

ブロックの階層性を用いた表の自動加工制御法

1K-5

清末 徳之、児島 治彦、秋山 照雄
NTTヒューマンインタフェース研究所

1. まえがき

画像として入力した文書を認識し、内容をコード化しておけば、文書の再加工が容易になる。この再加工を円滑に行うためには、ユーザが指示した箇所以外の所に生じる歪をなくす加工、いわば修正のための付加的な加工を自動的に行う機構が必要である⁽¹⁾。そこで我々は、これまでに表を対象に、自動加工のためのデータ構造と制御構造について検討してきた⁽²⁾。今回はこれらの構造を用いて幾つかのオペレーションをインプリメントし、システムの実現可能性を確認したので報告する。

2. 表のデータ構造

2. 1. ブロックによる表の表現

ここで対象とする表の例を図1に示す。図のように表の見出し項目(カテゴリブロック[A]~[H])は階層性をもつ場合があること⁽³⁾、表の加工は主としてそのブロック内データに対して行われ、データを包含するブロックに対する処理は付加的な加工であることを考慮して、ブロックをインデックスで管理するデータ構造を提案してきた⁽²⁾。ここでブロックとは罫線で囲まれた表中の方形枠領域のことである。即ち、ブロックは文字などのデータを包含し、この変更(例えば文字の追加)に伴う、データを包含するブロックはもとより他のブロックのサイズを変更する付加的な加工が必要である。

2. 2. インデックス管理構造

我々は個々のブロックの識別番号をインデックスと呼び、属性として付与する。図1の表中ブロックにインデックスを付与した例を同図中に示す(例えばブロック[A]のインデックスは(0,0))。1つのブロックには、インデックスベクトルとして表の次元の数だけのインデックスが付与される。表の次元とは、表の見出しの軸の数であり、通常は縦方向、横方向の2軸を備えた次元2の表が用いられる。従って、このインデックス構造は次元3以上の表に対してもそのまま拡張できる。インデックスの値は、表のカテゴリブロックの階層構造をもとにして決定される。

3. 表加工のための制御構造

3. 1. 自動加工

付加的な処理を自動的に行うためには、データ構造の変更やオペレーションの実行に関し、拘束関係をもとにしてこれらを管理するルールベースが必要である。データ構造はインデックス管理構造を用いる。罫線を除いた表の加工に関するオペレーションの種類を表1に示す。

[A] 0, 0	[B] 1, 0		
	[C] 1 1, 0	[D] 1 2, 0	
		[E] 1 2 1, 0	[F] 1 2 2, 0
[G] 0, 1	[I] 1 1, 1	[K] 1 2 1, 1	[M] 1 2 2, 1
[H] 0, 2	[J] 1 1, 2	[L] 1 2 1, 2	[N] 1 2 2, 2

図1. 対象とする表およびインデックス付与例

表1. 表加工に関するオペレーション

対象	オペレーション	
	基本オペレーション	複合オペレーション
表	サイズ変更	—
ブロック	サイズ変更	行(列)削除 ・挿入
データ (文字)	フォント変更 文字挿入	(行(列)) 交換・削除

3. 2. ルール

ルールには、表とブロック、ブロックとブロック、ブロックとデータ間のサイズに関する拘束関係が記述されている。これを満足していない要素が存在したとき、拘束を満足するようにオペレーションを起動する。オペレーションと表を構成する要素、およびそれらを結び付けるルールの存在を示したものが図2である。図中上部の楕円はオペレーションで、下部の円はデータ構造で表現されている各要素を示している。また両者を結ぶ矢印のリンクがルールの存在を示している。

3. 3. ステータスフロー

付加的な加工は、それまでにどのような加工がなされたかを考慮して行う必要がある。従って、ルールがオペレーションを選択するにあたり、注目している要素が加工済みかどうかを判断するための制御情報が必要である。このためにstatusという属性を各要素に付与する。加工に伴って変化するstatusのフロー図を図3に示

"An Automatic Control Method for Table Processing
based on its Hierarchical Block Structure"

Yasuyuki KIYOSUE, Haruhiko KOJIMA, Teruo AKIYAMA
NTT Human Interface Laboratories

す。加工が施されていない要素のstatusはstaticであり、加工の対象となるブロックの処理、および付加的な加工処理が行われている間activeになる。付加的な加工も含め全ての加工が終了すれば再びstaticになる。なお、加工が失敗した時のリカバリのためstatusがstaticからactiveになるときにstatic要素のコピーを予めとっておく。後続する付加的な加工が失敗した場合はactiveな要素を削除し、このコピーした要素を用いることによって、加工が行われる前の状態を再現する。

