〔研究ノート〕

グリム兄弟の「伝説」および「メルヒェン」のテキスト・データベース化とその利用について

山田善久
中山淳子

1. はじめに

龍谷大学社会科学研究所の共同研究「グリム兄弟の『ドイツの伝説』に関する多角的研究」の一環として，伝説（Deutsche Sagen）およびメルヒェン（Kinder- und Hausmärchen）のテキスト・データベース化を試みた。伝説が585話555ページ，メルヒェン（以下KHMとする）が200話と10話からなるKinderlegendenを含めて計787ページ，磁気記憶媒体にして伝説が約1Mバイト，KHMが約1.5Mバイト，計約2.5Mバイトの分量である。KHMについては，1857年の決定版に拠っているが，これとさらに手稿版（1810年），第1版（1812年），第2版（1819年）に共通して収められている約40話についても，それぞれの版のデータ化を予定している（一部は完成）。これによって，伝説とKHMの語法の比較のみならず，KHMの各版間の比較といったことも可能になる。

ドイツ語文献のテキスト・データベースとしては，国内では九州大学の樋口忠治氏によるトーマス・マン・ファイルおよびゲーテ・ファイルがよく知られており，またドイツ本国では文学作品のみならず，多様なジャンルの文献を集めたいわゆるMannheimer Korpus I, IIや，最近では，聖書やゲーテ

全集などのテキスト・データベースがパソコン用の CD-ROM ないしはハード・ディスク版の形で市販されるようになっている。我々のデータベースは，これらに比べると分量的にささやかなものであるが，現在そして今後予想される機械可読テキストの普及に，一石を投じるものと考える。目下のところは，我々のプロジェクト内での私的な使用に限定しているが，今後，一般公開の可能性・方法について検討していくつもりである。

従来の活字メディアによる文献をテキスト・データベース化することの意義は改めて述べる必要もないであろう。機械可読化することで必要な用例を即座に検索したり，用語索引の作成，さらには文体研究に必要な統計的処理を加えるなど，さまざまな利用法が考えられる。もちろんこうした処理を可能にするためには，そのためのプログラムが必要である。我々の場合は，データベース作成と同時並行的に，TEDDY というプログラムを開発した。以下このプログラムの機能および利用例にふれながら，データベースの概要を紹介することにしたい。

2. データ・ファイルの形式

2.1 データの作成

データの作成は，ワープロやエディタを使って，基本的には手作業で行うわけであるが，我々の場合，一部について，バーズ情報科学研究所から発売されている PCR-SWAN というパソコン用の OCR（文字読み取りソフト）とハンディ・イメージスキャナ（オムロン HS7R）を援用した。KHM の約半分（およそ 400 ページ）はこれを使ってデータ化した。認識率，スピードとも本格的な OCR と比べると，パフォーマンスは落ちるであろうが，テキストの読み込み状態に合わせてうまくフォントの辞書登録をし，32 ビット機のような高速機を使用すれば，何とか実用になる。ただ，これは本格的な

OCR の場合も同様と思われるが，認識率 100% ということはまず有り得ないので，どうしてもデータを後で修正する必要があるが，これに案外時間がかかるというのが使用しての印象であった。ドイツ語スペル・チェッカーが援用できれば，この作業でも楽ができるであろう。

データの編集は，高電社から発売されている多国語ワープロ KOA-Techno Mate を使用した。これはドイツ語文字が画面表示できる数少ないワープロとしてゲルマニストの間ではよく知られているソフトであるが，欧文禁則処理や WYSIWYG（What You See Is What You Get）が実現できる点は，やはり魅力である。しかしこれはあくまでもワープロであり，その多機能のためにプログラムが重くなっており，エディタなどに比べるとレスポンスの点でやはり多少見劣りするし，約 60 K バイトという編集サイズの制限も気になる点である。テキスト・データ入力用のツールとしての理想を言えば，エディタ並みの軽快さと編集能力を持った上で，上記欧文禁則処理（ワード・ラップ）と WYSIWYG が実現でき，さらに各国語のスペル・チェックにもオプションで対応できるもの，ということになろうか。

2.2 ドイツ語特殊文字の取扱い

我々のデータベースは，パソコン上での利用を想定しているので，アスキー形式の MS-DOS テキスト・ファイルの形にしている。この場合問題なのは標準アスキー・コード（00_H～$7F_H$）に含まれないウムラウトやエスツェットなどドイツ語の特殊文字をどう扱うかである。

