パーリ大蔵経のデータベース化による文献学的研究

— 自動読み取りシステムを用いて —

課題番号　05451007

平成６年度文部省科学研究費補助金 一般研究(Ｂ) 研究成果報告書

1995年3月

研究代表者　中谷　英明
（神戸学院大学人文学部教授）
研究分担者　江島　惠教
（東京大学文学部教授）

パーリ大蔵経のデータベース化による文献学的研究

— 自動読み取りシステムを用いて —

課題番号　05451007

平成６年度文部省科学研究費補助金 一般研究(Ｂ) 研究成果報告書

1995年3月

研究代表者　中谷　英明
（神戸学院大学人文学部教授）
研究分担者　江島　惠教
（東京大学文学部教授）

平成５年度～平成６年度 文部省科学研究費 一般研究 (B)

パーリ大蔵経のデータベース化による文献学的研究

— 自動読み取りシステムを用いて —

研究代表者：中谷英明（神戸学院大学）

研究成果報告書

目　次

はしがき

研究種目

文部省科学研究費一般研究 (B)

研究課題

パーリ大蔵経のデータベース化による文献学的研究 — 自動読み取りシステムを用いて —
（課題番号： 05451007）

研究機関名・研究期間

神戸学院大学 （研究機関番号：34509） ・ 平成５年度～平成６年度

研究組織

研究代表者： 中谷 英明 （神戸学院大学人文学部教授）
研究分担者： 江島 惠教 （東京大学文学部教授）

研究経費

平成５年度	２，７００	千円
平成６年度	９００	千円
計	３，６００	千円

研究発表

(1) 学会誌等

1. Hideaki NAKATANI, "Metre and Euphony (Sandhi) in the Nīlamata-purāṇa." *A Study of the Nīlamata. Publication of the Institute for Research in Humanities, University of Kyoto*, pp.311-354. Kyoto, March 1995.
2. 中谷英明・江島惠教,「パーリ三蔵データベースの構築と仏典研究」『パーリ学仏教文化学』８巻（印刷中）

(2) 口頭発表

1. 江島惠教・中谷英明,「パーリデータベースの作成と利用」, パーリ学仏教文化学会, 愛知学院大学, 平成６年１月７日

謝 辞

本研究は，平成３～４年度文部省科学研究費補助金 総合研究 (A)「コンピュータによる古代インド語文献の処理 — 韻律分析プログラムとデータベースの作成 —」（研究代表者 中谷英明）を継承，発展して実施された．データベースの構築は上記研究班員を中心に結成された「パーリデータベース作成会」（略称「アーブティ」）会員の献身的協力によって成し遂げられたものであることをここに銘記する．

またコンピュータプログラム作成，ソフトウエアー等に関する情報提供に労苦を惜しまず協力下さった高島淳氏（東京外国語大学アジアアフリカ言語文化研究所助教授）に甚深の謝意を捧げる．

平成 6 年度文部省科学研究費補助金 一般研究 (B) 研究成果報告書

パーリ大蔵経のデータベース化による文献学的研究

— 自動読み取りシステムを用いて —

(課題番号 05451007)

平成 7 年 3 月 25 日

研究代表者　中谷 英明

序

本研究の意義 — テキストデータベースと人文科学

コンピュータの普及と各種データベースの整備は，人文科学に大きな変化をもたらし始めている．インド学・仏教学においても，例えば日本印度学仏教学会作成の論文データベース「汎」は，論文名，著者名，年代，キーワード等から論文を瞬時に検出する事を可能にした．

このように論文データベースの利便は大きいが，テキストデータベースはそれに加えて，研究に質的変化をもたらしつつある．単なる利便に留まらず，後述のごとく研究上の新視野を開く点に，テキストデータベースの重要さが存在するのである．

最近 1 年間に於ける大型テキストデータベースの公開は，学界に大きなインパクトを与えた．例えば，タイ Mahidol 大学作成のタイ版パーリ三蔵データベースが CD-ROM として発売され，また徳永宗雄京都大学教授作成になる Mahābhārata, Rāmāyaṇa が公開された．他方，日本印度学仏教学会でも「テキストデータベース作成検討委員会」が発足し，『大正新脩大蔵経』のデータベース化に向けて，具体的作業が開始されている．

このような折，パーリ大蔵経データベースの構築と，その利用法の研究は，わが国における新しいインド学・仏教学の創造のために，極めて有意義と考えられる．

本研究の概要 — パーリ大蔵経データベース構築と研究方法の確立

ここに，平成 5 年度から 6 年度まで，表記の研究課題の下に行った研究を報告する．

研究は大きく 2 部に分かれる．すなわち，(1) パーリ大蔵経データベースの構築，(2) パーリ大蔵経データベースを用いた研究方法の確立，である．

(1) パーリ大蔵経データベース構築は，1) 底本として Pali Text Society 出版本を用い，光学読み取り装置により電子データとした後，2) 誤読修正プログラムにより修正，3) こうしてできた素データを研究者が分担して校正する，という手順をとった．校正分担のため，平成 5 年春にパーリテキストデータベース作成会「アーブティ」(APTI = Association for Pali Text Inputting; 代表者 江島恵教) を組織し，会員は現在 36 名を数える．作業上必要な情報は，4 種のマニュアルに纏めた．なお「アーブティ」は，Pali Text Society (Chairman: Prof. K. R. Norman) より，PTS 本をデータベース底本とすることに関する認可を得たことを付言する.[1]

(2) パーリ大蔵経データベース構築中に開発された「索引作成」,「音韻分析」などのプログラムは，研究にも活用可能である．別途に作成された「韻律分析」プログラムも，仏典研究の新視野を開きつつある．本報告の後半部においては，これらの新しい「道具」が可能とした事例を報告する．コンピュータによる研究には，多くの可能性がなお潜在し，ここに報告することはその極く一端に過ぎないであろう．しかし既に，従来の文献学的手法による研究に比して，幾分かの質的変化が現れているのではなかろうか．

[1] K. R. Norman 教授よりの江島恵教アーブティ代表宛 1994 年 4 月 19 日付書簡による．

本報告は次の各項からなっている.

1. テキストデータベース構築
- **(1) テキストデータベースの構築方法**
- **(2) OCR によるデータベース構築手順**

2. 「アープティ」テキストデータベースの現況
アープティテキストファイル校正者一覧

3. データベース活用プログラムと仏典研究
- **(1) 検索プログラム**
- **(2) 韻律分析プログラム**
- **(3) 索引作成プログラム**
- **(4) 作表プログラム**

本報告末尾には，付録として次のものを添える.

(1) マニュアル
a. APTI への誘い　b. RECOGNITA PLUS 使用法　c. OCR データ処理手順
d. インド学研究とコンピュータ利用（初級編）
(2) Sanskrit 転写法
a. KH 方式　b. OCR 方式（光学読み取りソフト Recognita Plus 用）

1. テキストデータベースの構築

1. テキストデータベースの構築方法

構築方法には，大別して 2 種が考えられる.

1) 手入力.
2) 光学読み取り装置 (Optical Character Recognition (=OCR) System) による入力.

両方法ともそれぞれ異なった利点があり，一概に優劣は論じられない. 手入力によれば，入力者は読みつつ入力するから，内容をチェックできるが，読み飛ばしなどの事故を避けることが比較的困難である. OCR による入力は，システム構築に多大の時間を要するが，システムが出来上がれば，ほぼ遺漏なく入力できる. 従って大量の同質データ入力には後者が適していると言えよう.[2]

Pali Text Society 本を底本として採用した我々がとったのは，後者の方式である.

2. OCR によるデータベース構築手順

(1) OCR 入力

a. OCR ソフトの選択：

光学読み取りソフトウエアーの方式には，

1) 1 区画内をドットに分割して全体として判読する方式
2) 字線の連結形態から判読する方式

の 2 種がある. 前者には大型の機械装置を要し，後者が 100 万円以下でソフトを含む全装置を整えることができるのに対し，10 倍近い費用を要する. 我々は後者を採用した.

- ソフトウエアー： Recognita Plus (Ver. 1.02, 後に Ver. 2.00)
- ハードウエアー： コンピュータ：IBM コンパティブル PC
  スキャナー：NEC 製 (300DPI 及び 400DPI)

[2] ただし手入力による場合も，弁別記号付き文字を 1 タッチで入力できるキーに割り当てて入力すれば，かなり速く正確な入力が可能となる. 習熟すれば 1 時間にシュローカ 400 詩以上を入力できるという.

Recognita Plus はハンガリーのブダペスト大学が開発したアルファベット読み取りソフトで，精度，速度の両面において優れた性能を備えている．

b. 文字セット選択・学習・入力

Recognita Plus において，(1) 文字セット ROMAN8 を基本文字に設定し，(2) 不用な文字をセットから削除した後，(3) コピーの要領でまず試行読み取りを行う．PTS 本は東大，京大両大学に蔵される初版本を用いたが，それでも印字不良の頁が少なくなく，たとえば a は，右上部，左下部などでかすれていると，s や e に読み誤られることがあった．そこで，(5) 適当な読み取り明度を決めた後，(4) かすれた字形を登録する（「学習させる」と言う）．この明度決定と登録の仕方が読み取りの成否を左右する．Ver.2.00 ならば，標準状態の印字の場合，99.5 パーセント以上の精度を達成できた．詳しくは『RECOGNITA PLUS 使用法』（「アーブティマニュアル 1」）参照．ここに例として Vinaya Piṭaka の冒頭部を掲げる．

テキスト 1 — OCR 原データ

```
        VINAYAPITAKAM.
             Mg H 「 Yg G Gg.
   Namo tassa bhagavato arahato samm タ sambuddhaasa.
                    I.
Te ヌ a samayena buddho bhagav タ U r u v el タ y a m vih ■ ati
najj タ Nera キ jar タ ya t ム re bodhirukkham テ le paFham タ bhisam-
buddho. atha kho bhagav タ bodhirukkham ■ e satt タ ham eka-
palla キ kena nis?di vimuttisukhapatisamved ム. II 1 II atha kho
bhagav タ rattiy タ pathamam y ■ am paFiccasamupp タ dam
```

(2) SED による処理

a. 2 バイトコード文字等をアルファベットへ

Recognita Plus の文字セット Roman 8 における â, ñ は JIS コードでは各々タ，キという表示になる．また例えば文字列 āk は印字上連結していて 1 文字と見なされ，「棄」という 1 個の 2 バイトコード文字として読み取られることがある．これらを元の â, ñ, āk 等に変換する．これには Stream Editor (=SED) を用いた．SED は予め一つのファイルに書き入れておいた命令を，テキストデータの各行に対し次々に実行する．例えば上記 3 種の置換は次の 3 命令行によって実行される．

```
s/タ/aa/g
s/キ/~/g
s/棄/aak/g
```

s/棄/aak/g は 'Substitute aak for 棄 globally.' の意味で，これによってテキスト中の全ての「棄」は 'aak'（=āk）に置換される．言うまでもなく，行初の s は 'substitute', 行末の g は 'globally'（1 行につき最初の 1 回だけ置換を実行するのではなく，全ての該当箇所において実行すること）を意味する．このような置換命令行を次々に作って一つのファイルとし，命令集ファイル（Script File と呼ぶ）を作る．このファイルは最終的には，310 余の命令行を収める 4 ファイルとなった．

ただしこの段階では，例えば ā を aa と転写している．この転写方式（OCR 方式と呼ぶ）については本報告末尾付録参照．

b. 残存 2 バイトコード文字の摘出と置換

下記のスクリプトによって，なお変換されずに残る 2 バイトコード文字を検出した後，それらの置換命令行を付加し，全文字をアルファベットに変えた．

```
/[^ -~]/{
```

```
s/^/¥f ¥1: /p
}
```

c. 文頭大文字を小文字に変換

PTS 本では文頭および固有名詞語頭が大文字で表記されているので，これを SED の y コマンドを用いて小文字に変換した．これは単語一覧等に必要な処置である．

```
y/ABCDEGHIJKLMNOPRSTUVY/abcdeghijklmnoprstuvy/
```

d. KH 方式への変換

次に置換コマンドによってテキストを KH 方式に変換する．KH 方式は，サンスクリットの全文字が各々アルファベット 1 文字に対応するよう考案された文字セットであり，記憶の便をはかって 2 原則に従っている．すなわち (1) 通用のローマ字転写の下点付き文字（母音の ṛ を含む）および長母音は，その大文字（例えば ṭ は T, ā は A など），(2) 喉音と口蓋の両鼻音は，各系列の有声破裂音の大文字（ṅ, ñ は G, J）を用いる．以上の 2 原則以外，特に記憶しなければならないのは，ś が z, 長母音の ṝ が q, 母音の ḷ が W 表記となるという 3 点である．その全文字表は本報告末尾付録参照.[3]

こうして上記テキスト 1 は次のように変換される．なおこの段階で，巻数・頁数 (^1_1 など) が自動的に挿入される．

テキスト 2 — KH 方式テキスト

```
^1_1
     vinayapitakaM.
          mg h A yg g gg.
  namo tassa bhagavato arahato sammAsambuddhAsa.
                i.
teUa samayena buddho bhagavA u r u v el A y a M vih|rati
najjA neraJjarAya tIre bodhirukkhamUle paThamAbhisam-
buddho. atha kho bhagavA bodhirukkham|le sattAhaM eka-
pallaGkena nis?di vimuttisukhapatisaMvedI. I 1 I atha kho
bhagavA rattiyA pathamaM y|MaM paTiccasamuppAdaM
```

e. 弁別記号付き文字，誤読文字の校正

文字下部の弁別記号は，Recognita Plus にとって判読が困難である．例えば上記テキストの pitaka は piṭaka の誤読である．また，vih|rati は a がかすれて読めなかったようである．これら誤読は同種のものが多発するから，SED によって一括変換することができる．

```
s/pitak¥([aoeA]¥)/piTak¥1/g
s/vih|rati/viharati/g
```