3. 4. サブステータスフロー

status情報に加え、ブロックの階層構造を考慮して加工の伝搬を制御する必要がある。ここでは以下の2つの禁止項目を設けた。

- ① 下位レベルのブロックから上位ブロックに対して加工が伝搬された場合は、他の下位レベルのブロックへ加工が伝搬することを禁止する。例えば、図1[E]が横方向に拡大された場合は[D]に加工が伝搬し同様に横方向への拡大が行われるが、この加工が[F]へ伝搬することを防止する。
- ② 下位レベルであるが、直接の親子関係を有さないブロック（甥ブロック）から加工が伝搬された場合は、同位レベルへの加工の伝搬を禁止する。例えば、図1[E]が縦方向に拡大された場合は[C]に加工が伝搬し同様に縦方向への拡大が行われるが、この加工が[D]へ伝搬することを防止する。

これらの加工の伝搬を制御するため、各要素にsub-statusという属性を付与する。このsub-statusフローを図4に示す。①の場合、加工が伝搬されたブロックはupper-activeとなり、以後下位レベルのブロックへ加工を伝搬しない。②の場合、加工が伝搬されたブロックはno-side-activeとなり、以後同位レベルのブロックへ加工を伝搬しない。

4. 実験結果

以上述べたデータ構造、ルールベース、および表1の基本オペレーションを、ミニコン上で、プロダクションシステム構築用の言語とFORTRANで実現した。表示メニューからオペレーションを選択し、実時間で処理を行うことができる。

5. まとめ

多様なメディアを含む文書を自動的に加工するための初期段階として表のデータ構造とその自動加工のための制御構造を提案した。これによって、ユーザが指示した初期の加工を行った結果生じる付加的な加工操作を行う必要がなくなり、ユーザの負担を軽減できる機構を実現できた。今後は基本的な構造をベースに罫線の省略が行われている表など、より多様な表に対処できるように発展させる。

[参考文献]

- (1) 田原他 第37回情報処理学会大会「データ加工のためのデータ構造の実現の一手法」
- (2) 清木他 第37回情報処理学会大会「再加工を考慮した表のデータ構造に関する一考察」
- (3) 児島他 第37回情報処理学会大会「複雑な構造を持つ表の認識に関する基礎検討」

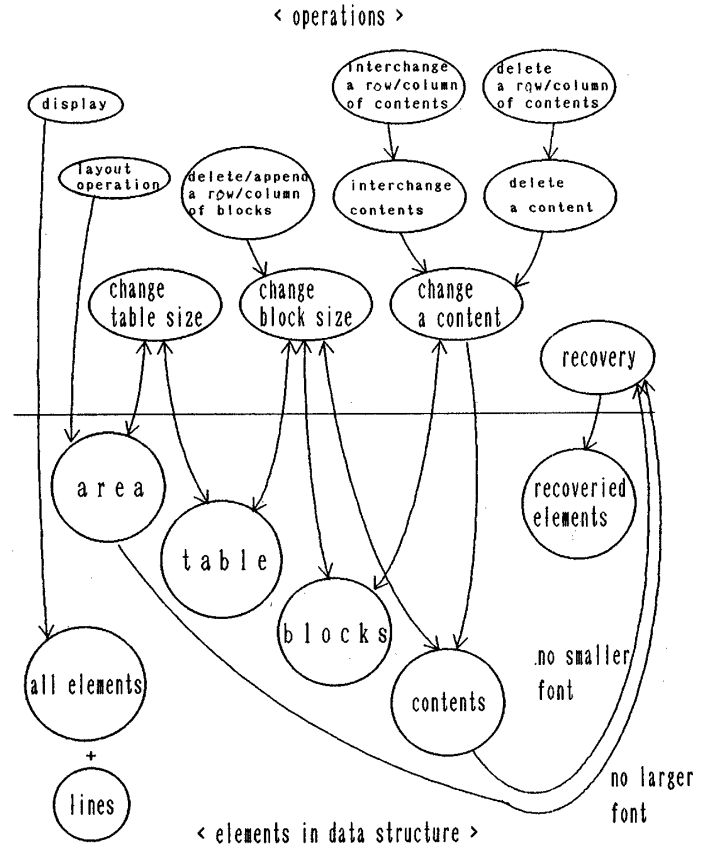


図2. オペレーション、データ構造、ルール

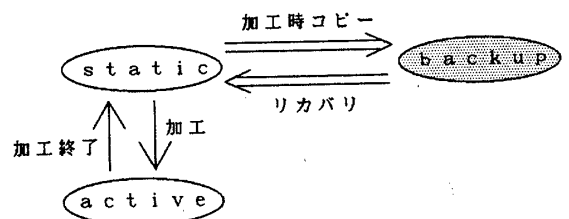


図3. statusフロー

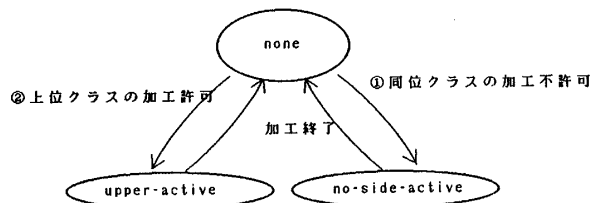


図4. substatusフロー