現在，ドイツ語など欧米語の特殊文字を処理するコードとして共通化されているものに，二系統の体系がある[1)]。一つは ISO（国際標準化機構）の規格による 7 ビット・コード体系であり，この体系では標準アスキー・コードのうちアルファベット以外の比較的使用頻度の低い文字コード 12 字を，各国用の文字に切り替えられるようになっている。これは NEC の PC-9800 シリーズ機では，プリンタ部（PC-PR 系）で対応されている。もう一つは海

外IBM機で採用されており，いわば国際標準になっているもので，拡張アスキー・コード（80_H～FF_H）を利用した8ビット・コード体系である。この体系では利用できるコードが128文字あり，アクセント符号付き文字の多いフランス語やギリシア文字にも対応している。海外で市販されているパソコン用のテキスト・データベースは，もちろんこのコード体系に依拠している。

我々のデータは，基本的には国内での使用ということで，前者の7ビット・コード体系を採用している。8ビット・コード体系への変換は，ストリーム・エディタSEDのようなパターン置換ツールを使ったり，また自前でも簡単なプログラムで対応することができるので問題はない。この7ビット・コード体系でのドイツ語特殊文字と標準アスキー文字の対応は次の通りである。

Ä=[，Ö=¥，Ü=]，ä={，ö=¦，ü=}，ß=~

先にふれたように，この体系の場合，特殊文字の表示が，プリンタ印字にしか対応していないという問題があるが，ディスプレイ画面表示は，後述するように，自前のプログラム TEDDY で対応している。

2.3 アポストロフィとシングル・クォーテーションマーク

アポストロフィおよびシングル・クォーテーションマークは，キーボードの文字数の制限のため，同一のキー［'］（27_H）を流用して文書を作成している場合が多いと思われる。ワープロの文書や用例検索のみを目的にしたデータベースならばこれでもよいのであるが，文字列を単語として切り出すような処理を行う場合，これらははっきり区別されなければならない。つまり前者は単語中の省略記号として有意の文字であり，後者は punctuation の記号として切り出しの段階で捨て去るべきものである。グリム・データベースでは，原典の引用部分がシングル・クォーテーションマークになっているため，この部分は［'］（27_H）を使用している。一方本来のアポストロ

フィ記号には，この文字は使えないので，変則的な扱いであるが，［゛］（60_H）を使用して重複を避けている。ちなみに文字の設定は，TEDDY 側でデータの状態に合わせて変更できるようになっているので，この設定でなければ処理できないというわけではない。

2.4 データの実例とファイルの編成

処理例1は，ファイルに格納されているデータの一部（KHM 26「赤ずきん（Rotkäppchen）」の冒頭）を TEDDY の印刷機能を使ってハード・コピーしたものである。ここでは行末で禁則処理（ワード・ラップ）が施されているが，もちろん解除して出力することもできる。一見してわかるように，データには，例えば九州大学のトーマス・マン・ファイルにあるようなレコード番号がなく，いわばワープロ文書のイメージに近いものである。これは入力時の容易さ，処理の汎用性を考慮した結果である。検索処理の場合，検索と同時にその出現位置を明示する必要があるが，我々のデータ・ファイルでは，ページ区切りを識別させる制御記号［$］を，原典のページ末尾に対応する箇所に付加しておくことで，ページ単位の出現位置を表示できるようになっている。実用上は，これで十分ではないかと考えている。データ中には，この他にも，機械処理を行うために不可欠な若干の制御記号が書き込まれているが，これらの詳細は4章で説明することにしたい。

データ・ファイルは，各話ごとにファイル化している。ファイル名は，KHM の場合，M001.TXT から M200.TXT であるが，一部方言で書かれている話は識別のため，拡張子を .MND にしている。また KHM の末尾に収録されている Kinderlegenden は，L001.TXT から L010.TXT となっている。伝説の場合は，S001.TXT から S585.TXT まで 585 個のファイルに分けている。処理にあたっては，単一のファイルの処理のみならず，これらのファイルを一括して処理する必要があり，その工夫を考えなければならないが，TEDDY では，？にファイル名の数字部分にマッチする一種のワ

