

日本語 APL における  
帳票作成に関する一考察

2Y-6

金子 聡

日本アイ・ビー・エム株式会社

サイエンス・インスティテュート

1. はじめに

オフィスにおける事務作業には帳票と呼ばれる定型的な伝票や帳簿類の作成、処理が多く見受けられる。その他にも種々のデータを表の形式でまとめる場合も多く、帳票類を作成・処理するための専用のソフトウェアも市販されている。しかし、あらゆるケースを満足するような汎用的なソフトウェアを開発することは困難であり、また使いやすさの観点からも検討の余地があると考えられる。これに対してユーザー自身が簡単に必要な帳票処理機能を備えたプログラムを作成できることが一つの方向と考え、そのためのプログラム (APL ではユーザー定義関数) を日本語 APL を用いてユーザー自身が直接書く場合に、これを容易に行なうために必要となる罫線、下線等の指定に用いる属性データの構成方法について考察した。

上記の各々のレベルに対応するデータと属性値は以下のように構成することができる。つまり、フィールドのレベルではそのフォーマットを指定するデータの中に含まれ、カラム・レベルでは1個の数値マトリックスとして表現される。

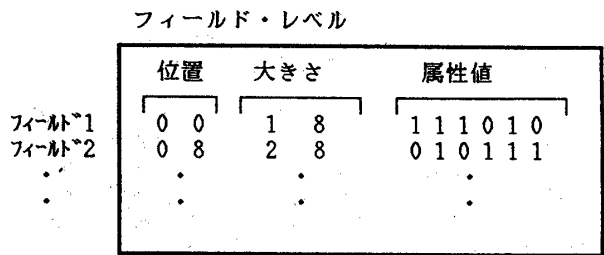


図1-A フィールド・フォーマット

2. APL における表処理

APL はそのオブジェクトの基本型を配列としているため、作表を前提とした作業に対しては極めて有効な言語であるが、一般に帳票や表は罫線によって各フィールドを分けて表示、印刷するものが多く、従って、作表プログラムの容易性は罫線の取り扱ひの容易性に依存するところが大きい。罫線の表示や印刷の方法としては、テキスト列の一部として定義される場合と文字ボックスの一部に併存させて定義される場合がある。日本語 APL で提供される罫線は表示・印刷とも後者に属するため、ここではその場合に限定する。

以後、罫線や下線等を属性と呼び、それを指定するためのデータを属性値と呼ぶことにする。この属性値も作表時の主体となる文字データ同様、APL のオブジェクトとして保持し、APL のセッションやユーザー定義関数の中で操作できることが望ましい。

カラム・レベル

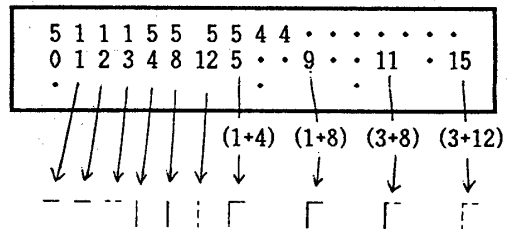


図1-B 属性値マトリックス

3. 作表用属性データ

作表に用いられる属性値はその定義される単位としてフィールド、カラムの2レベルが考えられ、それぞれ次に示すような特徴を持つ。

単位	特徴
フィールド	<ul style="list-style-type: none"> <li>フォーマット指定のデータにより区分される矩形領域</li> <li>表の構造上、または意味上の最小単位に対応</li> <li>罫線は矩形の枠を指定</li> </ul>
カラム	<ul style="list-style-type: none"> <li>文字表示の最小単位に対応</li> <li>詳細な罫線の指定が可能</li> </ul>

表1. 作表データのレベル

表のフォーマットを作成する場合には、通常フィールド単位で定義し、複数のフィールドの定義 (大きさ, 位置, 属性) を含んだ1個の数値マトリックスで表現される。

例えば、図1-Aはフィールドを定義しているフォーマット・データであり、1行で1個のフィールドを表わしている。ここでの属性値はフィールド単位で与えられた属性の内どれが指定されているかを0, 1で表わしたものである。図1-Bでは数値マトリックスの各要素がそれぞれ対応する文字データの1カラム分の属性パターンを定義している事を示している。

属性値の変更もこれらのうちの必要な値を操作することになる。あるフィールドの中に特定の罫線パターンを定義したい場合に、それをフィールド単位で指定するのはパターンが多くなると不都合である。これはカラム単位で定義された属性値のマトリックスを操作することにより対応できる。

4. 日本語 APL による具体例

日本語APLで表を作成・印刷する場合には通常、補助プロセッサAP205(ディスプレイ用)、AP80(プリンター用)を用いる。

提供される属性はAP80、AP205によって異なり、主なものは次の通りである。

	レベル	罫線	下線	明滅	高輝度	反転	太字
		上下左右					
AP205	フィールド	△△△△	○	○	○	○	○
AP80	カラム	● ●	○				○

(注) △:実線のみ  
 ●:実線,太実線,破線の選択可  
 表2. 日本語APLで提供される属性

表2に示されるような属性の違いは、日本語APL固有のものであり無いほうがユーザーの作成するプログラムの構造をより単純化できるが、ディスプレイ、プリンターそれぞれの持つ機能、使用目的の違いを考慮すればある程度の差異は止む終えないものと思われる。属性を定義するレベルの違いもAP205は入出力、AP80は出力のみを対象としているため、生じている。次に、前述の属性データの構成方法を用いた表の印刷・処理の例を示す。