この種の修正スクリプト行は現在約 1,500 行に達している．これによる修正後，テキストは以下の状態になる．

テキスト 3 — 校正用テキスト

```
^1_1
```

[3] KH とは Kyoto-Harvard の略．1989 年 Harvard 大学 Michael Witzel 教授が滞洛中に京都大学インド学関係者と共に検討して出来上がったもの．中谷が『印度学仏教学研究』第 39 巻 2 号 pp.825-823.（「コンピューター利用に関するパネル — ウィーンにおける第 8 回世界サンスクリット会議消息 —」東京・1991 年）に紹介したものは当初案であるが，その後頻度を勘案して，母音と子音の ḷ の記号を交換した．

```
vinayapiTakaM.
mghA yggg.
namo tassa bhagavato arahato sammAsambuddhAsa.
i.
teUa samayena buddho bhagavA uruvelAyaM viharati
najjA neraJjarAya tIre bodhirukkhamUle pathamAbhisam-
buddho. atha kho bhagavA bodhirukkhamUle sattAhaM eka-
pallaGkena nisIdi vimuttisukhapatisaMvedI. /1/ atha kho
bhagavA rattiyA pathamaM yAmaM paTiccasamuppAdaM
```

ちなみにこれを通用転写に変換すれば以下のとおり.

```
vinayapiṭakaṃ.
mghā yggg.
namo tassa bhagavato arahato sammāsambuddhāsa.
i.
teūa samayena buddho bhagavā uruvelāyaṃ viharati
najjā nerañjarāya tīre bodhirukkhamūle pathamābhisam-
buddho. atha kho bhagavā bodhirukkhamūle sattāhaṃ eka-
pallaṅkena nisīdi vimuttisukhapatisaṃvedī. /1/ atha kho
bhagavā rattiyā pathamaṃ yāmaṃ paṭiccasamuppādaṃ
```

3. 校正

これを見ればなお誤読が残存することがわかる. これらは一々訂正してゆくことになる.「アーブティ」では 1 人が PTS 本 300 頁相当を担当して校正を行った.

4. 単語リストの作成

校正に資するため, また次に記述する音韻分析プログラムに用いるために, 全単語リスト（出現回数付き）を作った. ただしここに「単語」と呼ぶのは, 単にスペースあるいは改行によって区切られる文字列を指し, 複合語等を文分に分かったものではない.

アスキー社から出ているプログラムと sortf.exe を次のように組み合わせた：[4]

```
sed -f fphap.sed %1 | word | sortf | uniq -c | sortf -nr > %2.wlc
```

例えば Vinaya 第 1 巻の単語リストは次のとおり.

表 1 Vinaya 第 1 巻単語リスト（左列数字は出現度数）

```
2296 ti
1997 kho
1239 na
1141 bhikkhave
 920 ca
 805 atha
 767 vA
```

[4]ただし fphap.sed は, タイトル行などを取り除くのみ. 次の 2 行からなる：

```
/^*/d
s/'/ /g
```

```
714 bhikkhU
676 hoti
668 pana
612 pi
573 bhagavA
505 tena
497 evaM
  .
  .
  .
```

出現度数が1回のものには誤読の語が多く，この単語リストを2次校正の手がかりとした．

5. 検査プログラム PHAP

上記の単語リストを Phonetic Analysis Program (= PHAP) にかけて，パーリ語には有り得ない，あるいは希少である音群を摘出し，なお残っている誤読の発見につとめた．プログラムは例えば次のようなスクリプト41種から成る．

```
/[kcTtp][gjDdb]/{
s/¥([kcTtp][gjDdb]¥)¥([h]*¥)/__¥1¥2__/g
p
}
d
```

kg, kj, ... cg, cj,... などの綴（無声破裂音＋有声破裂音）はパーリ語には含まれないから，検出されたものは誤綴である．またパーリでは，母音連続や，子音群の前の長母音も原則として回避されるから，この種のものは吟味しなければならない．後者の摘出プログラムは以下のとおり．

```
/[AIU][bcdghjklmnprstvyGJTDNmMSL][bcdgjklmnprstvyGJTDNmMSL]/{
s/[AIU][bcdghjklmnprstvyGJTDNmMSL][bcdgjklmnprstvyGJTDNmMSL]h*/__&__/g
p
}
d
```

特定語彙にのみ許容される綴は，その語彙を除いて摘出するよう，許容語彙（文字列）の一覧を作った．

6. スペルチェッカーによる検査

上の検査プログラム PHAP の他，英語用スペルチェッカー MicroSPELL 1.0J の辞書をパーリ語に置き換えたスペルチェッカーを作った．辞書は現在約6万語から成っている．チェックが極めて速く（1メガバイトのテキストを数十秒で終える）高い有効性が確認された．ただし今後，辞書の整備が必要である．

2. 「アーブティ」テキストデータベースの現況

上記経緯により成立したパーリテキストデータベース作成会「アーブティ」（江島恵教代表）は，現在，Khuddaka Nikāya を除く 4 Nikāya の校正を9割方終了し，Khuddaka Nikāya についても9ファイルを校了している．Vinaya および 4 Nikāya に関しては，1995年5月末に1次校正を完了する予定である．アーブティの作成ファイルと校正担当者は次のとおりである．なお校正要領とソフトウエアーに関するマニュアルは付録参照のこと．

アーブティテキストファイル校正者一覧

平成 7 年 2 月 20 日現在

(テキスト名略号は Critical Pali Dictionary による)

ファイル名 (テキスト名・巻・頁)	担当者	1 次校了	2 次校了	3 次校了
1. Vinaya Piṭaka.				
VIN1 (Vin I pp. 1 - 300)	箕浦　暁雄			
VIN2 (Vin I p. 301 - II p. 240)	李　慈郎			
VIN3 (Vin II p. 241 - III p. 232)	朴　鍵			
VIN4 (Vin III p. 233 - IV p. 266)	松田　慎也			
VIN5 (Vin IV p. 267 - V p. 196)	羽矢　辰夫			
VIN6 (Vin V pp. 197 - p. 226)	鈴木　隆泰			
2. Suttanta Piṭaka.				
Dīgha Nikāya				
DN1 (DN I p. 1 - II p. 47)	乙川　文英	○	○	
DN2 (DN II pp. 48 - 347)	赤松　明彦	○		
DN3 (DN II p. 348 - III p. 293)	佐藤直実/赤松明彦			
Majjhima Nikāya				
MN1 (MN I pp. 1 - 300)	梶原　三恵子	○		
MN2 (MN I p. 301 - II p. 76)	榎本　文雄			
MN3 (MN II pp. 77 - III p. 110)	宮下晴輝/(船山徹)			
MN4 (MN III pp. 111 - 302)	松田　祐子			
Saṃyutta Nikāya				
SN1 (SN I p. 1 - II p. 60)	計良　龍成	○		
SN2 (SN II p. 61 - III p. 74)	佐久間　秀範			
SN3 (SN III p. 75 - IV p. 95)	苫米地　等流	○		
SN4 (SN IV pp. 96 - 395)	生井　智紹			
SN5 (SN IV p. 396 - V p. 288)	山極　伸之	○		
SN6 (SN V pp. 289 - 478)	引田　弘道			
Aṅguttara Nikāya				
AN1 (AN I pp. 1 - 304)	桂　紹隆	○		
AN2 (AN II p. 1 - III p. 34)	宇野　智行	○		
AN3 (AN III pp. 35 - 334)	石上　和敬	○		
AN4 (AN III p. 335 - IV p. 182)	種村　隆元	○		
AN5 (AN IV p. 183 - V p. 16)	宮崎　泉	○	○	
AN 6 (AN V pp. 17 - 316)	室寺　義仁			
AN 7 (AN V pp. 317 - 361)	宮下　晴輝			

ファイル名(テキスト名・巻・頁)	担当者	1次校了	2次校了	3次校了
Khuddaka Nikāya				
DHP (Dhp)	中谷　英明	○		
UD (Ud)	金　奠益	○		
IT (It)	引田　弘道	○		
SNP (Sn)	中谷　英明	○	○	
THTHI (Th, Thi)	中谷　英明			
JA1 (Ja I pp. 1 - 330)	谷川　泰教			
JA2 (Ja I p. 331 - II p. 136)				
JA3 (Ja II p. 137 - III p. 16)	安藤　充			
JA4 (Ja III p. 17 - 346)				
JA5 (Ja III p. 347 - IV p. 134)				
JA6 (Ja IV pp. 135 - 464)				
JA7 (Ja IV pp. 465 - V p. 295)				
JA8 (Ja V p. 296 - VI p. 115)	横地　優子			
JA9 (Ja VI pp. 116 - 445)	入山　淳子	○		
JA10 (Ja VI pp. 446 - 596)	松田　祐子			
CN (Nidd II pp. 1 - 287)	鈴木　隆泰			
CP (Cp pp. 1 - 38)	斎藤　明			
PET1 (Peṭ pp. 1 - 206)	金　奠益	○		
PET2 (Peṭ pp. 207 - 260)				
VM1 (Vm pp. 1 - 300)	金　宰晟	○	○	○
VM2 (Vm pp. 301 - 600)	金　宰晟	○	○	○
VM3 (Vm pp. 601 - 713)	金　宰晟	○	○	○
3. Abhidhamma Piṭaka.				
DHS (Dhs pp. 1 - 263)	斎藤　明			

上表の担当者名空欄は校正未着手ファイル．その外，次のテキストも校正担当者未定である．

Suttanta Piṭaka.

Khuddakapātha, Mahāniddesa, Patisambhidāmagga, Buddhavaṃsa, Milindapañha, Nettippakaraṇa.

Abhidhamma Piṭaka.

Vibhaṅga.

Aṭṭhakathā

Paramatthajotikā II, Atthasālinī.

3. データベース活用プログラムと仏典研究

1. 検索プログラム

(1) データの前処理

種々あるテキストデータベース活用法の中で，ある語形（文字列）を検出して，それを含む文章を一覧表にすることが，最も基本的かつ用途の広い使い方であろう．

このためにはPTS本と全同のデータを，予め次のように整形しておく必要がある．

a. 散文データの場合：

1) 2行に跨っており，分割されている1語の前後部を，途中の改行および行末ハイフンを消去して一つに結合する (m21.sed).

2) ピリオドを改行に変換し，1文を1行とする (pdlc.sed).

3) 各行頭に巻数・頁数を入れる (pgin1.sed; pgin2.sed).

このPTS形式データの，巻・頁数入りデータへのSEDによる整形プログラムPAGEINは，次のとおり．

1. m21.sed

```
:loop
$p
N
s/[※-L][ ]*※n *※([A-Za-z.,:;|]+※)/※1※n/
P
D
b loop
```

2. pdlc.sed

```
s/※.※.+pe※.※.+/※. /g
s/※.※.+/※. /g
s/※([^0-9IVX]※)※([.:;]※)※([ "]※)/※1※2※3※n/g
```

3. pgin1.sed[5]

```
/※^/{
:loop
N
s/※^※([0-9_. ]+※)※n※([^※^]*※)$/※2※n^※1/
P
D
$b
}
t loop
P
D
b loop
```

[5] pgin1.sed は左側頁，pgin2.sed は右側頁用である．

4. pgin2.sed

```
:loop
N
s/¥([^¥^]+¥)¥n^¥([0-9_. ]+¥).*/¥1 #¥2:/
$b
t loop
P
D
b loop
```

b. 韻文データの場合：

例えば Suttanipāta はこのような形でデータとなっている.

1. yo uppatitaM vineti kodhaM
   visataM sappavisaM va osadhehi
   so bhikkhu jahAti ora-pAraM
   urago jiNNaM iva tacaM purANaM
2. yo rAgam udacchidA asesaM
   bhisa-pupphaM va saro-ruhaM vigayha
   so bhikkhu jahAti ora-pAraM
   urago jiNNaM iva tacaM purANaM
   ...

この 1 詩を 1 行に整形する (joint.sed による).

5. joint.sed

```
:loop
$p
N
s/¥n  +/| /
t loop
P
s/^.*¥n//
b loop
```

(2) 検索ソフト YGREP の使用

このように整形したデータを，例えばフリー・ソフトウエアーである YGREP.EXE によって検索すれば次のようなデータが得られる.