処理例１　データ・ファイルの入力例

```
%$174%
% Rotk{ppchen %
 *Es war einmal eine kleine s}~e Dirne, die hatte jedermann lieb, der
sie nur ansah, am allerliebsten aber ihre Gro~mutter, die wu~te gar
nicht, was sie alles dem Kinde geben sollte. *Einmal schenkte sie ihm
ein K{ppchen von rotem Sammet, und weil ihm das so wohl stand und es
nichts anders mehr tragen wollte, hie~ es nur das Rotk{ppchen. *Eines
Tages sprach seine Mutter zu ihm 'komm, Rotk{ppchen, da hast du ein
St}ck Kuchen und eine Flasche Wein, +bring das der Gro~mutter hinaus+;$
sie ist krank und schwach und wird sich daran laben. +*Mach dich auf+,
bevor es hei~ wird, und wenn du hinauskommst, so geh h}bsch sittsam und
lauf nicht vom Weg ab, sonst f{llst du und zerbrichst das Glas, und die
Gro~mutter hat nichts. *Und wenn du in ihre Stube kommst, so vergi~
nicht, guten Morgen zu sagen, und +guck nicht erst in alle Ecken
herum+.' '*Ich will schon alles gut machen,' sagte Rotk{ppchen zur
Mutter, und gab ihr die Hand darauf. *Die Gro~mutter aber wohnte
drau~en im Wald, eine halbe Stunde vom Dorf. *Wie nun Rotk{ppchen in
den Wald kam, begegnete ihm der Wolf. Rotk{ppchen aber wu~te nicht, was
das f}r ein b¦ses Tier war, und f}rchtete sich nicht vor ihm. '*Guten
Tag, Rotk{ppchen', sprach er. '*Sch¦nen Dank, Wolf.' '*Wo hinaus so
fr}h, Rotk{ppchen?' '*Zur Gro~mutter.' '*Was tr{gst du unter der
Sch}rze?' 'Kuchen und Wein: gestern haben wir gebacken, da soll sich
die kranke und schwache Gro~mutter etwas zu$gut tun und sich damit
st{rken.' 'Rotk{ppchen, wo wohnt deine Gro~mutter?' '*Noch eine gute
Viertelstunde weiter im Wald, unter den drei gro~en Eichb{umen, da
steht ihr Haus, unten sind die Nu~hecken, das wirst du ja wissen,'
sagte Rotk{ppchen. *Der Wolf dachte bei sich 'das junge zarte Ding, das
ist ein fetter Bissen, der wird noch besser schmecken als die Alte: du
mu~t es listig anfangen, damit du beide erschnappst.' /*Da +ging er ein
Weilchen neben Rotk{ppchen her+, dann sprach er 'Rotk{ppchen, sieh
einmal die sch¦nen Blumen, die ringsumher stehen, warum +guckst du dich
nicht um+? ich glaube, du h¦rst gar nicht, wie die V¦glein so lieblich
singen? du +gehst ja f}r dich hin+, als wenn du zur Schule gingst, und
ist so lustig hau~en in dem Wald'./$
 Rotk{ppchen +schlug die Augen auf+, und als es sah, wie die
Sonnenstrahlen durch die B{ume hin- und her++tanzten und alles voll
sch¦ner Blumen stand, dachte es 'wenn ich der Gro~mutter einen frischen
Strau~ mitbringe, der wird ihr auch Freude machen; es ist so fr}h am
Tag, da~ ich doch zu rechter Zeit ankomme,' +lief vom Wege ab in den
Wald hinein+ und suchte Blumen. *Und wenn es eine gebrochen hatte,
meinte es, weiter hinaus st{nde eine sch¦nere, und lief darnach, und
+geriet immer tiefer in den Wald hinein+. *Der Wolf aber ging
geradeswegs nach dem Haus der Gro~mutter, und klopfte an die T}re.
'*Wer ist drau~en?' 'Rotk{ppchen, das bringt Kuchen und Wein, +mach
auf+.' '*Dr}ck nur auf die Klinke,' rief die Gro~mutter, 'ich bin zu
schwach und kann nicht aufstehen.' *Der Wolf dr}ckte auf die Klinke,
die T}re +sprang auf+ und er ging, ohne ein Wort zu sprechen, gerade
zum Bett der Gro~mutter und verschluckte sie. *Dann +tat er ihre
Kleider an+, +setzte ihre Haube auf+, legte sich in ihr Bett und +zog
die Vorh{nge vor+.
 Rotk{ppchen aber war nach den Blumen herumgelaufen, und als es so viel
zusammen hatte, da~ es keine mehr tragen konnte, +fiel ihm die
Gro~mutter wieder ein+, und es machte sich auf den Weg zu ihr. *Es
wunderte sich, da~ die T}re aufstand, und$ wie es in die Stube trat, so
+kam es ihm so seltsam darin vor+, da~ es dachte 'ei, du mein Gott, wie
{ngstlich wird mirs heute zumut, und ich bin sonst so gerne bei der
```

イルド・カードの機能を持たせ，複数個のファイルを連続処理できるようにしている。例えば，ファイルの指定の際に，

M???.TXT

とし，その下限と上限を

from 1 to 200

のように指定すると，プログラムがその間を自動的にインクリメントしていき，この場合は，KHM の方言で書かれたものを除く M001.TXT から M200.TXT までのファイルが一括処理される。