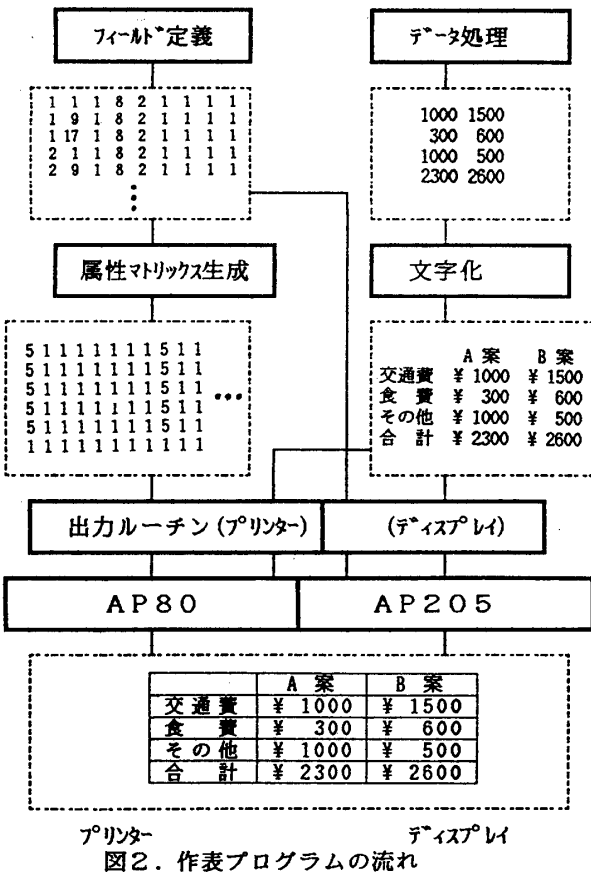


図2. 作表プログラムの流れ

図2に於いてディスプレイの入出力に関してはフィールドのレベルで定義された属性値を含むフォーマットデータと文字データを用いる。一方、プリンター出力はフォーマットデータから展開された属性値マトリックスと文

字データを用いる。属性値マトリックス生成ルーチンはフィールド・フォーマットから位置、大きさ、特定の属性値を用いて、カラム・レベルの属性値マトリックスを生成し、フィールドレベルの属性とカラムレベルの属性のマッピング・テーブルにより適当な変換を行なう。また、属性値マトリックスは1つのAPLオブジェクトであり、ユーザーは必要に応じてこれを直接修正する事によってより詳細な罫線の指定が可能である。

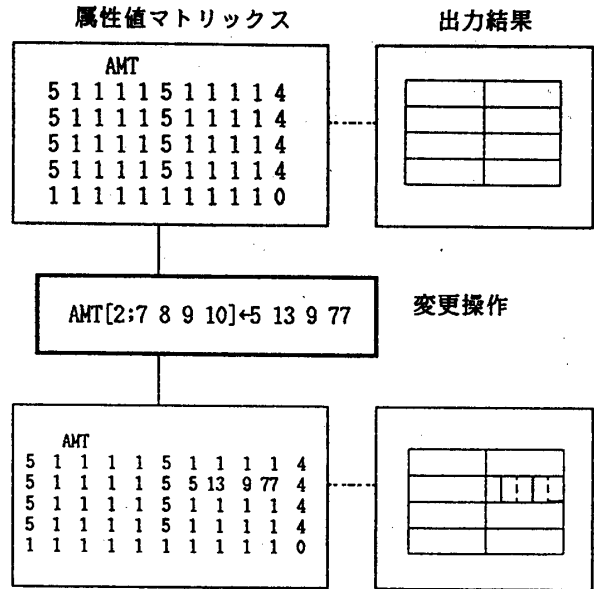


図3. 属性値の変更例

図3は、属性値マトリックスの一部を変更することにより、フィールドのフォーマットを変更することなく、複雑な罫線のパターンを任意の位置に書き込むことができる例を示している。また、この様な属性の変更もその属性値の代入、加算といったAPLの原子関数や作用素を用いて自然な形で行なう事ができるという特徴を持っている。この属性値マトリックスはカラム・レベルの文字データ(1個のオブジェクト)に対して従属的に定義されるが、文字データに対して可能な転置、回転等の操作は適用できない。これは属性値そのものが位置に依存する性質を持つためであり、その様な場合には、一般にフィールド・レベルでのフォーマットの再定義を必要とする。

5. おわりに

日本語APLにおける表作成に関して、フィールドおよびカラム・レベルの属性データを定義し、特にカラム・レベルの属性値マトリックスの利用が、詳細な罫線の指定に有効である事が明らかとなった。ここでは、フィールド・フォーマットの具体的な定義方法やデータ自身についての帳票処理固有の操作方法等に関する検討はなされていない。これらは今後の課題である。

参考文献

- [1] 日本語APLユーザーズ・ガイド, N:SB18-1088
- [2] 中村,他:「帳票作成支援の一方式」, 情報処理学会, 29回全国大会
- [3] 倉沢,他:「表の論理構造と表操作システム」, 情報処理学会, 31回全国大会