たとえば現在約 30 メガバイトに達する「アーブティ」データベースから，samāhita という語を含む行（ここに「行」とは，上記のように整形した 1 文を指す）を摘出するには，MS-DOS のプロンプト（命令待ち）状態において次のように入力する.

```
ygrep samAhit *.pg* > samahita.pal
```

約 30 秒 (32 bit cpu unit 使用) でデータ・ファイル（ここでは SAMAHITA.PAL）が出来る．現在の APTI データ中には 'samAhit' を含む行が 571 件検出されるが，Dīgha Nikāya と Suttanipāta の冒頭部データをここに掲げる.

```
-- a:¥pali¥pgi¥dn1.pg --
#1_13: 31. 'idha bhikkhave ekacco samaNo vA brAhmaNo vA Atappam anvAya padhAnam
anvAya anuyogam anvAya appamAdam anvAya sammA-manasikAram anvAya tathArUpaM  cet
o-samAdhiM phusati yathA __samAhit__e citte2 anekavihitaM  pubbe nivAsaM anussar
ati
#1_14: tam kissa hetu? ahaM hi Atappam anvAya padhAnam anvAya anuyogam anvAya ap
pamAdam anvAya sammA manasikAram anvAya tathA-rUpaM ceto-samAdhiM phusAmi yathA
__samAhit__e citte aneka-vihitaM pubbe nivAsaM anussarAmi __ seyyathIdaM ekam pi
 jAtiM .
. . .

-- a:¥pali¥pgi¥snp.pg --
174. sabbadA sIla-sampanno | paJJavA su__samAhit__o | ajjhatta-cintI satimA | og
haM tarati duttaraM
212. paJJAbalaM sIlavatUpapannaM | __samAhit__aM jhAnaratAM satImaM | saGgA pamu
ttaM akhilaM anAsavaM | taM va^api dhIrA muniM vedayanti
214. yo ogahane thambo-r-iva^abhijAyati | yasimiM pare vAcA pariyantaM vadanti |
taM vIta-rAgaM su__samAhit__indiryaM | taM va^api dhIrA muniM vedayanti
. . .
```

なおこれ以外にも，YGREP によれば，該当行の前後 1 行を同時にリスト・アップしたり，2 個以上の文字列が共に現れる文の検索など，種々の検索が可能である.[6]

2. 韻律分析プログラム

膨大なテキストからの文字列検索とならんで，コンピュータが最も能力を発揮するのは，詩節の韻律一覧の作成である．韻律研究としては従来 H. Smith の Suttanipāta, K.R. Norman の Thera-therīgāthā 研究など，数点の極めて優れた研究があったが，コンピュータによって検証すれば，なお不十分な点のあることがわかる．統計結果検討と統計方法（すなわち韻律形式の立て方）修正とを繰り返すことによってテキストに固有の韻律を明かにすることができるのである．ここでは詳細な報告は差し控え，高島淳氏（東京外国語大学アジア・アフリカ言語文化研究所）と中谷が共同作成した韻律分析プログラムの一端を紹介する．

(1) 韻律一覧作成

PERL という簡易言語によるこのプログラム（ADD_MT.PL）は，韻律形式一覧（別ファイル内蔵）を参照して一覧表を作成する.[7] 例として Suttanipāta 冒頭部 10 詩の韻律データを表 2 に掲げる．

Suttanipāta 冒頭の Uragasutta17 詩は，Aupacchandasaka という Mātrā 律である.[8] プログラムは各行を起部 (opening) と結部 (cadence) の 2 部に分かち，短音節を ∪，長音節を − で表記している．Odd Aupa. あるいは Even Aupa. は，Aupacchandasaka の奇数行あるいは偶数行を示す．10 s. 等は音節数 (10 syllables) である．

(2) パターン名称

表 2 の各行末に記される 3 文字（35P など）は，各行を先頭から 4 音節ごとに区切り，残りを 3，2，または 1 音節とした時の韻律パターンを表している．パターン名称は，4 音節パターンは 16 進法，3 音節以下は 16 進法の続きのアルファベットを当てた．ただし O は印字したとき数字の 0 との区別がつき難いの

[6] たとえば samāhita と appamāda が同時に現れる文章の検索は次のとおり：
ygrep -a samAhit:appamAd *.pg* > smhtpmd.pal

[7] プログラム名は ADD_MT.PL．韻律形式辞書は MTDIC.PAG, MTDIC.DIR．詳細は省略．

[8] ただし 7a は Vaitālīya である．

表 2 Suttanipāta 1–10 の韻律

1a	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
1b	∪ ∪ – – ∪ ∪	– ∪ – ∪ – ∪	Even Aupa.	12 s.	CD5
1c	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
1d	∪ ∪ – – –	∪ ∪ ∪ – ∪ – –	Even Aupa.(cad.int.rs.)	12 s.	C74
2a	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
2b	∪ ∪ – – ∪ ∪	– ∪ – ∪ – ∪	Even Aupa.	12 s.	CD5
2c	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
2d	∪ ∪ – – –	∪ ∪ ∪ – ∪ – –	Even Aupa.(cad.int.rs.)	12 s.	C74
3a	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
3b	∪ ∪ – – ∪ ∪	– ∪ – ∪ – –	Even Aupa.	12 s.	CD4
3c	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
3d	∪ ∪ – – –	∪ ∪ ∪ – ∪ – –	Even Aupa.(cad.int.rs.)	12 s.	C74
4a	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
4b	∪ ∪ – – ∪ ∪	– ∪ – ∪ – –	Even Aupa.	12 s.	CD4
4c	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
4d	∪ ∪ – – –	∪ ∪ ∪ – ∪ – –	Even Aupa.(cad.int.rs.)	12 s.	C74
5a	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
5b	∪ ∪ – – –	∪ ∪ ∪ – ∪ – ∪	Even Aupa.(cad.int.rs.)	12 s.	CF5
5c	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
5d	∪ ∪ – – –	∪ ∪ ∪ – ∪ – –	Even Aupa.(cad.int.rs.)	12 s.	C74
6a	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
6b	∪ ∪ ∪ – ∪ ∪	– ∪ – ∪ – –	?	12 s.	ED4
6c	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
6d	∪ ∪ – – –	∪ ∪ ∪ – ∪ – –	Even Aupa.(cad.int.rs.)	12 s.	C74
7a	– ∪ ∪ –	– ∪ – ∪ –	Odd Vait.	9 s.	65T
7b	– – – ∪ ∪	– ∪ – ∪ – –	Even Aupa.	11 s.	1AK
7c	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
7d	∪ ∪ – – –	∪ ∪ ∪ – ∪ – –	Even Aupa.(cad.int.rs.)	12 s.	C74
8a	– – ∪ –	– ∪ – ∪ – –	?	10 s.	25P
8b	– – – ∪ ∪	– ∪ – ∪ – –	Even Aupa.	11 s.	1AK
8c	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
8d	∪ ∪ – – –	∪ ∪ ∪ – ∪ – –	Even Aupa.(cad.int.rs.)	12 s.	C74
9a	– – ∪ –	– ∪ – ∪ – –	?	10 s.	25P
9b	– – ∪ ∪ ∪ ∪	– ∪ – – – –	? Even Aupa.	12 s.	3D0
9c	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
9d	∪ ∪ – – –	∪ ∪ ∪ – ∪ – –	Even Aupa.(cad.int.rs.)	12 s.	C74
10a	– – ∪ –	– ∪ – ∪ – –	?	10 s.	25P
10b	– – ∪ ∪ ∪ ∪	– ∪ – ∪ – –	Even Aupa.	12 s.	3D4
10c	– – ∪ ∪	– ∪ – ∪ – –	Odd Aupa.	10 s.	35P
10d	∪ ∪ – – –	∪ ∪ ∪ – ∪ – –	Even Aupa.(cad.int.rs.)	12 s.	C74

で除外した（Oの次のPが2音節パターンの先頭にあたり記憶に便利である）．パターン名は次のとおり．

表3 パターン名一覧

4音節		3音節	2音節	1音節
−−−−= 0	∪−−−= 8	−−−= G	−−= P	−= T
−−−∪= 1	∪−−∪= 9	−−∪= H	−∪= Q	∪= U
−−∪−= 2	∪−∪−= A	−∪−= I	∪−= R	
−−∪∪= 3	∪−∪∪= B	−∪∪= J	∪∪= S	
−∪−−= 4	∪∪−−= C	∪−−= K		
−∪−∪= 5	∪∪−∪= D	∪−∪= L		
−∪∪−= 6	∪∪∪−= E	∪∪−= M		
−∪∪∪= 7	∪∪∪∪= F	∪∪∪= N		

インドの伝統的韻律パターン名称は，周知のように3音節を1単位として ya (∪−−), repha (−∪−), ta (−−∪) などとされる．これはインド古典韻律には，4音節を基本単位とし，その先頭または末尾の1音節を任意とするものが多く（シュローカはその典型である），多くの場合，3音節パターンを指示すれば足りるからである．このインド人韻律家の洞察を尊重して，我々も4音節を韻律パターンの基本単位とする事にした．ただし名称は3音節ではなく，4音節に与えた．

このような名称は，統計プログラムに資するだけでなく，例えば表2を一瞥すればただちに，C74というパターンの多さに気付くであろう．~~この韻律形は従来看過されてきたが，Mātrā 律の結部先頭音節の融解 (resolution) 現象と解されるもので，Suttanipāta では頻用されている．~~ Suttanipāta の韻律の，全体的な高度の規則性をこのプログラムによって確認したが，詳細な報告は別稿に譲る．

(3) 統計プログラム

テキスト各章のŚloka, Triṣṭubh, Mātrācchandas, Āryā などの詩節数を合計して示すプログラムが高島淳氏によって作成されている．[9]

韻律分析プログラムは，韻律研究と平行して開発するため，なお完成していないが，完成の折には，その結果を詳細に統計・表示するプログラムを作成する予定である．

3. 索引作成プログラム

全単語索引（ただしここに単語とは，先述のとおり，スペースによって区切られる文字列）の作成プログラムは，ここに詳細を示すように，SED, PERL を使って簡単に作成できる．自作のプログラムの利点は，遺漏がないことを自ら確認し，また細部の変更を自由にし得ることである．

なおこの索引ファイルを利用して，次項に説明する「語形出現箇所一覧表」を直接作ることができる．[10]

(1) 散文データの場合：

先に 1. 検索プログラム の項下に説明した (1) 改行・行末ハイフン除去，(2) ピリオドの改行への変換を済ませた（すなわち m21.sed と pdlc.sed をかけた）テキストから出発して，次の手順で索引を作る．

1. 1wrd.sed ..... 1行に1単語とする．

```
s/["?!¥-]/ /g
s/ +/ /g
s/¥([^ ]+¥) +/¥1¥n/g
```

[9] プログラム名は MT_COUNT.PL. 詳細略．

[10] なお高島淳氏は語末から検索する，いわゆる逆引き索引のためのプログラムも作成された (rev_srt.pl).

2. **cl.sed** ..... 不要文字，タイトル行，空行を除去する.

```
s/^ +//
s/^芋.*芋([a-zA-Z']+芋)[.,:;!?      ]*$/芋1/
s/^[0-9CLVXI.,;:$ 芋[芋]]+$//
/#/d
/^$/d
```

3. **pnumb.sed** ..... 各行に頁番号を振る.

```
/芋^/{
:loop
N
s/芋^芋([0-9_.]+芋)芋n芋([^芋^]+芋)$/芋2芋n^芋1/
P
D
$b
}
t loop
P
D
b loop
```

4. **ppost.sed** ..... 頁番号を行末に移動.

```
:loop
N
s/芋([a-zA-Z']+芋)芋n^芋([0-9_]+芋)芋.*/芋1 #芋2,/
$b
t loop
P
D
b loop
```

5. **cl2.sed** ..... 再び不要行の削除.

```
/芋^/d
/^[':;.,]$/d
```

6. **SSS29.EXE** ..... サンスクリット順にソートする.[11]

7. **collapse.pl** ..... 同一語の頁数字を 1 行に纏める.[12]

```
line: while (<>) {
        chop;
        ($word,$num) = split(/芋t/);
        while ($word ne $prev) {
              if ($. > 1) {print "芋n";}
             $prev = $word;
              print $word,"芋t",$num;
              next line;
```

[11] SSS29.EXE は中園・金沢両氏作.
[12] collapse.pl は高島淳氏作.

```
        }
        {
         print " ",$num;
        }
    }
    {
         if ($. > 1) {print "¥n";}
    }
```

8. refrm.sed ..... 不要文字削除，行末ピリオド挿入などの整形.

```
/^ * ¥t*#/d
s/#//g
s/^ *¥([a-zA-Z.,']+¥)¥t* *¥([0-9]¥)/¥1       ¥2/
s/,$/./
```

(2) 韻文データの場合：

散文と同様，検索プログラムの整形を終えた (joint.sed をかけた) データから出発する.

9. abcd.sed ..... 各行に a, b, c など行名を付加.

```
/^#/d
s/^[0-9MCLVXImclvxi.,;:        /]+$//
s/¥^/ /g
s/^¥([0-9A-H]+¥.¥)¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥
(.*¥) *| ¥(.*¥) *| ¥(.*¥)[.,:; ]*$/¥2 ¥/¥1a,¥
¥3 ¥/¥1b,¥
¥4 ¥/¥1c,¥
¥5 ¥/¥1d,¥
¥6 ¥/¥1e,¥
¥7 ¥/¥1f,¥
¥8 ¥/¥1g,¥
¥9 ¥/¥1h,/
s/^¥([0-9A-H]+¥.¥)¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥
(.*¥) *| ¥(.*¥)[.,:; ]*$/¥2 ¥/¥1a,¥
¥3 ¥/¥1b,¥
¥4 ¥/¥1c,¥
¥5 ¥/¥1d,¥
¥6 ¥/¥1e,¥
¥7 ¥/¥1f,¥
¥8 ¥/¥1g,/
s/^¥([0-9A-H]+¥.¥)¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥
(.*¥)[.,:; ]*$/¥2 ¥/¥1a|¥
¥3 ¥/¥1b|¥
¥4 ¥/¥1c|¥
¥5 ¥/¥1d|¥
¥6 ¥/¥1e|¥
¥7 ¥/¥1f|/
```

```
s/^¥([0-9A-H]+¥.¥)¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥(.*¥)[.,:; ]*$/
¥2 ¥/¥1a|¥
¥3 ¥/¥1b|¥
¥4 ¥/¥1c|¥
¥5 ¥/¥1d|¥
¥6 ¥/¥1e|/
s/^¥([0-9A-H]+¥.¥)¥(.*¥) *| ¥(.*¥) *| ¥(.*¥) *| ¥(.*¥)[.,:; ]*$/¥2 ¥
/¥1a|¥
¥3 ¥/¥1b|¥
¥4 ¥/¥1c|¥
¥5 ¥/¥1d|/
s/^¥([0-9A-H]+¥.¥)¥(.*¥) *| ¥(.*¥) *| ¥(.*¥)[.,:; ]*$/¥2 ¥/¥1a|¥
¥3 ¥/¥1b|¥
¥4 ¥/¥1c|/
s/^¥([0-9A-H]+¥.¥)¥(.*¥) *| ¥(.*¥)[.,:; ]*$/¥2 ¥/¥1a|¥
¥3 ¥/¥1b|/
s/^¥([0-9A-H]+¥.¥)¥(.*¥) *¥(.*¥)[.,:; ]*$/¥2 ¥/¥1a|/
```

10. vnumb.sed ..... 各単語に詩節・行番号を振る.

```
s/      / /g
s/¥//#/
s/-/ /g
:loop
s/¥([^ #][^ #]*¥) ¥(.*¥)¥(#.*¥)$/¥1¥3¥
¥2¥3/
p
s/^.*¥n//
t loop
s/#.*//
s/^ *¥**//
```

11. clt.sed ..... 不要文字削除, タブ挿入.

```
s/^  *//
s/^["]*¥([a-zA-Z']+¥)[.,:;"-]¥([^ ]+¥)$/¥1¥2/
s/|$/,/
#/^$/d
#n
/^ *#[0-9]/d
s/#/      #/p
```

この後は, 散文の場合の 6 以降と同じ処理を行う (6. SSS29.EXE, 7. collapse.pl, 8.refrm.sed).

