

7H-3

盛屋邦彦

有澤博

日立ソフトウェアエンジニアリング(株)

横浜国立大学工学部電子情報工学科

1. はじめに

データを論理的に扱うために各種のファイルシステムがあらわれてきた。しかし直接アクセス装置の機能を十二分に活かしたファイルアクセス法は少ない。たとえばBSAMやQSAM、BDAMではファイルをレコードの順序集合と仮定しているが、ファイルの途中位置での挿入という操作は許されていない。これは物理的な装置の構造に引きずられているためである。

V SAMは高機能なアクセス法であるが、複雑で、自由な途中位置での挿入操作は許されていない。

UNIX*ファイルは非常に論理的であるが、ファイルは単純な文字の並びであることが仮定されており、データを抽象化するためのレコードの概念がない。更に途中位置での挿入は行えない。

本稿ではファイルを抽象データ型と捉え、集合の概念を基とした高機能なファイルアクセス法(AAM)を提案する。

2. AAMのファイル構造と基本概念

ここではファイルをADT(抽象データ型)と捉え[AHU83]、AAMの基本となる概念とその操作を定義する。

要素とその集合

- (1) データの最小基本単位: アトム
アトムは値を持つ。
- (2) アトムの集合: アトム集合
- (3) 集合の集合: 集合集合
- (4) 集合はその要素(アトムまたは集合)に全順序がついた順序集合とする。
- (5) 要素は複数の集合にまたがって存在できない。

要素識別子

- (6) 要素は全て一意の名前の要素識別子(EID)を持つ。
- (7) EIDはその要素が含まれている集合のEID(集合IDと呼ぶ)とその集合の中で一意なキーの結合により表現される。
- (8) キーはユーザが明示的に指定するかシステムが決定する。

階層型ファイル構造

- (9) (5)より集合同士は木構造で関連づけられる。
- (10) 根(root)は全ての集合の頂点である。
- (11) したがってEIDはrootからの完全パス名となる。
- (12) 集合の階層的な構造を反映してパス名は集合IDとキーの間を'/'で区切る。

ファイル構造の例を図1に示す。(大きな楕円は集

合をまた小さい丸は要素を表す。%で始まる数字はシステムが決めたキーである。またこのファイルはアトムを文字列とする。)

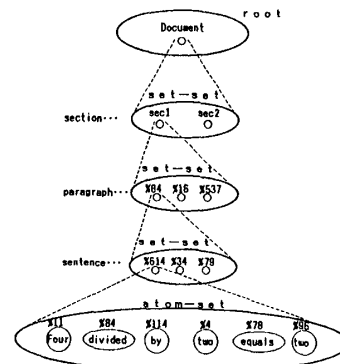


図1. AAMファイル構造

3. ADT操作

AAMの基本概念と操作を示す。

現在子

現在注視している、集合を現在集合、要素を現在要素と呼び、また現在集合の位置を現在集合位置、現在要素の位置を現在要素位置と呼ぶ。この現在子により、巡航的に操作が行われる。またファントムと呼ばれる集合の末尾に置かれる仮想的な要素がある。これは集合が作られたときに必ず一つ存在し消すことができない。集合が空とはファントムしかない時を言う。

更新操作

- (1) insertatom(a, k)
aは挿入するアトムの値でkはキーを指定する。要素(アトム)は現在要素位置に挿入される。
- (2) insertset(t, k)
tは挿入する集合の要素タイプでkはキーを指定する。要素(集合)は現在要素位置に挿入される。
- (3) delete(m)
現在要素位置の前のm個の要素が削除される。これらの操作は現在要素位置を変えない。

現在要素位置の変更操作

- (4) skip(m)
現在要素位置から相対的にm離れたところを現在要素位置とする。
- (5) search(p)
パス名pの要素の位置を現在要素位置とする。

現在要素の情報取得操作

- (6) getvalue()
現在要素(アトム)の値を返す。

Advanced File Access Method AAM
Kunihiko Moriya* Hiroshi Arisawa**
* Hitachi Software Engineering Co. Ltd.
** Yokohama National University

UNIXソフトウェアは米国AT&T社ベル研究所が開発したオペレーティングシステムの名称である。

- (7) gettype()
現在要素 (集合) のタイプを返す。
- (8) geteid()
現在要素のEID (完全パス名) を返す。
- (9) getposition()
現在要素位置 (先頭位置からの距離) を返す。
- (10) getsize()
現在集合に含まれている要素の数を返す。

