表の配置グラフに基づく X-Window 上での表操作ツール TableX について

尾々野正和 * 古川善吾 ** 牛島和夫 *

* 九州大学大学院システム情報科学研究科
** 九州大学 情報処理教育センター

文書の整形において表は重要な役割を果たしている。表においては項目の間の関連性を包含関係によって記述することが可能である。この表をモデル化すると、印刷のための物理構造と、データ間の関連を示す論理構造がある。物理構造における表枠の包含関係に基づく階層構造を配置グラフという木構造で表現することができる。現在までに、配置グラフに基づいて LaTeX 上で表を記述する tree 環境を試作した。

しかしながら、LaTeX のテキスト形式による記述では、表の項目の包含関係の把握が困難である。これを解決するために X-Window 上で配置グラフに基づいて、表および配置グラフの表示、表要素の挿入削除等の編集、tree 環境を用いた LaTeX のテキスト出力、を行なうツール TableX を試作した。TableX を用いることによって表の構造の記述が容易になることが確認できた。実現に当たっては、配置グラフの構造が操作の順序によって異なることを避けるために標準形式を導入した。

A Prototype of a table manupulation tool based on a location graph using the X-Window system.

Masakazu Ohno *, Zengo Furukawa** and Kazuo Ushijima*

*Graduate School of Information Science and Electrical Engineering, Kyushu University.
**Educational Center for Information Processing, Kyushu University.
6-10-1 Hakozaki, Fukuoka 812, Japan.

A table plays important roles for formatting a document. Relationships of items in a table can be described using includings of items and directions of item lines. A table model includes a location model and a logical relationship. The location of items for printing is described in a location graph as a tree structure which describes including relationships of items. The **tree** environment in LaTeXwas implemented for describing a tree structure of a location graph.

A description of a **tree** environment with text is complex and it is difficult to understand the described **tree** environment. A tool TableX supports the description and understanding of the **tree** environment on the X-Window system. TableX reads a **tree** environment from a file and accepts instructions for editing from a keyboard and a mouse. Text of the **tree** environment can be written in a file. The standard form of a tree structure of a table is used for preserving the uniformness of operations.

1. はじめに

日本語の文書において、表は基本的な構成要素である。現在日本語の文書を書く多数のツールがあり、表生成機能を備えている。しかしながら、これらの表生成機能は、利用者の要求を充分に満たしていないことがある。利用者の要求に少しでも応えるために、これまで表の特性を分析し、そのモデル化と表生成システムの試作を行なってきた[1]。

表には印刷のための物理構造と、データの間の関連を示す論理構造がある。物理構造をモデル化したものを「表物理モデル」と呼ぶ。表物理モデルでは、表枠の包含関係に基づいた階層構造を木構造で表現し、それを「配置グラフ」と呼ぶ。これまでに LaTeX 上で配置グラフに基づいて表を記述する tree 環境を試作した[2]。

表が複雑になると tree 環境を用いた記述と出来上がる表との対応を頭の中だけで考えるのは困難である。そこで、X-Window 上で配置グラフに基づいて、表および配置グラフの表示、表要素の挿入削除等の編集、tree 環境で記述した表のテキスト出力、を行なうツール TableX を試作した。

2. 表物理モデル

表物理モデルは、印刷される表の線や文字、大きさ、順序などの物理的な配置情報をモデル化するものである。

表物理モデルは、以下に示す要素で構成される。

1. 線：表の枠を表す。直線だけでなく、見えない線や点線もあり得る。
2. 枠：４本の線で構成されており、内容や見出しを記入する範囲を指定する。また、表をいくつかの領域に分割していると考えることができる。
3. 文字：内容や見出しのデータは、文字によって表される。
4. 文字列：文字の列である。
5. 領域：枠によって区切られた範囲である。この中に行を並べる。
6. 行：枠で区切られた領域の幅は有限であり、文字列の長さをその範囲内におさまる形に制限する必要がある。文字列を並べる単位を「行」と呼び、枠の中を行に区切る。この行の中に文字列が書かれる。
7. 箱：領域と枠を合わせたものである。箱を並べることにより表を組み立てることができる。
8. 物理表：箱の並べ方を規定するために導入する抽象的なものであり、以下のように定義する。
   (a) 箱は物理表である。
   (b) 物理表を並べたものは物理表である。物理表の並べ方には、以下の３つがある。（）内は並びの演算子であり、n は並ぶ物理表の数を示す。
   - 縦並び（＝n）
   - 横並び（‖ n）
   - 格子型並び（＋）
9. 属性継承（$\overset{\text{属性}}{\longrightarrow}$）：「高さ」や「幅」などの属性を継承する場合がある。
10. 属性合成（$\overset{\text{属性}}{\Longrightarrow}$）：「縦並びの物理表の高さが、並んでいる物理表の高さの和である」という形において現れる。