4. 作表プログラム

(1) YGREP による索引ファイルの検索

上記のように作成した索引を YGREP によって検索すると, 例えば次のようなファイルが生成する.

表 4 tuvaṃの検索結果

```
- a:¥indx¥dn1.ind -
- a:¥indx¥dn2.ind -
tuvaM 2_288, 2_288.
  - a:¥indx¥dn3.ind -
- a:¥indx¥mn1.ind -
tuvaM 1_165, 1_165, 1_166, 1_166, 1_240, 1_240, 1_240, 1_240.
- a:¥indx¥mn2.ind -
tuvaM 1_338, 1_338.
- a:¥indx¥mn3.ind -
tuvaM 2_99.
- a:¥indx¥mn4.ind -
- a:¥indx¥sn1.idx -
tuvaM 1_131, 1_202.
- a:¥indx¥sn2.idx -
- a:¥indx¥sn3.idx -
- a:¥indx¥sn4.idx -
- a:¥indx¥sn5.idx -
- a:¥indx¥sn6.idx -
- a:¥indx¥an1.ind -
- a:¥indx¥an2.ind -
- a:¥indx¥an3.ind -
- a:¥indx¥an4.ind -
- a:¥indx¥an5.ind -
tuvaM 4_401.
- a:¥indx¥an6.ind -
- a:¥indx¥an7.ind -
- a:¥indx¥dhp.ind -
- a:¥indx¥snpv.ind -
tuvam 31.d, 545.b.
tuvaM 377.a, 377.d, 378.a, 545.a, 545.a, 545.a, 571.a, 571.a, 571.a, 571.b, 841.d, 1064.d, 1102.c,
1121.c, 1124.c.
- a:¥indx¥snpp.ind -
```

このリストに見られるように，tvaṃの併用語，svarabhakti を伴う tuvaṃという形は，4 ニカーヤの特定箇所に集中的に用いられ，決して一般用語に属していないことが解る（上記 dn1〜an7. しかも該当箇所の大半は韻文である.）．一方，Dhammapada (dhp) および Suttanipāta 散文 (snpp) には使用されず，Suttanipāta 韻文 (snpv) には 17 箇所に見える．このように索引ファイルの検索から，興味深いリストを得ることができる．

(2) 作表プログラム

作表プログラム（高島淳氏作 idx_rev.pl, tbl_cmp.pl, tbl_uni.pl）は，上記と同種のデータ（すなわち索引データを YGREP で検索した結果ファイル）をもとに，セクション毎に事象を集計表示する．表 5 は Suttanipāta 韻文について，Vagga 毎に集計したものである．

検索した事象は以下のとおり.「東部方言」あるいは「偈頌言語」に属するとされるものの数語形について使用状況を探った．古代インド・アーリヤ語と連なる古い事象 (-āse [13])，東インド方言による俗語的表現 (tuvaṃ, Prefixed verb + -tvā) 等の使用状況に関するこの種のデータを集積し，その分布を，韻律や内容と比較すれば，経典成立史研究の有力な手掛かりとなるであろう．

A. -āse (Nom.pl.): paṇḍitāse, upaṭṭitāse, etc.

B. -aṃhi (Loc.sg.): taṃhi, lokaṃhi, etc.

C. -smiṃ (Loc.sg.): tasmiṃ, lokasmiṃ, etc.

D. tuvaṃ (Pronoun, 2nd nom. sg.).

E. tvaṃ (Pronoun, 2nd nom. sg.).

F. p+tv (Ger, Prefixed verb + -tvā.): abhivādetvā, āruhitvā, etc.

[13] -āse に関しては，C. CAILLAT, "Doublets désinentiels en moyen indo-aryen." in *Bopp-Symposium 1992 der Humboldt-Universität zu Berlin*, Heidelberg 1994, pp.46 ff. 参照.

G. p+tn (Ger, Prefixed verb + -tvāna.): nisīditvāna, pabbajitvāna etc.

H. -āmase (Pres. 1st .pl): carāmase, bhaṇāmase, etc.

I. -are (Pres. 3rd pl.): upadissare, socare, etc.

表 5 Suttanipāta 韻文言語事象分布

		A -āse	B -aṃhi	C -smin	D tuvaṃ	E tvaṃ	F p+tv	G p+tn	H -āmase	I -are
snpv 1 uraga	1-17	2								
snpv 2 dhaniya	18-34				1				2	1
snpv 3 khagga	35-75		3	1			7			
snpv 4 kasibhv	76-82									
snpv 5 cunda	83-90									
snpv 6 parābhv	91-115			1						
snpv 7 vasala	116-142			1			1			1
snpv 8 metta	143-152			1						
snpv 9 hemavat	153-180			1			1			
snpv 10 āḷavaka	181-192									
snpv 11 vijaya	193-206			1						
snpv 12 muni	207-221		2							
snpv 13 ratana	222-238		1	2						
snpv 14 āmagndh	239-252					1				
snpv 15 hiri	253-257			1						
snpv 16 mhmaṅgl	258-269									
snpv 17 sūcilom	270-272									
snpv 18 dhcariy	274-283									
snpv 19 brhmdhm	284-315			2			3			
snpv 20 nāvā	316-323			1			4			
snpv 21 kiṃsīla	324-330						1			
snpv 22 uṭṭhāna	331-334									
snpv 23 rāhula	335-342			1						
snpv 24 vaṅgīsa	343-358					1				
snpv 25 smmprbbj	359-375	1								
snpv 26 dhammik	376-404	1			3	1				
snpv 27 pabbajj	405-424			1			1			
snpv 28 padhāna	425-449		1			1				1
snpv 29 subhāst	450-454									
snpv 30 sdbhrvj	455-486		1			3				
snpv 31 magha	487-509		1							
snpv 32 sabhiya	510-547			2	4	1				
snpv 33 sela	548-573			1	4					
snpv 34 salla	574-593									1
snpv 35 vāseṭṭh	594-656		1	4				1		1
snpv 36 kokāliy	657-678			2						
snpv 37 nālaka	679-723		1	3			1			1
snpv 38 dvyṭānp	724-765			2						
snpv 39 kāma	766-771									
snpv 40 ghṭṭhak	772-779	2		1						
snpv41duṭṭhaṭ	780-787									
snpv 42 suddhaṭ	788-795	1		1						
snpv 43 paramaṭ	796-803	1		2						
snpv 44 jara	804-813			1						
snpv 45 tmettyy	814-823			1					1	
snpv 46 pasūra	824-835	1	1	2		4	1			
snpv 47 māgandi	835-847			1	1					
snpv 48 purābhe	848-861			1						
snpv 49 kalahav	862-877	2		3						
snpv 50 cūḷavyh	878-894	1		1			1			
snpv 51 mahāvyh	895-914	2	1				2			
snpv 52 tuvaṭak	915-934			2						
snpv 53 atdaṇda	935-954			1						
snpv 54 sāriput	955-975		2							
snpv 55 vtthgth	976-1031		6	2			2			
snpv 56 ajitamṇ	1032-1039	1		1						2
snpv 57 tsmtymṇ	1040-1042									
snpv 58 pṇṇkamṇv	1043-1048			1						
snpv 59 mttgūmṇ	1049-1060			2		1				
snpv 60 dhtkmṇ	1061-1068				1					
snpv 61 upsīvmṇ	1069-1077									
snpv 62 nandamṇ	1077-1083	6					1			
snpv 63 hemkmṇv	1084-1087									
snpv 64 tdyymṇv	1088-1091			2						
snpv 65 kappmṇv	1092-1095			2						
snpv 66 jatukṇṇ	1096-1100			1						
snpv 67 bhdrvdh	1101-1104			1	1					
snpv 68 udymṇv	1105-1111									
snpv 69 poslmṇv	1112-1115									
snpv 70 moghrjm	1116-1119									
snpv 71 pṅgymṇv	1120-1123				1					
snpv 72 pratīka	1124-1149				1	2				

ＡＰＴＩへの誘い

江島 惠教・中谷 英明

1994.6.12.

APTI (= Association for Pali Text Inputting) は，PTS版パーリ語テキストデータベースの作成ならびにその利用を目的とするグループである．現在，テキストをOCRで入力し，その校正作業を遂行している．

１．校正作業に必要なハードウェアとソフトウェア

1. パソコン

- ハードディスクを装備していることが望ましい．
- マッキントッシュの場合はLHAが使えないので注意．
- NEC,EPSON,IBMは問題なし．

2. ソフトウェア

- エディター(VZの使用を標準とする)
- LHA.EXE(ファイルの圧縮・解凍のため)……手持ちでない場合はAPTIでドキュメントを添えて配布する
- 上記２点は校正作業のために不可欠．その他，FD, FMのようなファイラーがあれば便利．

２．校正作業の対象となるテキストファイルの状態

- ファイルは，PTS原本をOCRにより機械的に読み取ったもの．
- 原本300頁分が1ファイルになっている．
- ローマ字表記はKH方式に直している．[1]
- 頁数は各頁の冒頭に ^112 (=p.112) のように表示されている．

３．校正要領

1. 頁数の確認

頁数を機械が読み取れなかった場合，数字がなく，^ の記号だけが表示されていることがある．この場合は数字を補う．数字が入っている場合でも，機械が読み誤っている場合もあるので，必ず確認する．

2. 章題番号および章題等の取扱い

(1) 章題番号・パラグラフ番号・詩節番号は機械が読みとっているが，読み取れなかった場合，あるいは読み間違えている場合もあるので，確認する．

[1] KH方式の転写方法は本文書末尾の表参照．

(2) 番号の数字の種類 (1, 2, 3, .. or I, II, III) および大文字・小文字の使い分けは原本の通りとする.
(3) 番号の後のピリオドは, 機械が読み落としていることが多いので, 特に注意して, 原本に存在 してファイルで欠落している場合には必ず補う (なお, ピリオドの有無は原本によって一定していない. すべて原本通りを原則とする).

3. ファイルから削除する要素

この作業は, PTS 原本の本文だけを復元することを目的としている. したがって, 以下のものは, 機械が読みとっているが, すべて削除する.

(1) 各頁の最初に読みとられているヘッダ (柱)
(2) 本文中の注番号 (読みとられている場合)
(3) 各頁の最後に読みとられている脚注

4. 改行の処理

PTS 原本の通り物理的に改行する. 原本で 1 語が 2 行にわたっている場合も, ハイフンを残して原本通りに改行する.

5. 綴りの処理

(1) 綴りは PTS 原本をそのまま KH 方式に転写するのを原則とする.
ただし、PTS 本は文頭、および固有名詞の語頭を大文字としているが、これらは単語検索の便宜上小文字とする.OCR データの処理過程において (F21.SED) このように処理しているから原則としてデータテキストでは小文字になっているはずである.
(2) 明らかに誤植と思われる場合は, 誤植を含む語の直後に正しいと思われる語を< >で囲んで挿入する. 連続する語群で誤植がある場合は, 当該語群の直前に ¥ を挿入し, 当該語群の直後に正しいと思われる語 (群) を< >で囲んで挿入する.

ex. amAntesi < Amantesi >
¥sammA sambuddassa < sammAsambuddhassa >

(3) PTS 原本の表記には分かち書き方式について不統一がみられるが, すべて原本の通りとする (paJJaJca と paJJaJ ca とを統一しない).
(4) アヌスヴァーラ (ṃ) と m は、機械によって互いに読み誤られることが多く、また時には PTS 原本に不統一も見られるが、いちいち訂正しない (のちに一括修正を施す).

6. 記号の取扱い

(1) 複合語におけるハイフンの有無は, 原本の通りにする.
(2) 句読点 (. , ?) は原本の通りとする.
(3) ただし, 引用符はシングル, ダブルともにダブル・クォーテーションに揃える (オープンはクロ ーズで代用する). この結果, 引用符はすべて " になる (これは, シングル・クォーテーション・ マークとアポストロフィとが混同されるのを避けるためである).
(4) ダッシュはハイフン2字 (ー ー) で代用する.
(5) リーダーの点の数 (.. pe, ... pe) も原本の通りにする.

KH 方式[2] によるサンスクリット・パーリ語転写

Sanskrit	KH Transcr.	Sanskrit	KH Transcr.
ā	A	ṅ	G
ī	I	ñ	J
ū	U	ṭ	T
ṛ	R	ḍ	D
ṝ	q	ḷ	L
ḷ	W	ṇ	N
ṃ	M	ś	z
ḥ	H	ṣ	S

[2]中谷が『印度学仏教学研究』に発表した表では、ḷ の子音に W、母音に L を当てているが、パーリ、古典サンスクリットにおける使用頻度に鑑み、逆に子音に L、母音に W を当てることとする.

アープティマニュアル2

RECOGNITA PLUS 使用法

＜Ver.2 用＞

中谷 英明

1994.5.10.