操作の直交性

AAMは操作の数がなるべく少なくなるように設計されている。更に操作がそれぞれ直交されており、ある処理を行う際、ただ一通りの操作列のみが対応する。

4. AAM操作の関数を用いた記述

操作の関数を用いた記述は、プログラムの検証や最適化、並列化等に利点がある。そこでAAM操作を関数とみた記述法を述べる。AAMの操作はすべて単純値またはnull値が返るためこれを関数とみることができる。関数は最初に引数から評価され、その後関数が評価される。そこで関数の記述を引数を増やした形に拡張する。

function(arg1, arg2, ..., x-arg1, x-arg2, ...) x-arg1, ...は関数の本来の引数ではなく、操作の前提となる副作用を予め起こすためのものである。引数は右から左へと評価される。例えば集合の末尾にアトムを挿入するappend(a, k)は次のようになる。

```
insertatom(a, k, skip(getsize()-getposition()))
```

この操作はまず現在要素位置を集合の末尾にし、そこでアトムを挿入する。

現在要素位置は副作用により変化するため、現在の状態を保存する[X]記述を導入する。これはpascal風に記述すると次の様なものである。

```
push(getid()); /* 現在子を記憶する。*/
X; /* Xを評価する。*/
search(pop()); /* 現在子を戻す。*/
```

これを用いて例えば集合を空にするnullify()操作が定義できる。

```
delete([getsize(search(...)), search(%))
```

ここで '%' はファントムのEIDを '..' は現在要素を含む集合のEIDを示す。この操作ではまず集合の大きさを求めその後その大きさを全てで要素を削除している。集合の大きさを求めた段階で現在子が変わるため[X]を用いている。

同様にEIDがpの集合の先頭に現在位置を移動するtop(p)は次のようになる。

```
skip([-getsize(search(p)), search(p | %))
```

ここで | はパス名の結合を行う演算子である。

5. 応用例

ここでは非正規形関係モデルを表現することを考える。取り上げる非正規形関係はフィールドに複合値を許すもので[AMM83]、例として図2を示す。

Auth	Titl	Publ	Year	Kyud
Kernighan, Ritchie	The C	PH	1978	C, UNIX
Aho, Hopcroft, Ullman	Data Structure	AW	1983	ADT, Complexity

図2.NFRの例

この例では著者の順番に意味がある。またフィールドは複数の値が入るため一般に不定長である。

これらのデータベース要素をAAMでの基本要素 (集合、アトム) を用いて表現することを考えて設計した例が図3である。

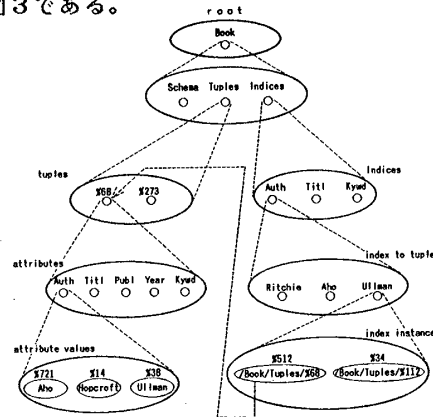


図3. NFRのAAMでの表現

関係 (Book) をまず schema、tuples、index の3つの要素からなる集合と考える。schemaは属性の集合 (図では表されていない)、tuplesはタプル集合、indicesはインデックスの集合を表す。tuplesの中の要素は各タプルを表現しており一つのタプルは各々の属性の集合 (attributes) となる。そして属性はそのフィールド内にある値 (アトム) の集合 (attribute values) である。インデックスはキーとして属性値を持つことによりタプルを直接にアクセスできるための構造である。このように表現すれば不定長のフィールドや順序も実現できる。

この構造での操作は、例えばUllmanが書いた本を探す場合、インデックスよりUllmanを著者として含むタプルを求めそのタプルのタイトルの値を得ることにより求められる。

```
getvalue(top
(getvalue(top(/Book/Indices/Auth/Ullman)) | /titl))
```

6. まとめ

ファイルをADTとしてみた高機能ファイルアクセス法の提案を行った。現在本アクセス法の初期バージョンを試作し稼働している[AOS87]。集合の順序構造についてはB+木を拡張した重み付きB+木を用い、キーに付いては拡張ハッシュ技法を用いて設計している。その結果、全ての操作の計算量は要素の数Nに対して、log(N)のオーダとなっている。

参考文献

- [AHU83] A.V.Aho, et al.:Data Structure and Algorithms, Addison-wesley, 1983
- [AMM83] H.Arisawa, K.Moriya and T.Miura: Operations and the properties on NFR relational databases, VLDB'83
- [AOS87] 有澤、岡山、鈴鹿、久保:高機能アクセス法 AAMの形式モデルとその実現方式、情報処理学会論文誌、1987
- [MAS90] K.Moriya, H.Arisawa, T.suzuka, K.Oyanagi Advanced File Organization and Its ADT Operations, to be appeared in ICSI'90