以上の構成要素によって表の物理構造を記述する。

また、表物理モデルでは、物理表の包含関係に基づいた階層的関係を考えることができる。階層的関係を木構造で表現したものを「配置グラフ」と呼ぶ。表物理モデルでは、データ間の関連性を考慮しない。このため、物理表の包含関係のとらえ方によって、配置グラフは何通りも考えられる。例えば、図 1 の表に対して図 2 に示す配置グラフを考えることができる。

見出し１		
見出し２	見出し３	見出し４
	内容３	内容４
見出し５	見出し６	見出し７
	内容６	内容７

図 1: 表の例

3. tree 環境

tree 環境は、配置グラフに基づいて、表を木構造で記述する LaTeX 上の環境である。図 1 の表を 図 2 の配置グラフに基づいて tree 環境で

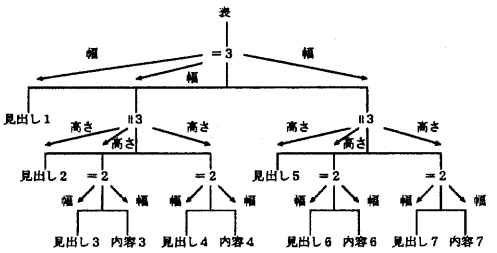

図 2: 配置グラフの例

記述した例を 図 3 に示す。木構造の記述中の v と h はそれぞれ縦並びと横並びを示す。木構造は、各ノード毎に、(親ノード, 並びの種類, 子ノード,···, 子ノード) という形で記述する。

```
\begin{tree}
% 木構造の記述
\set{(top,v,term1,term2,term3)
(term2,h,term4,term5,term6)
(term5,v,term7,term8)
(term6,v,term9,term10)
(term3,h,term11,term12,term13)
(term12,v,term14,term15)
(term13,v,term16,term17)}
% データの記述
\term1{見出し1}
\term4{見出し2}
\term7{見出し3} \term8{内容3}
\term9{見出し4} \term{10}{内容4}
\term{11}{見出し5}
\term{14}{見出し6} \term{15}{内容6}
\term{16}{見出し7} \term{17}{内容7}
\end{tree}
```

図 3: tree 環境で記述した例

LaTeX における tabular 環境では、表の構造のための記述とデータとが入り混じっているためデータを一括して変更することが困難である。これに対して、tree 環境では、図 3のソーステキストから分かるように、木構造の記述部分とデータの記述部分が分離しているため、データの記述部分のみをさしかえることにより、データを一括して変更することが容易になる。また、木構造の記述部分を保存しておくことにより、スタイルファイルとして利用することも可能である。

しかしながら、図 1 程度の表でも、木構造の記述は理解が困難である。表がさらに複雑になると、木構造の記述と出力結果との対応はますます把握が難しくなり、木構造の記述を頭の中だけで考えるのは困難である。これは、tree 環境の木構造の記述法の分かりにくさよりも、そもそも LaTeX がコマンド埋め込み方式であるので、コンパイルしないと出力結果が分からない、ということが大きな原因である。木構造の記述と出力結果との対応を把握する負担を軽減するためには、WYSIWYG(What You See Is What You Get) 方式の GUI(Graphical User Interface) が必要である。tree 環境で記述したソーステキストを編集するのではなく、X-Window 上に表示した「表」を編集することにより、ソーステキストを自動的に書き換えるシステムが実現できれば、木構造を記述する際の負担を軽減できる。

4. TableX

4.1 概要

3 章で示した tree 環境を用いて X-Window 上で表を操作するツール TableX を試作した。TableX のデータの流れを図 4に示す。

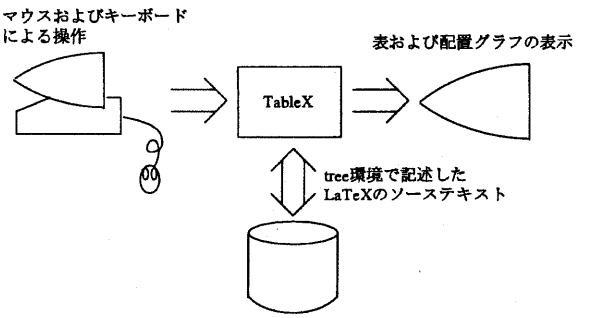

図 4: システム概要

入力、及び出力は以下の通りである。

[入力]

1. tree 環境で記述した LaTeX のソーステキスト
2. マウスおよびキーボードからの入力

[出力]

1. ディスプレイ上の表の表示 (メイン画面: 図 5参照)
2. 表の配置グラフの表示 (配置グラフ表示画面: 図 6参照)
3. tree 環境で記述した LaTeX のソーステキスト