このマニュアルは、文字読み取りソフト (Optical Character Recognition (= OCR) Software) "Recognita Plus" を使って、サンスクリット・テキストを入力し、データとして取り出す手順を述べる。記述項目は以下の通りである。

目次

"Recognita Plus"はMS-DOS 上で作動させる。Windows 上での使用も可能であるが、メモリー不足に陥ることがあるので薦められない。

東大、神院大の "Recognita Plus" は初期設定済みであるから、使用者は「3．立ち上げ・オープション設定」以降を読めばよい。

1．必要なハードウェアとソフトウェア

1.1 パソコン

- IBM またはコンパティブルのコンピューター。486 であることが望ましい。
- 4メガのラム (RAM)。
- ハードディスクには4メガ以上の空き領域が必要。
- スキャナー。300 DPI 以上であれば機種を問わない。

1.2 ソフトウェアおよびファイル

- Recognita Plus (SZKI Recognita Corp. H-1012 Budapest, Márvány u. 17, Hungary), Ver.2.0
- MS-DOS Ver.3.0 以上
- sktfg.set （本研究会にある）

2．インストールと初期設定

2.1 インストール

インストールは、"Recognita Plus" Ver.2.00 の場合には、3.5 インチフロッピーディスク3枚からなっているので、その1をドライヴに挿入し、

inst (return key)

で開始される。後は指示の通りに従って行けばよい。これでハードディスクへのインストールは終了する。

2.2 初期設定

以下の初期設定は最初に1度行えばよい。東大、神院大の機器は既に設定済みだから、設定変更の場合でなければ必要はない。

(1) スキャナーの設定、英語モードの変更などはDOS PROMPT 状態で、"Recognita Plus" を置いてあるディレクトリーに入り、
SETUPD (return-key)
とすることによって行う。

(2) "Recognita Plus"はマウスにより操作する。現在マウスはGMOUSE.SYS により作動している。[1]これはAUTOEXEC.BAT に
gmouse
という1行を入れることによって使用可能となる。

3. 立ち上げ・オープション設定

3.1 立ち上げ

"Recognita Plus" を立ち上げるには、例えば"Recognita Plus" をA ドライヴの 'Recogn' というディレクトリーに置いたとき、DOS PROMPT の状態で次のようにする。

(1) 'Recogn' というディレクトリーに入る： A > cd ¥recogn (return-key)

(2) "Recognita Plus" の起動： A > recogn (return-key)

(3) 初期メニュー画面における設定： 上記のようにして起動すれば、"Recognita Plus"の初期メニュー画面が現れる。そこにおいて以下に述べる項目の設定を行う。

3.2 レイアウトの設定

(1) テキストの紙長と紙幅

- 初期メニュー画面の 'LAYOUT' をクリックし（マウスのボタンを押し)、SCANNING AREA（スキャン領域）画面を開く。
- 単位を *cm* とする（UNITS の '*cm*' をクリック)。
- テキストの紙の長さ、左、下のマージンの長さを指定する。
  例： TEXT LENGTH: 25.0 *cm* MARGINGS: LEFT: 1.0 *cm*; TOP: 1.0 *cm*; RIGHT: 1.0 *cm*; BOTTOM: 3.0 *cm*

(2) 試し撮り： 'SCAN IMAGE' をクリック。適正な状態が得られるよう指定を変更する。

(3) スキャン領域は、マウスを左側のスキャン画面に移動し、クリックしたまま移動することによってWINDOW 領域を設定して、より正確に指定できる。

3.3 文字セットの選択

'OPTIONS' をクリックし、CHARACTER SET の選択欄において sktfg.set を選択してクリックする。

sktfg.set は、サンスクリット入力に必要な文字のみを残し、それ以外を削除した文字セットで、ROMAN8を基に作成されたものである。サンスクリット入力にはこのセットを用いるのが最も効率がよい。[2]

3.4 明度の決定

[1]マウスは他のシステムでも動く。
[2]文字セットの作り方は、"Recognita Plus" のマニュアル参照。

(1) 文字セットの選択を終えてから明度を決定する。'OPTIONS' のメニュー画面において 'OK' をクリックし、さらに WARNING の画面の 'OK' をクリックして初期メニュー画面に戻る。明度 (BRIGHTNESS) を表示マークの左から6番目くらいに合わせ、読み取り (READ) を実行。

(2) 1頁の読み取りが終われば、'NO MORE' を選択・クリック。SAVE TEXT FILE の項に適当なファイル名 (例えば xxx1.txt) を入力した後、OK をクリック。初期メニュー画面に戻る。[3]

(3) 'STATISTICS' をクリック。結果を明度ごとに記録。上記 BRIGHTNESS (左から数えたマーク数)、SCANNING TIME, DECOMPOSITION TIME, RECOGNITION TIME、 ACCURACY をノートする。

(4) 'O.K.' をクリック。初期メニューに戻り、BRIGHTNESS を1マーク右寄せし (LIGHTEN 方向に一つ動かし)、テキストの同じ頁を先ほどと同じように読み取り、STATISTICS を記録する。

(5) ACCURACY の値が次第に高くなり、次いで低くなって行くのを確認し、ACCURACY が最大値をとった時の BRIGHTNESS を当該テキストの最適明度とする。
ACCURACY の値が同じ時は、所用時間の少ない方を取る。

(6) 最適明度は主としてテキストの紙色によって決まる。したがって紙色が本の途中で変化する場合は、紙色に応じた最適明度をその都度決定する。
・普通の紙色では、最適明度は7または8である。

4．文字例の登録

4.1 文字学習機能のメニュー

明度を決定した後、特殊文字や、印字不良などのために読み取られない文字を登録する。これを"Recognita Plus"の「文字学習機能」と呼んでいる。

(1) スキャナーに代表的な1頁を置き、初期メニュー画面の 'CORRECT AND LEARN' を選択し、'READ' をクリックすればスキャンが始まる。

(2) スキャンは不明文字のところで止まり、学習メニュー画面が表示される。

(3) 'SUGGESTION' の欄に呈示された文字が正しければ 'ACCEPT'、誤っていれば正しい文字を 'CORRECT TO' に入力。

(4) 'SUGGESTION' に文字が呈示されない場合も、正しい文字を 'CORRECT TO' に入力。

(5) 正しい文字の入力法には、

- 文字をキーボードから入力、
- 'SHOW CHARACTER SET' をクリックして文字一覧を出し、その中から当該文字を選択して入力、
- ALT キーを押しつつ、文字コード番号によって入力。[4]

の3種がある。

(6) ただし次の文字は KH 方式とは異なる転写 (OCR 方式と呼ぶ) を用いるから注意を要する。文字学習時に入力する「正しい字」は、この OCR 方式によらなければならない。[5]

---

[3]ただし、「5.1 データ入力」の項の 「(3) 注意」参照。

[4]文字コード番号は、"Recognita Plus"のマニュアルの ROMAN 8 の項参照。あるいは上記 'CHARACTER SET' で、当該文字にカーソルを合わせると番号が下に示される。

[5]例えば PTS 本では文頭、および固有名詞の語頭が大文字になっているが、これらは単語検索の便宜上すべていったん小文字に変換しなければならないので、OCR 方式を用いるのである。(F21.SED が行う)。

Sanskrit	KH 方式	OCR 方式	Sanskrit	KH 方式	OCR 方式
ā	A	â	ṅ	G	Q
ī	I	î	ñ	J	ñ
ū	U	û	ṭ	T	F
ṛ	R	!	ḍ	D	Z
ḷ	W	#	ḷ	L	Ł
ṃ	M	%	ṇ	N	W
ḥ	H	&	ṣ	S	$

(7) 正字形を入力し、学習させる場合には、'LEARN' をクリックして、その形を登録する。無視または廃棄の場合は、'DROP GARBAGE' をクリックする。4～5頁分を学習させることが望ましい (4.2 (2) の第4項参照)。その場合 'OK' をクリックすると次の頁の読み取りが開始される。最後に 'NO MORE' をクリックする。学習文字形は、'tre' という拡張子の付いた学習文字ファイル (USER TREE と呼ばれる) として保存される。

(8) 最初の文字登録時には、USER TREE のベースネームを登録するためのメニュー画面が自動的に開く。そこに適当なベースネームを登録する。

例： AAA1.TRE (Abhisamayâlaṅkārâlokā Prajñāpāramitāvyākhyā の第 1TREE の場合)

(9) この学習文字登録が、"Recognita Plus" の設定において最も重要である。どの文字形をどこまで教えるかが "Recognita Plus" の性能を大きく左右する。

4.2 文字学習機能使用上の注意

文字例の登録にあたっては、次の諸点に注意する。

(1) 数字を 0 から 9 まで登録すること。そのほかスラッシュやその他の記号など、テキストに固有の文字は、あらかじめ一覧を作って、それらが実際に現れる頁をスキャンしてすべて登録する。

(2) ただし典型的な文字例のみを登録すること。

- メニュー画面に現れた文字形がインク滲みなどを伴っているとき、それがよくある種類のものでないときは、滲みの部分を画面上で消去して登録する。消去はカーソルを画面の不用部分におき、シフトキーを押す。
- 登録した文字例が多すぎると、RECOGNITION に時間がかかり、場合によってはメモリー不足も結果する。できるだけ異なる典型的な例を登録すること。
- 登録不用の場合には、正しい文字を 'CORRECT TO' に入力後、'LEARN' をクリックする代わりに、'CORRECT' をクリックする。こうすればその文字はその箇所で正されるが、USER TREE には登録されない。
- 文字例登録はテキストの状態によっても異なるが、およそ4～5頁分を行うとよい。こうしてできた TREE ファイルの大きさは 5,000 バイト程度が適度のようである。(勿論これは機械の性能によってはもっと多くしてもよい。)

(3) データとして採用しない文字の扱い：
例えば注釈をデータに入れないとした場合 (APTI はそのようにした) は、注番号や、注釈部分の文字は、文字登録してはならない。これらが学習メニュー画面に現れたときには、'DROP GARBAGE' をクリックする。

4.3 TREE の変更

初期メニュー画面に戻し、'OPTIONS' をクリック。下段の USER TREE の枠内をクリックし、新たな名称を入力する。'OK' をクリックした後、WARNING の 'OK' を再度クリックする。場合によってはメッセージに従って処理し、'OK' をクリック。初期メニューに戻る。

5．データの入力と取り出し

5.1 データ入力

(1) テキストファイル名の登録：
初期メニュー画面において、入力するテキスト名を例えば AAA1.TXT などと登録する。
(2) 本をスキャナーのガラス面の「右・上」に詰めて置く。
(3) 注意
- 本の頁の縁がガラス面の外枠と平行になるように置く。斜めになっていると著しく判字能力が落ちる。
- ガラス面にぴったりとは付けられない頁の綴じ目のところは、スキャン領域に入らないように注意する。この部分がスキャンされると、影を文字として解読しようとして大量のメモリーを消費し、ハング・アップする事が多い。

(4) 「文字セット」「明度」「USER TREE」などを確認し、また 'CORRECT AND LEARN' がクリックされていないか（■ではなく□ならよい）を確かめた上、'READ' をクリック。
(5) 30 ～ 40 秒で 1 頁のスキャン終了。
(6) 次頁をスキャナーに置いて、'O.K.' をクリック。すぐにスキャンが始まる。
(7) 50 頁を一つのファイルに保存するため、50 頁終了時点で、'NO MORE' をクリック。
(8) スキャン途中でハング・アップしたときは、機械本体の 'RESET' ボタンを押す。ハング・アップする直前までの頁のデータは、先に登録しておいたテキストファイル名によって保存されている。
(9) 50 頁終了時、あるいはハング・アップしてリセットした時は、テキストファイル名を例えば AAA2.TXT などと変えて次の 50 頁のスキャンを続ける。

5.2 データの取り出し・連結

(1) ハードディスクの 'RECOGN' というディレクトリー内には、データファイルが蓄積される。
(2) これを NEC のパソコンに移すには、3.5 インチフロッピーディスクの場合は、645 または 720 キロバイトにフォーマットした 2DD のものを使う。2HD は IBM 系には使えない。5 インチフロッピーディスクの場合は、2HD も使える。
(3) ハードディスクどうしを直接繋いでデータを転送するには MAX-LINK を用いてもよい。
(4) 取り出したデータは、6 ファイルを繋いで一つのファイルとする。300 頁分で 0.5 メガバイト弱になる。これくらいが最も扱い易い。
(5) ファイルの連結には、CAT.EXE が便利である。
- 例えば AAA1.TXT ～ AAA6.TXT を連結して AAA-1.TXT とするには：

```
A > cat aaa1.txt aaa2.txt aaa3.txt aaa4.txt aaa5.txt aaa6.txt >  aaa-1.txt
```

- あるいは一つの作業ディレクトリーに AAA1.TXT～AAA6.TXT を入れ、

```
A > cat  *.* > aaa－1.txt
```

としてもよい。

(6) こうしてできたデータ・ファイルの処理に関しては、マニュアル 2 「OCR データの処理手順」参照のこと。

アープティマニュアル3

ＯＣＲデータの処理手順
＜サンスクリット用＞

中谷　英明

1994.5.12.

