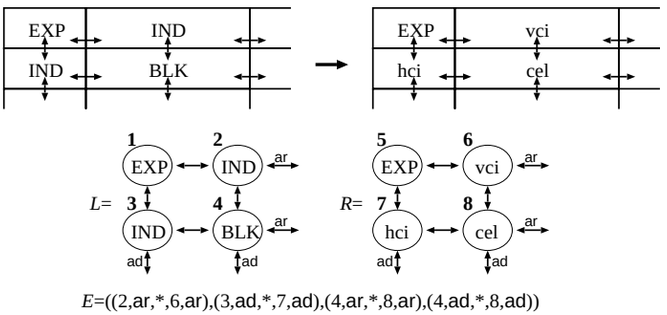


Fig. 4. A production rule for table.

Finally, we used 6745 rules to analyze Fig.1. Note that they are generated from 56 meta rules as they are combinations of geometrical and logical box relations. Experimentally, 31 table form documents were successfully analyzed with two types of meta grammar.

4 Conclusion

In this paper, we revealed that the graph grammar can be a powerful tool for structure analysis system of complex table form documents. We have shown that the system can deal with complex table forms considered in prior systems. Owing to its ability of expressing 2 dimensional relations, the grammar can easily be extended to deal with various complex table forms.

References

1. Lopresti, D., Nagy, G.: A Tabular Survey of Automated Table Processing. LNCS **1941** (2000) 93–120.
2. Watanabe, T., Luo, Q., Sugie, N.: Layout Recognition of Multi-Kinds of Table-Form Documents. IEEE PAMI **17** 4 (1995) 432–445.
3. Cesarini, F., Gori, M., Marinai, S., Soda, G.: INFORMys: A Flexible Invoice-Like Form-Reader System. IEEE PAMI **20** 7 (1998) 730–745.
4. Bing, L., Zao, J., Hong, Z., Ostgathe, T.: New Method for Logical Structure Extraction of Form Document Image. SPIE Proc. **3651** (1999) 183–193.
5. Rahgozar, M., Cooperman, R.: A Graph-based Table Recognition System. SPIE Proc. **2660** (1996) 192–203.
6. Amano, A., Asada, N., Motoyama, T., Sumiyoshi, T., Suzuki, K.: Table Form Document Synthesis by Grammar-Based Structure Analysis. 6th ICDAR (2001) 533–537.