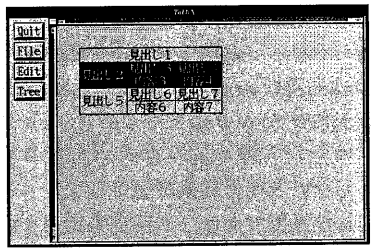

図 5: メイン画面

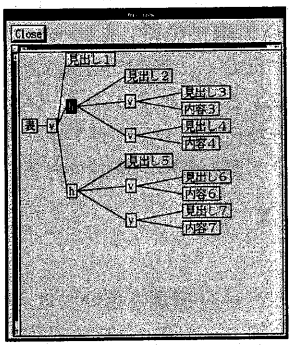

図 6: 配置グラフ表示画面

4.2 機能

4.2.1 起動・終了機能

起動: コマンドラインからコマンド TableX により起動

終了: 表表示画面の Quit ボタンを押すことにより終了

4.2.2 入出力機能

メイン画面の File ボタンを押すと図 7のメニューが表示され、4.1節で示した入出力のうち、tree 環境で記述した LaTeX のソーステキストの入出力を行なうことができる。また、メニュー中の New を選択すると表示される画面上で、文字列の入力を行なうことにより、新規に表を作成することができる。

図 7: File ボタンのメニュー

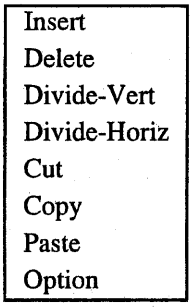

図 8: Edit ボタンのメニュー

4.2.3 表編集機能

- マウス操作による物理表の選択
  削除、挿入、分割、移動・複写といった編集作業は、マウス操作により、物理表を選択した後行なう。マウスの操作は表 1の通りである。

表 1: マウス操作による物理表の選択

マウス操作	配置グラフでの意味
表の上で左ボタンをクリック	マウスポインタが指す配置グラフの葉を選択
選択された物理表の中で左ボタンをクリック	1 つ上のレベルのノードに移行
選択された物理表の中で右ボタンをクリック	1 つ下のレベルのノードに移行

  選択された物理表および配置グラフ上のノードを反転表示する。(図 5、図 6参照)
- 文字列の変更
  文字列を変更したい箱の上で、マウスの中ボタンをダブルクリックする。文字列入力の画面が表示されるので、新しい文字列を入力することで文字列の変更ができる。
- 削除
  マウス操作により、物理表を選択する。次に、Edit ボタンを押すと表示されるメニュー (図 8) から、Delete を選択することにより削除できる。図 9で、data2 を選択し、削除を行なうと図 10のようになる。
- 挿入
  マウス操作により、物理表を選択する。次に、図 8のメニューから、Insert を選択すると挿入画面 (図 11) を表示するので、文字

data1	data2
	data3

図 9: 簡単な表の例

data1	data3

図 10: データ削除の例

列の入力および挿入位置の指定を行なうことにより、挿入できる。図 9で、data1 を選択し、data1-1 を下 (below) に挿入するように指定すると、図 12のようになる。挿入位置としては、下の他に、上 (above)、左 (left)、右 (right) を指定できる。

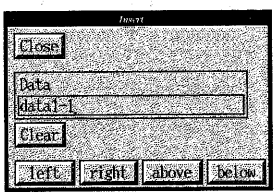

図 11: 挿入画面

data1	data2
data1-1	data3

図 12: データ挿入の例

- 分割
  マウス操作により、物理表を選択する (ただし、分割の場合、選択するのは葉ノードでなくてはならない)。図 8のメニューから、Divide-Vert あるいは Divide-Horiz を選択すると、縦分割あるいは横分割が行なえる。分割の結果新たに生成される葉ノードの文字列としては、選択していた葉ノードの文字列が複写される。図 9で、data1 を選択して、縦分割を行なうと、図 13のようになる。

data1	data2
data1	data3

図 13: 分割の例

- 移動・複写
  マウス操作により、物理表を選択する。図 8のメニューから、Cut あるいは Copy を選択する。Cut を選択した場合には、選択していた物理表が削除される。次に、マウス操作により、表の一部を選択した後、図 8のメニューから、Paste を選択する。貼りつける位置を指定する画面が表示されるのでその指定を行なうことにより、Cut もしくは Copy していた領域を貼りつけることができる。貼りつけ位置の指定法は、挿入の場合と同様である。
  図 9で data2 および data3 を選択し、その領域を Copy する。そして、再度 data2 および data3 を選択して右側に貼りつけると図 14のようになる。