このマニュアルは、「テキスト・データベース研究会」による作業の一環として、文字読み取り（OCR）ソフト[1] "Recognita Plus"から生成したサンスクリットテキスト・データを機械的に処理し、また手作業の校正を行ってデータベースとするまでの手順を述べる。

記述項目は次の通り。

1．既成スクリプトによる処理
2．頁番号付加・新スクリプト (1) の作成
3．新スクリプト (1) の実行
4．誤文字列訂正の新スクリプト (2) の作成と実行
5．VZ による校正
6．CLUP の利用

ただしデータは、"Recognita Plus"においてサンスクリット用の文字セット"sktfg.set"を用いて入力されたものとする。またソフトに関する参考文献などはこのマニュアルの末尾に挙げる。

1．既成スクリプトによる処理

"Recognita Plus"が作ったサンスクリットテキスト・データを、既に作られているスクリプト・ファイル[2]によって処理する。スクリプト・ファイルはバッチ・ファイル OCRSKT.BAT によって纏めて実行される。必要ファイルを一つのディレクトリーにおさめ、DOS PROMPT の状態で次のように入力し、リターン・キーを押す。[3]

○ 実行例：A ＞　ocrskt　b.txt　b1
○ 必要ファイル：sed.exe, ocrskt.bat, skt1.sed, skt2.sed, skt3.sed, skt4.sed, skt5.sed
○ 結果：　B1.OCR というテキストができる。

この B1.OCR はなおおそらくアルファベット以外の文字を含むので次の処理（EXALST.BAT）を実行する。 含まない場合も同じ処理を実行し、頁番号付けのみを行う。

[1] Optical Character Recognition Software
[2] 現在、計良龍成・中谷英明が作成した skt1.sed, skt2.sed, skt3.sed, skt4.sed, skt5.sed がある。今後これを増やすことが望ましい。
[3] ドライヴ A 内の一ディレクトリーにおいて、B.OCR というテキストを処理した場合を想定して以下に記述する。

2．頁番号付加・新スクリプト(1)の作成

2.1 非アルファベット文字摘出と変換スクリプト作成

EXALST.BAT は B1.OCR 中に残存するアルファベットに変換されていない文字を見つけだし、未完スクリプトの形で提出する。[4]

○ 実行例：A > exalst b1.ocr b1 （リターン・キー）

○ 必要ファイル：exalst.bat, ctvcln.sed, ctrl.sed, page.sed, tr.sed, exab1.sed, exab2.sed, ls2.awk, mr3.awk, mksed.sed, kara.sed, sortf.exe, awk.exe.

○ EXALST.BAT の実行内容詳細は次の通り。

*ただし %1 は入力ファイル名（いまは b1.ocr）、%2 は出力ファイルのデータベース名（末尾3字の拡張子を除いた部分。いまは b1）。

(1) 頁番号付け

```
sed －f ctvcln.sed %1 | sed －f ctrl.sed | sed －f page.sed
> %2.pg
```

(2) アスキー番号変更

これは後に非アルファベット文字を、その前後にアンダーバー（_）を置いて抽出するののに備え、あらかじめ非アルファベット文字の前後にアンダーバーを加え、抽出時に文字変換（Dump の変更）が起こらないようにするもの。

```
sed －f tr.sed %2.pg > %2.pgt
```

(3) 非アルファベット文字の抽出（出現回数付き）

%2.pln は非アルファベット文字を出現頻度順に列挙する。

```
sed －f exab1.sed %2.pg | sed －f exab2.sed | sortf |
uniq －c | sortf －nr > %2.pln
```

(4) 上記文字列を1字のみと2字以上に分け、それぞれの出現回数付きリストと SED スクリプト・ファイルを作る。

```
awk －f ls2.awk %2.pln > %2.ls2 （出現回数付き）
awk －f mr3.awk %2.pln > %2.mr3 （出現回数付き）
sed －f mksed.sed %2.ls2 > %2l2.sed （SED 未完表）
sed －f mksed.sed %2.mr3 > %2m3.sed （SED 未完表）
```

○ 結果：B1.PGT, B1LS2, B1.MR3, B1L2.SED B1M3.SED ファイルができる。

2.2 B1N.SED の作成

EXALST.BAT がテキスト B1.OCR から抽出した非アルファベット文字のリスト B1L2.SED[5] の各文字が原テキスト中ではどの文字に対応しているかを、B1.PGT（注意：B1.PG ではない！）と原テキストを比べて確認し、対応する文字列をスクリプト右列に書き込む。これを B1N.SED[6] として保存。 なおローマ字転写方式は KH 方式を用いる。

[4] EXALST は Extra-alphabetical List の略。

[5] B1L2.SED は、B1.TXT の2コード以下 (less than 2 codes) の非アルファベット文字（ローマ字、半角カナ、漢字など）のリストという意味。同様に B1M3.SED は、B1.TXT の3コード以上 (more than 3 codes) の非アルファベット文字のリストである。

[6] B1N は、B1.TXT の New Sed Script の略。

○ B1L2.SED の例：
↓非アルファベット文字
s / キ / / g
s / タ / / g
s / チ / / g

○ B1N.SED の例：
↓原テキストの文字
s / キ / J / g
s / タ / A / g
s / チ / e / g

(注意)

- テキスト中に於ける対応文字の確認は、少なくとも3箇所に於いて行うこと。
  データベースの1文字が異なった文字に対応するときは、頻度の高いものをとること。
- B1L2.SED、B1M3.SED に於いて空白と見える部分にも文字はある。その文字は VZ のファンクションキー f.5 によって記憶させ、CTRL+C でテキスト中に探すこと。[7]
- f.5 キーは直前に記憶した文字列を SHIFT + f.5 によって出すことができる。

2.3 B1M3.SED の処理

B1M3.SED に於ける文字列が B1L2.SED の文字対応に一致するかを確認し、一致しない場合にそれを書き込む。一致したものは消去。B1M3.SED と B1L2.SED を B1M3.SED を先頭として結合、B1N.SED とする。(この作業は実際上ほぼ不必要であるから、詳しい説明は省略。)

3．新スクリプト (1) の実行

- B1N.SED ができたならば、これをテキスト B1.OCR にかけて、非アルファベット文字をすべてアルファベットに変換する。これで新テキスト B2.OCR ができる。
  ○ 実行例：
  A > sed －f b1n.sed b1.ocr > b2.ocr
  ○ 必要なファイル：B1N.SED
  ○ 結果： 非アルファベット文字を含まないテキスト B2.OCR ができる。
- 実行後、正しく変換されているかを確認する。VZ に新旧両テキスト B1.OCR と B2.OCR を読み込み、f.4 キーで画面分割する。
  確認事項は次の2点。

(1) スクリプトの正否の確認：
上記の状態で SHIFT を押しつつ f.3 キーを押す。カーソルの位置は両画面の同じ位置になければならない。カーソルは両テキストが互いに異なっている位置まで動いてとまる。
テキスト間に違いがあってカーソル位置がずれたときは、対応する位置に置き直せば、その位置以降の両テキストの比較を行うことができる。こうして新テキストが元のテキストからどのように異なっているか、それがスクリプト通りであるか、幾つかの例で確認しておく。

(2) SED の動作確認： SED は処理するテキストが大きくなりすぎると、ときに途中で作業を中止し、以後のテキストを切り落として終了することがある。両テキストの末尾を比較して、このようなことがないか確認する。(テキスト末尾を見るには、コントロール・キーを押しつつ C を押す。)

- B1N.SED に異常がないか確認できれば、先に述べた (第1項)「既成スクリプト」との整合性を確認し、既成スクリプトの末尾に付け加える。

[7] VZ 使用法の詳細は、下記 「5. VZ による校正」参照。

4．誤文字列訂正の新スクリプト(2)の作成と実行

B2.OCRはなお多くの誤文字列を含む。しかもそれは一定のパターンに収まるものが多い。a, s, e などの間に起こる誤読、h を l と i に読むなど、印字のかすれによる一定の誤読は、一括変換することが可能である。

上記1．に述べた「既成スクリプト」により既に修正されたものもあるが、なお極めて不十分である。VZ で変換してしまわずに、B1N.SED と同様、変換作業を SED スクリプトの形で蓄積してゆくことが望ましい。

ただし、このスクリプト・ファイルの作成には、B1N.SED の場合よりさらに正確な「SED.EXE」および「正規表現」に関する知識が必要である[8]。以下にその手順と注意事項を概説する。

4.1 空スクリプト・ファイルへの書き込み

- 空のスクリプト・ファイル KARA.SED が用意してあるのでそれに変換文字列を書き込む。

  KARA.SED
  s///g
  s///g
  s///g

  例えば、tam とあるべきところに、データ中では tariG, tarii, tariz などとあれば、

  s/ariG/aM/g
  s/arii/aM/g
  s/ariz/aM/g

  と書き込めばよい。ただし上の3行は、

  s/ari[izG]/aM/g

  と纏めることができる。この方が処理速度も上がることになる。
- 同様に

  s/istA/iSTA/g
  s/ista/iSTa/g
  s/isth/iSTh/g

  は

  s/ist¥([aAiIuUeoh]¥)/iST¥1/g

  と纏めることができるが、この場合は余分に変換してしまった vistara- などを次のように回復しておかなければならない。

  s/viSTar/vistar/g
  s/viSTIr/vistIr/g

  変換は、詳細な条件を立てて正しい変換のみを行うより、このように小数の例外を無視して行い、後に修正するのがよい。

4.2 スクリプト・ファイル作成上の注意

- 作成したスクリプトの行が多数にのぼるときには、ソートをかけて、重複規定がないか調べる。例えばスクリプト・ファイルを B2N.SED とした場合、アルファベット順、またはサンスクリット順にソートする。ここではサンスクリット順の例。

  ○ 実行例： A > ssss29 b2n.sed > b2ns.sed

[8]「SED」「正規表現」に付いては、『MS-DOS を 256 倍使うための本』Vol.3.（アスキー出版）参照のこと。

○ 必要なファイル： sss29.exe, sss.ord, b2n.sed.

例えば次の2行が見つかったとし、かつ上の行のスクリプトが妥当とすれば、下の行は不用である。

s/arG/an/g

s/arGu/anu/g

- ただしソートのおり、次のような2行の順序が上下逆にならないように注意すること。[9]

  s/upaGi/upan/g

  s/paGi/pari/g

- そのほか「正規表現」において特殊な意味を持つ記号に注意すること。例えば、

  s/k.S/kS/g は s/k¥.S/kS/g でなければならない。

- 大文字と小文字で意味の異なるコマンドがあるから注意。

4.3 スクリプトの実行

できあがった B2N.SED を B2.OCR にかけ、B3.OCR として保存。

○ 実行例： A > sed -f b2n.sed b2.ocr > b3.ocr

実行後、正しく SED が機能したか、スクリプトファイルの書き方が正しかったかを、B2.OCR と B3.OCR を比較して確認。(VZ を用いての確認方法は上記、「3. 新スクリプト (1) の実行」参照)

こうして校正用テキスト B3.OCR ができる。また新たにできた B2N.SED も既成スクリプトとの整合性を確認の上、それに付け加える。

5．VZ による校正方法

エディター上での校正には一括変換を多用する。類似の誤読が多いからである。エディターは MIFES でもよいが、データの大きさ（約 0.5 メガバイト）からすると、一括変換の速い VZ が望ましい。[10]

- VZ 上にデータを読み込み、次の手順で校正する。

  (1) 誤文字列にカーソルを当て、f.5 キーを押して記憶させる。記憶の範囲は画面の左下に出る。1度で文字列の全体が記憶されないときは続けて f.5 を押す。

  (2) f.7 を押して置換のウィンドウ「検索文字列」を開く。

  (3) 上向き矢印キー（↑）を押して先に記憶した文字列をウィンドウ中に呼び出す。できるだけ多くを一括変換するため、誤った変換とならない最小形を考え、ウィンドウ中の不用な文字を消す。[11]

  (4) リターンキーを押し、「置換文字列」のウィンドウに変える。

  (5) 再度↑を押して先に登録した「検索文字列」（誤文字列）を呼び出す。これを正しく修正した後、リターンキーを押して一括変換。

  (6) バックアップをこまめに取り、誤って取り返しのつかない変換をした場合に備えること。

  (7) VZ のバックアップ・ファイルは、カレント・ディレクトリーに入ると煩わしいので、autoexec.bat に

  set VZBAK =¥BAK

  という1行を入れ、¥BAK というディレクトリーに入るようにするとよい。

- KH 方式では読みづらい場合には、VZ 画面上で弁別記号付文字が見える VZSKT を用いてもよい。このVZは、検索、置換において正規表現を使うこともできる。

[9] SED は各行に対して、上の行のスクリプトから順番に適用してゆく。今の場合逆になれば、2行目、すなわち s/upaGi/upan/g は意味がなくなる。

[10] VZ に関しては、『VZ 再履修』武井一巳、兵藤嘉彦著（ビレッジセンター出版）が簡明である。

[11] 例を Pāli にとるならば、bhikbhave という誤読があるとき、bhikbhav > bhikkhav というスクリプトでは、数カ所の訂正に留まるが、kbhav > kkhav ならばかなり多くを訂正できる。この場合、Pāli には kb という子音群は存在しないうえ、字形の近さも勘案して、kb > kh と置換して大過はなく、実際、W21.SED において行っている。また音韻分析プログラム PHAP の利用については、APTI マニュアル2参照。

(1) VZSKT に必要なファイル：

skt98.com, skt98.def, skt98.bat, ezkey.com, fntld.com, vwx.com,
kh2vz.def, vz2kh.def, tm2vz.def, vz2tm.def, vzskt.key, vzrexp.doc.

(2) このうち kh2vz.def は、KH 方式のテキストを VZSKT 方式に変換するマクロである。

使い方は、SKT98.BAT によって VZSKT を立ち上げ、KH 方式テキストを読み込んだ後、ESC キーを押し、続いて ^ キーを押してマクロメニューを呼び出し、kh2vz.def を選んでリターンキーを押せば、画面上で KH 方式は弁別記号付き文字に変換される。

(3) この状態で弁別記号付き文字を入力するには、NFER キーを押したあと、たとえば a と入力すれば、ā が入力され、画面に現れる。文字配列一覧は vzskt.key にある。

(4) 逆に VZSKT 方式を KH 方式に転換するには、vz2kh.def を用いる。テクノメイトとのデータ交換も可能（tm2vz.def, vz2tm.def)。

(5) ただし NEC ノートパソコンのレジューム機能使用中は、VZSKT をノートパソコンから立ち上げることはできない。いったんフロッピーから立ち上げるとパソコンからも立ち上がるようになる。

(6) また FD の VZ 立ち上げ機能や、その他 VZ のマクロを使うもの（スペルチェッカー、n_tex.def など）とは共存できない。

6．CLUP の利用

パーリ・テキストには PHAP (= Phonetic Analysis Program for Pali) があり、ことにその PHAPERP (= PHAP for Listing up Error Possible Forms) は、校正に大いに資しているが、サンスクリット用のものは未完である。

しかしテキスト中のすべての音群を出現回数順にリストアップする CLUP (= Cluster List-up Program) は、これだけでも校正に十分活用することができる。[12] 出現回数の少ない音群は、しばしば誤綴であるからである。

上記の一次修正テキスト B2.OCR ができた時点で、誤文字列変換スクリプト作成時に利用する。

○ 実行例： A > clup b2.ocr b2

○ 必要なファイル： clup.bat, clear_cl.sed, c_clus.pl, v_clus.pl, jperl.exe, sortf.exe

○ CLUP.BAT の実行内容は次の通り

(1) 行頭記号の添付と不用文字の消去

sed -f clear_cl.sed %1 > %2.clr

(2) 古い pag ファイルと dir ファイルの消去

del * . pag / del * . dir

(3) 母音どうし、子音どうしの 1 字以上の連結をすべて列挙、出現回数順に並べる。

jperl c_clus.pl %2.clr > %2.c
jperl v_clus.pl %2.clr > %2.v
sortf -t= -n +1 -2 +0f -1 %2.c > %2.c2
sortf -t= -n +1 -2 +0f -1 %2.v > %2.v2

最後に出て来る %2.c2 と %2.v2 の中の回数の少ない音群を検討して、誤った綴を見つけ、SED または VZ で修正する。

[12] 高島淳氏作の JPERL によるプログラム C_CLUS.PL および V_CLUS.PL に中谷が手を加えた。

＜参考文献＞

(1) MS-DOS に関して

・『入門 MS-DOS』,『実用 MS-DOS』,『応用 MS-DOS』　村瀬康治著　（アスキーラーニングシステム，アスキー出版）

(2) VZ に関して

・『ＶＺ再履修』武井一巳、兵藤嘉彦著　（ビレッジセンター出版）

(3) SED, 正規表現, および uniq.exe などの TOOLS に関して

・『MS-DOS を 256 倍使うための本』Vol.1, Vol.2, Vol.3 （アスキー出版）

(4) AWK に関して

・『プログラミング言語ＡＷＫ』エイホ・カーニハン・ワインバーガー著、足立高徳訳（トッパン情報科学シリーズ）

インド学研究とコンピュータ利用

＜初級編＞ Ver. 1.02

中谷 英明

1994.7.14.

はじめに

このマニュアルは、インド学研究遂行上に必要なコンピュータ知識を獲得させるために編んだものである。初級、中級、上級の3編からなり、それぞれ次の目標にそっている。

- 初級 ..... コンピュータのハードとソフトに関する基礎知識。初めてコンピュータに接した人が、基本的なソフトを使いこなせるようになるまで。
- 中級 ..... 種々のソフトの特性把握。すなわち種々の研究目的を最も簡単に達成する方法に関する知識。
- 上級 ..... 韻律分析プログラム、文字列分析プログラム、スペルチェッカー、統計表作成プログラム、統計分析プログラムなどの使用法ならびに修正方法の修得。

初級は基本ソフト、中級はひろく必要なすべてのソフトを手順どおりに使う方法、上級はそれらソフトを組み合わせるなどして作られたプログラムの修正や応用を解説する。従って中級終了時点で、有用なソフトに関する一応の知識を得る事になる。[1]

初級編目次

- 初級編はパソコンの購入から、バッチファイルの使用法までを解説する。
- 解説は、機械へのソフトのインストールから始め、ソフトの基本的な操作にいたるまで、作業の手順に従っている。従って本マニュアルの記述通りに作業を行えば、必要なファイルが各自のパソコンにしかるべく配置され、その基本的な使用法が理解できるようになっている。

[1]中級、上級編はもとより、この初級編も現時点 (1994 年 7 月 10 日) においては未完成である。種々改善・補筆すべき点について、助言下されば幸いである。

1．ハードウェアー

1.1 ハードウェアーの選択

- コンピュータ機器は大きく、MS-DOS 搭載の NEC・IBM 系と、Macintosh OS (=Operating System) による Macintosh に分かれる。ただし DOS とは、Disk Operating System の略であり、Microsoft 社の DOS を MS-DOS と呼ぶ。
- NEC・IBM 系と Macintosh の優劣は一長一短があって一概に言うことはできない。大まかには、パーソナルなプログラムの開発や、大量のデータ処理には MS-DOS が向いており、簡単で美麗な印刷には Macintosh が便利、ということになるだろうか。
- 本文書は、MS-DOS によるソフトを解説する。しかしながらここに解説する多くのソフトは Macintosh でも使えるか、相当する Macintosh 用ソフトが存在するから、Macintosh ユーザーの参考になることもあるだろう。
- 価格と機能との関係からすれば、初心者にはノートパソコンが薦められる。
- プリンターとディスプレイ（ノートパソコンはディスプレイ付き）は予算に応じて整えることにし、留意すべきことを以下に若干述べる。

1.2 必ず必要なもの

(1) CPU (= Central Processing Unit)

CPU （中央演算装置）はコンピュータの中心部であり、その機能は 1 度に処理する bit 数（16 bit, 32 bit[2] など）、および、コンピュータ信号のクロック周波数（20 MHz, 40 MHz[3]など）によって示される。

データの量、扱い方にもよるが、文字情報のみであっても、32 bit 機、その中でもできれば 486 機が望ましい。クロック数も 20 MHz 以上が望ましい。

(2) ハードディスク

ハードディスク（外部記憶装置の一つ）は、100 メガバイト[4]以上の容量が望ましい。

1.3 あることが望ましいもの

(1) ラム

ラム（RAM = Random Access Memory）とはデータの書き込み、読み出しが可能な半導体記憶素子。ハードディスクより高速なデータ処理が可能。

初心者は無しですますこともできる。もし備えるとすれば、8 メガバイト以上のものが望ましい。後に述べる TEX は 2 メガバイトのラムを必要とする。

(2) 目を保護するスクリーン

デスクトップのディスプレイには、画面前面に付けて目を保護するスクリーンを付けた方がよい。

---

[2]bit はコンピュータにおける最小情報単位。binary digit（2 進数）の略。

[3]MHz は「メガ・ヘルツ」。

[4]バイト (byte) とは、コンピュータが一纏めに扱う情報単位。8 個の bit を 1 バイトとして扱う。bit は 2 進数であるから、8 個では 256 通りを表現できる。

2．MS-DOS を立ち上げる

2.1 セットアップ，フォーマット，領域確保，システムの転送[5]

(1) パソコンのセットアップ

新しくパソコンを購入した場合には、先ず付属の「セットアップディスク」によってパソコンをセットアップする。手順どおりに行うと、最後にそのディスクに入っていた MS-DOS が立ち上がり、命令（コマンド）待ちの状態となる。これを DOS PROMPT 状態と呼び、

```
A >
```

が表示され、その右側で四角形（カーソルと呼ぶ）が点滅している。

(2) ハードディスクのフォーマットと領域確保

次にハードディスクの「フォーマット」および「領域確保」を行う。MS-DOS のヴァージョン５の１枚目のディスクをドライヴに入れ、DOS PROMPT 状態で

```
A > format b: [6]
```

と入力後、リターンキーを押す。

「領域確保」は、１ドライヴに 128 メガバイト（ハードディスクが 128 メガバイト以上の場合）を確保しておくとよい。こうすれば後に光磁気ディスクによってファイル（データやプログラムのテキスト）のバックアップを取るとき、3.5 インチのディスク１枚に収まって好都合である。[7]

(3) MS-DOS のインストール

ハードディスクへの MS-DOS のインストールは、MD-DOS の１枚目のディスクをフロッピーディスクドライヴに差し込み、リセットボタンを押せば、作業が指示されるからその通りに行えばよい。

(4) MS-DOS のシステムの転送

すでに使用している MS-DOS のヴァージョン・アップを行うときは、単に前の DOS を消去、新しいものをコピーするだけではだめである。MS-DOS のシステムファイル

IO.SYS
MSDOS.SYS

は、ディスク中においてあるべき位置が決まっており（ルートディレクトリーの先頭）、そこに置くために、

SYS.COM

がある。例えば新ヴァージョンの MS-DOS の入ったフロッピーディスクのドライヴが A、それをインストールするべきドライヴが B の時は、DOS PROMPT の状態で、

```
A > sys  b:  (return key)
```

とすることによって、上の２個のシステムファイルと COMMAND.COM が B ドライヴに転送される。他の MS-DOS ファイルは例えば DOS というディレクトリーを作って、そこにコピーするだけでよい。コピー方法については後述。

---

[5] 近ごろのパソコンには、セットアップ、フォーマット、領域確保済みのものが多い。その場合にも、以下の「(4) システムの転送」の項を読んでおくこと。

[6] ここに記述するのは、フロッピーディスクのドライヴが A、ハードディスクのドライヴが B の場合である。「ドライヴ」と「ディレクトリー」については後述。異なっていれば別の入力になるから注意。

[7] 近頃のハードディスクは「フォーマット」「領域確保」済みのものも多いが、領域は 128 メガ確保されているようである。

2.2 ドライヴ，ディレクトリー，CONFIG.SYS，AUTOEXEC.BAT

(1) ドライヴとディレクトリー

データやプログラムのファイルを格納しておく場所には、ドライヴとディレクトリーがある。

- ドライヴとはDisk Drive （駆動装置）のことである。フロッピーディスク用、ハードディスク用、ラムディスク用などがある。
- ディレクトリーとは、一つのドライヴメディア（フロッピー、ハードディスクなど）の中に幾つか設け、ファイル（ディスクに記憶されたデータやプログラム）を分類して格納するための区分けのこと。一つのディレクトリー中にサブディレクトリーを幾つか設け、さらにその中にサブディレクトリーを設ける、というように階層を作る。これを木の枝分かれに喩えてディレクトリーの「ツリー構造」という。
- ドライヴはディレクトリーに対する上位区分であるが、ディレクトリーの場合と異なり、一つのドライヴを超えては作業しないプログラムが少なくない。その意味でドライヴは1本の木に当たる。異なる木の間の連絡は、後述の「パス（PATH）」によってつけることもできるが、そうしたとしても、例えばファイルのありかを探すQW.EXEというプログラムは、1ドライヴ内を探すだけである。

(2) コンピュータへの電源の投入

(a) BOOT LOADER, IPL, FAT, DIRECTORY:

上に述べたようにしてハードディスクににMS-DOSをインストールしたのち、コンピュータに電源を通じると、コンピュータは先ず本体付属のROM (Read Only Memory) に書き込まれているBoot Loaderというプログラムを読み、それに基づいてハードディスクに書かれているIPL (Initial Program Loader)、FAT (File Allocation Table)、Directory情報、を読み込む。

FATは各ファイルの在処（ありか）を示す情報であり、この部分が何らかの原因で破損するとファイルは完全な形で取り出せなくなる。

逆に、例えばファイルをDELETE（消去）したとしても、FAT情報が書き変えられるのみで、本体テキストは元のままであるから、消去の直後であれば消去ファイルを復活することができる。[8]

(b) CONFIG.SYS:

こうしてドライヴ内のファイルに関する情報を得たコンピューターは、次に、そこにCONFIG.SYSというファイルがあればそれを読み込む。

CONFIG.SYSにはMS-DOSの動作状態 (Configuration) を決定する事項が書き込まれているので、それが実行される。

すなわちCONFIG.SYSは、一時に読み込む「ファイルの数」や、データを一時ためておくための記憶領域である「バッファの数」を指定したり、「ラムボードの使用法」や「日本語システム」を登録するなどする。

(c) AUTOEXEC.BAT:

次にコンピュータはAUTOEXEC.BATというファイルを探し、そこに書き込まれている命令を実行する。

例えばパス (PATH) はここに指定されている。パスはCOMMAND.COMがコマンド（命令）を受け取ったときに、それを実行するプログラムを探すべきドライヴ、ディレクトリーを指定するものである。

[8]消去ファイルを復活するには、DOS PROMPTにおいて

```
undelete (return key)
```

とし、後はその指示に従えばよい。

また AUTOEXEC.BAT には特定のソフトの起動命令を書いておくこともできる。例えば電源投入時にファイル・マネージャーである FD.COM を自動起動させるには、

1）FD.COM のあるディレクトリーにパスを通す。「パスを通す」とは、AUTOEXEC.BAT に

path = a:¥fd （FD.COM が A ドライヴの FD というディレクトリーに在るとき）

などと書くこと。これによって FD.COM の所在が COMMAND.COM に知られることになる。

2）次に AUTOEXEC.BAT に

fd

と書くだけでよい。これでコンピュータに電源を投入したとき、様々な設定がなされた後、自動的に FD が立ち上がって入力待ちの状態になる。

(d) まとめ

電源投入（あるいはリセット）されたとき、コンピュータが自動的に行うことはここまでである。これまでの動作を図示すれば下の通り。

＜電源投入＞ ＜命令待ち状態＞

↓ ↑

BOOT LOADER → IPL, FAT, DIRECTORY → CONFIG.SYS → AUTOEXEC.BAT

3．MS-DOS の内蔵コマンド

3.1 IO.SYS, MSDOS.SYS, COMMAND.COM

MS-DOS の起動時における作業の流れは上に見たが、ここで MS-DOS のシステムを構成する 3 個のファイルの働きを見ておく。

- IO.SYS
  MS-DOS 本体部（次に述べる MSDOS.SYS）とコンピュータのハードウェア（キーボード、ディスプレイ、プリンター、ハードディスクなど）の連絡をおこなう。入出力 (Input-Output) システム部。
- MSDOS.SYS
  MS-DOS の本体部。FAT (File Allocation Table) や Directory のツリー構造などを管理するシステム部。
- COMMAND.COM
  コマンド（命令）処理部。IO.SYS、MSDOS.SYS を通じて送られてきたコマンド文字列を判別し、実行する。