data1	data2	data2
	data3	data3

図 14: 複写の例

5. 評価・議論

5.1 木構造の違い

木構造が異なっていても、表が同じになる場合がある。例えば、図 15の (a) と (b) は、共に (c) の表を表現した配置グラフである。

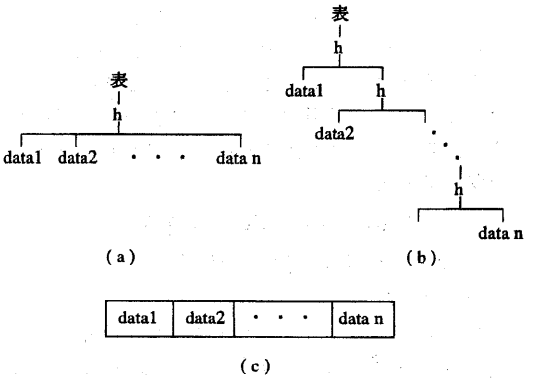

図 15: 木構造の違い

マウス操作により物理表を選択する際に、木構造が異なると操作が変わることがある。図 15(a) の木構造を持つ場合、data n の箱の上でマウスの左ボタンを 2 回クリックすると表全体が選択される。一方、図 15(b) の木構造を持つ場合では、data n の箱の上でマウスの左ボタンをク

リックする度に箱1つ分ずつ選択される領域が増えていき、表全体を選択するには、n回のクリックが必要となる。図15(b)の木構造を持つ場合でも、data 1の箱の上でマウスの左ボタンをクリックする場合には、2回のクリックで表全体が選択される。このような違いは、利用者を混乱させる可能性がある。

そこで、木構造をできるだけ一意に定めるように、挿入や貼りつけを行なう際に、木構造を標準化することにした。標準化とは、並びが親と等しい場合には、同一レベルの子にすることである。図15の例でいうと、(a)のような木構造にすることである。

しかしながら、図15(c)で、data 1とdata 2の部分だけを選択したいとしても、図15(a)の木構造では、不可能である。このため、図15(b)のように標準化の行なわれていない木構造も利用者が意図する場合には、作成できる必要がある。

そこで、木構造の標準化を行なわずに箱を増やす分割という操作を用意した。表を作成していく過程で、挿入と分割を使い分けることにより、表の意味(論理構造)の一部を配置グラフで表現することが可能になる。

しかしながら、既に作成してしまった表の場合、論理構造を表現するために、表の一部分の木構造を変更したいことがあっても、現在のところ、その部分を一旦削除して新たに作成する必要がある。このような手間を削減するためには、表の内容を変えず木構造だけを変更する操作が必要となる。これを実現するには、非終端ノードの挿入・削除といった操作を配置グラフ表示画面上で実現する必要がある。

5.2 TableXの機能に関する問題点

TableXの機能に関する問題点として、次のことが挙げられる。

- 表物理モデルにおいて定義されている中で、tree環境およびTableXで実現されていない項目がある。例えば、表枠の線種として実線以外に破線や見えない線も定義しているけれども、現在のところは実線のみが扱える。また、並びの種類として格子型並びを定義しているけれども、利用できるのは縦並びと横並びのみである。これらの定義については、今後、tree環境およびTableXで実現する必要がある。
- 表の一部の選択は、木構造をたどりながら行なうため、部分木になっていない領域を選択し編集することができない。これについては、格子型並びの取り扱いや5.1節で述べた木構造だけを変更する操作を実現することにより、解決できると考えられる。
- 現在まで、構造の編集に重点をおいてTableXの試作を行なってきたので、データとなる文字列に関する処理は十分ではない。文字列に関して行なう必要があるのは、行への分割、領域の大きさに合わせた文字サイズの選択、である。すなわち、枠の幅あるいは高さが指定された時、長過ぎる文字列を折り返したり、文字サイズを調整して領域内に文字列を格納する必要がある。

6. おわりに

X-Window上で配置グラフに基づいて、表を操作するツールTableXを試作した。また、TableXを用いることによって表の構造の記述が容易になることを確認した。

今後の課題として、以下のことが挙げられる。

- 表物理モデルでは、定義されているがtree環境およびTableXで実現されていない点を実現する。
- 表と表を合成することにより新たな表を作成する機能や、配置グラフ表示画面での編集作業、アンドゥ機能などを実現し、TableXの機能を拡張する。

参考文献

[1] 三浦好弘 : 複雑な表を生成するための表生成系の開発, 九州大学大学院工学研究科修士論文,1993.

[2] 橋本俊一, 古川善吾, 牛島和夫 : 表の物理構造に基づいた表作成ツールのLaTeXにおける実現, 情報処理学会九州支部研究会報告, pp271-279, 平成8年3月13日.

[3] 尾々野正和, 古川善吾, 牛島和夫 : 表の物理構造を表現した配置グラフのX-Window上での操作ツールの試作, 情報処理学会第53回全国大会講演論文集, pp. 4-135-136, 1996.