  COMMAND.COM 自身もプログラムを幾つか内蔵している。これを内臓コマンドと呼ぶ。先に見た PATH も実はこの内臓コマンドの一つである。これに対し、COMMAND.COM 以外のすべてのプログラムを外部コマンド（外部プログラム）と呼ぶ。

  入力されたコマンドがこの内蔵プログラムの（内臓コマンド）に対するものであれば COMMAND.COM 自身が実行し、結果は COMMAND.COM から MSDOS.SYS を経て IO.SYS に送られる。IO.SYS はそれを画面やハードディスク、プリンターなどに出力する。外部コマンドであれば、COMMAND.COM は外部プログラムに実行命令を出し、結果は直接 MS-DOS に送られる。図示すれば下の通り。

＜コマンド入力（キー入力）＞ ＜画面、ディスクなどへの出力＞

↓ ↑

IO.SYS → MSDOS.SYS → COMMAND.COM → MSDOS.SYS → IO.SYS

↓ ↑

外部プログラム

3.2 内臓コマンドを使う

さて電源を投入すればMS-DOSが起動し、DOS PROMPT（A > の右側でカーソルが点滅する）状態で止まって入力待ちとなる。ここでCOMMAND.COM内臓コマンドを使ってみよう。

(1) DIR　(ディレクトリー・ファイル一覧)

キーボードからdirと画面に入力し、リターンキーを押す。

```
A > dir (return key)
```

画面にはCOMMAND.COMファイルを始め、Aドライヴにあるすべてのファイルとディレクトリーの一覧が出る。

もしファイルやディレクトリーが多すぎる場合は、画面は上にスクロールして先頭部は見えなくなる。このようなときは

```
A > dir /w (return key)
```

とする。

(2) MKDIR　(ディレクトリーの作成)

次にFDという名の新しいディレクトリーを一つ作る。次のように入力し、リターンキーを押す。

```
A > mkdir fd (return key)
```

ここで再び

```
A > dir (return key)
```

とすると今度は

FD < DIR >

が追加されていることがわかる。

(3) COPY　(ファイルのコピー)

ここで新しく作ったFDというディレクトリーに、FD関係のファイルをコピーする。

研究室に用意されている<FD関係ファイル>というディスクをフロッピーディスクドライヴに入れる。

フロッピーディスクのドライヴがBとすると次のように入力後リターンキーを押す。

```
A > copy  b:¥fd¥fd*.*  a:¥fd
```

(4) CD　(カレントディレクトリーの変更)

ここでFDというディレクトリーをのぞいて見よう。そのためには作業場所、すなわちカレントディレクトリーを、現在のAドライヴのルートディレクトリーからFDというディレクトリーに変更しなければならない。

```
A > cd fd (return key)
```

ここで先ほどと同様にしてファイル一覧を見る。

```
A¥fd > dir (return key)
```

すると以下の4種のファイルがそこにあることがわかる。

```
FD.COM, FD.CFG, FD98.COM, FDCUST2.COM
```

4. FD を使う

これまでの作業によってファイルマネージャー FD が使えることになった。これ以降の作業はこの FD を使ってする事にする。FD を使うと、上記の MS-DOS のコマンドをいちいち書く手間を省くことができる。

上に述べた内臓コマンドはすべて既に FD に組み込まれているから、いずれも２～３タッチで済ませることができる。またその他の外部コマンドも、FD のメニューに登録しておけば、同様に起動することができる。[9]

以下に FD の便利な機能をいくつか紹介しよう。[10]

(1) ファイルのコピー・移動・消去

- スペースキーによりファイルをマークする（すなわち、カーソルを当該ファイルに合わせ、スペースキーを押すとアステリスクがファイル名先頭に付加される)。
- こうしてコピー・移動・消去などを行う。
- 先に我々は内臓コマンドにより、FD というディレクトリーを作り (MKDIR)、そこへ FD 関係ファイルをコピーした (COPY) が、このディレクトリーにはこのほか ISH, RISH, MIEL, QW などの EXE, COM ファイルや VED.BAT などを入れておくのがよい。これらは用意したフロッピーディスクに入っている。フロッピーのドライヴを B とすれば手順は次の通り。
  (a) FD のファイル一覧画面において L(ログ・キー) を押し、次に b を入力して RETURN を押し、ドライヴを B に変える。
  (b) FD<DIR>にカーソルを合わせ、RETURN を押し、FD というディレクトリーに入る。
  (c) HOMECLR キーを押して、ディレクトリー内の全てのファイルにアステリスクを付ける。
  (d) SHIFT+C を押し、次に A を入力してドライヴ A のディレクトリー・ツリーから FD をカーソルによって選択、RETURN を押す。
  (e) 先にコピーしてあったものもあるので、「同名のファイルがあります」「新しい日付をコピー」というメッセージが出るがそのまま RETURN を押せばコピーが完了する。

(2) ディレクトリーの削除

- カーソルをディレクトリーに合わせ、SHIFT+D キーを押すことによって、サブディレクトリーも含めて一挙に消去する。

(3) VZ の立ち上げ

- カーソルを編集するファイルに当てて、SHIFT + RETURN とすると登録してあるエディター (VZ) が当該ファイルを読み込んで立ち上がる。
- 複数のファイルを立ち上げるときは、スペースキーにより読み込むファイルをマークした後、XFER あるいは NFER きーを押してメニュー画面を開き、V キー（VED.BAT 起動キー）を押すと、マークされたファイルをすべて読み込んでエディターが立ち上がる。

(4) ファイルの圧縮（凍結）・解凍

- SHIFT+f.10 により圧縮、f.10 により解凍を行う。

(5) 圧縮ファイルの内容を見る

- カーソルを圧縮ファイルに当て、XFER+M を押すことにより、MIEL.EXE が立ち上がり、圧縮ファイル中のテキストを見ることができる。

[9]登録することができるのは 52 種まで。
[10]FD の機能全般に関しては、FD.DOC を参照のこと。

(6) その他

- この他ディレクトリー作成やディスク（ドライヴ）情報、ファイル属性を知ることもできる。詳しくはFD.DOC参照。
- VZ, VZS, DOS, DOCなどのディレクトリーを作り、そこに上の要領でフロッピーディスクから各ファイルをコピーしておく。例えばVZというディレクトリーを作る手順は次の通り。
- Kを押すと、「新しいディレクトリー名を入れて下さい」というメッセージが出るから、VZと入力し、RETURNを押すとVZという新ディレクトリーが作成される。

5. VZとVZSKTを使う

1. VZの機能[11]

(1) 記憶(1) 文字列

- 文字列にカーソルを当て、f.5キーを押して記憶させる。記憶の範囲は画面の左下に出る。1度で文字列の全体が記憶されないときは続けてf.5を押す。
- こうしていったん記憶した文字列は、いろいろに利用できる。例えば次のようである。

(2) 検索・置換

- そのまま、CONTROL+Cを押せば、前方方向にその文字列を検索する。
- いったんf.6で「検索文字列」のウィンドウを開き、上向き矢印キー（↑）を押して先に記憶した文字列をウィンドウ中に呼び出す。これに修正を加え、RETURNを押し、先と同じ要領でCONTROL+Cによって前方検索ができる。
- 置換のウィンドウにも同様にして記憶させた文字列を呼び出すことができる。

(3) 挿入

- 上のように記憶（あるいはf.6のウィンドウでそれを修正）した文字列を、SHIFT+f.5を押すことによって、何度でも任意の箇所に挿入できる。
- あるいはSHIFT+f.7（複写2）によってウィンドウを開き、上向き矢印キーを押して以前に記憶した文字列を次々と呼び出して、適当なものを何度でも挿入することができる。

(4) 記憶(2) 複数の行

- 複数行の記憶は、f.10を押してブロックモードに入り、矢印キーでカーソルを移動して記憶するべき範囲を指定して、SHIFT+f.8によって記憶する。
- こうして記憶した複数行、あるいは複数行にわたる文字列は、SHIFT+f.9(ペースト)によって、繰り返し挿入することができる。

2. VZSKTの使用法

(1) KH方式では読みづらい場合には、VZ画面上で弁別記号付文字が見えるVZSKTを用いてもよい。このVZは、検索、置換において正規表現を使うこともできる。

- VZSKTに必要なファイル：

  skt98.com, skt98.def, skt98.bat, ezkey.com, fntld.com, vwx.com,
  kh2vz.def[12], vz2kh.def, tm2vz.def, vz2tm.def, vzskt.key, vzrexp.doc.

[11] VZ Ver.1.5の機能中、我々にとって特に便利な機能を紹介する。
[12] kh2vz.defは、KH方式のテキストをVZSKT方式に変換するマクロである。

(2) 立ち上げ

SKT98.BAT によって VZSKT を立ち上げ、[13]KH 方式テキストを読み込んだ後、ESC キーを押し、続いて ^ キーを押してマクロメニューを呼び出し、kh2vz.def を選んでリターンキーを押せば、画面上で KH 方式は弁別記号付き文字に変換される。

(3) 入力

この状態で弁別記号付き文字を入力するには、NFER キーを押したあと、たとえば a と入力すれば、ā が入力され、画面に現れる。文字配列一覧は vzskt.key にある。

(4) 転写方式の転換

逆に VZSKT 方式を KH 方式に転換するには、vz2kh.def を用いる。要領は上記、kh2vz.def を使ったのと同じく、ESC+$\dot{}$キーによってマクロメニューを呼び出し、vz2kh.def にカーソルを合わせてリターンキーを押す。テクノメイトとの相互変換も可能（tm2vz.def, vz2tm.def による）。

(5) 終了

VZ を終了しただけでは VZSKT はメモリーから消去されない。DOS PROMPT 上で、SKT98 -z と入力し、リターンキーを押すと、「メモリーを解放しました」というメッセージとともに VZSKT は終了する。

(6) ただし VZSKT は、FD の VZ 立ち上げ機能や、[14]その他 VZ のマクロを使うもの（スペルチェッカー、n_tex.def など）とは併用できない。

6. バッチ・ファイルを使う

MS-DOS は拡張子が BAT というファイルの中の各行をコマンドとして解釈する。そしてそのコマンドを 1 行ごとに実行してゆく。

(1) 例えば各行に一つづつコマンドを書いたファイルを作り、これを ABC.BAT という名で保存する。こうして DOS PROMPT 状態で、

```
A > abc (return key)
```

とすれば書かれているコマンドを次々に実行する。従って複数のコマンドを纏めて実行する時は、いちいち 1 行づつ書くより遥かに便利である。

(2) リダイレクション

バッチファイルの中で生成したファイルは、適当なファイル名を付けて保存しなければならない。このときそのファイル名はリダイレクト記号の後に書いておけばよい。例えば

```
sed -f a1.sed b1.txt > c1.txt
```

と書けば、a1.sed というスクリプトファイルが、b1.txt というテキストファイルに対して実行され、その結果が c1.txt というファイルとして生成することになる。この時の>がリダイレクトの記号である。

(3) パイプ

上記の c1.txt に対してさらに a2.sed を実行するとき、

```
sed -f a1.sed b1.txt > c1.txt
```

[13]ただし FD から立ち上げるときは、SKT98.BAT を実行後、いったん FD を終了し、DOS PROMPT 状態で ESC キーを押すと、読み込みウィンドウが開く。
[14]前注参照。

```
sed -f a2.sed c1.txt > c2.txt
```

と2行に分けてバッチファイルに書いてもよいが、c1.txt が必要でないなら、1行に

```
sed -f a1.sed b1.txt | sed -f a2.sed > c.txt
```

と書くほうがよい(結果として生じる c.txt は c2.txt に等しい)。この記号 — をパイプと呼ぶ。パイプは前の作業の結果ファイルを次の作業の対象ファイルとして渡す役目をする。

(4) 引き数

特定のファイル名を記さずにバッチファイルを書いておくこともできる。例えば

```
sed -f a1.sed %1 | sed -f a2.sed > %2.txt
```

と書き、これを ABC.BAT と名付けておけば、

```
A > abc b1.txt c (return key)
```

によって先ほどと同様の結果を得ることができる。勿論同じ作業を別のテキストに課す時はテキスト名と、ベースネームを変えればよいのである。この記号% 1, % 2 などを引き数という。

[付録] Sanskrit 転写： KH 方式と OCR 方式

Sanskrit	KH 方式	OCR 方式	Sanskrit	KH 方式	OCR 方式
ā (â)	A (^A)*	aa (â)†	ṭ	T	F
ī (î)	I (^I)	ii (î)	ḍ	D	Z
ū (û)	U (^U)	uu (û)	ḷ	L	#
ṛ	R	&	ṇ	N	W
ṝ	q	q	ś	z	$ (š)
ḷ	W	!	ṣ	S	%
ṅ	G	Q	ṃ	M	@
ñ	J	~ (ñ)	ḥ	H	+

* 括弧内は ā と â を区別した場合の後者の転写.

†括弧内は OCR ソフト *Recognita Plus* の画面上における文字. 括弧に入っていないものは, SED などにおける用字.

平成６年度 文部省科学研究費 一般研究 (B) 研究成果報告書
「パーリ大蔵経のデータベース化による文献学的研究
— 自動読み取りシステムを用いて —」
（課題番号： 05451007）

執筆者
中谷 英明・江島 惠教

編集者
中谷　英明
〒651-21 神戸市西区伊川谷町有瀬
神戸学院大学 人文学部
Tel. (078) 974. 1551.