3. TEDDY を利用した処理

3.1 TEDDY の概要

TEDDY の機能は，大別すると次の二つである。

1) 語彙統計（ワード・リストおよび頻度集計表の作成）
2) 用例検索（センテンス単位およびパラグラフ単位）

これは，よく知られているオックスフォード大学のコンコーダンス・プログラム OCP (Oxford Concordance Program)[2]のワード・リスト（語彙一覧表）作成の機能に用例の検索機能を付加したものと考えていただければよい。現在のところ対応機種は NEC の PC-9800 シリーズ（MS-DOS）のみであるが，今後 IBM 機など他の機種への移植を予定している。現行版（Ver 1.10）の開発言語は N88BASIC コンパイラであるが，これも今後，より高速で汎用性の高い処理系に移行する予定である。

N88BASIC を使用したのは，言語仕様がわかりやすくプログラミングが容易であることもあったが，大きな問題は，半角ドイツ語特殊文字のディスプレイ画面表示の容易さである。2.2で少しふれたように，国内製のパソコンは，JIS規格に準拠しており，仕様のままではこうした半角文字を利用することができないので，ユーザー側で半角外字を利用するための何らかの仕様変更を行わなければならないが，N88BASIC はこれが簡単な仕掛で実現できる。この詳細については山田（1990）で述べたので，そちらを参照していただきたい。

グリム•データベースのファイル容量は，冒頭で述べたように，伝説が約1Mバイト，KHMが約1.5Mバイトである。こうした大量のデータを一括処理させる場合，ソフトウェアの処理能力が問題になる。とくに語彙統計処理のようにソーティングが必要な場合は，メモリ上で操作できる配列には限りがあるので，何かうまい工夫を考える必要がある。TEDDY ではこれを次のように解決している。処理の流れを1)～4)に分けて，ポイントを説明しておく。

1) 出力ファイルのオープン

データ•ファイルから切り出した単語を書き出すため，12個の出力ファイルをオープンする。これらのファイルは，アルファベットの種類に応じて区分されている。区分は出現する語形の TYPE 数（3.2.1参照）に基づき，経験的に割り出した。例えば比較的 TYPE 数の多い a(A) や v(V) は，これだけで1ファイル，TYPE 数の少ないところでは，o(O), p(P), q(Q), r(R) や w(W), x(X), y(Y), z(Z) の場合，4文字で1ファイルとなっている。この12個というファイル数は，使用した処理系の制限によるもので，N88BASIC（MS-DOS）では同時オープン可能な最大ファイル数は13個である（1個は入力ファイル用に使用する）。

2） 単語の切り出し，ソート & カウント

データ・ファイルから単語を切り出し，これにソートおよび頻度カウントの処理を行い，1)でオープンしたそれぞれの出力ファイルに書き出す。小さなファイルであれば，これで処理は済むが，入力データ・ファイルのサイズが大きく，切り出した単語の文字数が処理系の文字列領域の限度（約 64 K バイト，約 8000 語程度）に達した場合，ここで一時切り出し処理を中断し，上のソート & カウント，ファイル書き出し処理を行ってから，再度切り出し処理を続行する。文字列領域の上限に達する度に，以上の作業を繰り返す。この間，出力ファイルは APPEND モードで，新しく処理したデータは前のデータに追加して書き出される。

3） 出力ファイルのマージ

2)の処理の間に書き出された出力ファイルのどれかのサイズがメモリ上で扱える限度（約 64 K バイト）に達したら，そこで処理を一時中断し，そのファイルにマージ処理を加え，ファイル・サイズを小さくしてからもとの処理に戻る。2)と同様に，ファイル・サイズが上限に達する度に，この作業を繰り返す。実際のデータを扱っていると，接続詞の und や冠詞類など頻出する単語があり，このマージ処理によってファイル・サイズを劇的に減らすことができ，それだけ処理可能なデータ量を増やすことができる。

4） リンク・ファイルの作成

全データ・ファイルについて以上の処理を行った後，12 個の出力ファイルを結合して，最終的にソートされ各頻度が書き出された単語リストのファイルが作成される。この場合，アルファベットの大文字と小文字を区別しない形でリストを編成したり，代用文字 [，¥，]，{，¦，}，~ を使用しているドイツ語特殊文字を辞書配列と同じように編成する必要があり，単純にファイルを結合するのではなく，ここでも再ソートの処理が必要である。この

最終的に作成されたファイル（リンク・ファイルと呼んでいる）を元にして，さまざまな語彙統計の処理が実行される。またこのファイルは，いわゆるCSV形式[3]になっており，他の市販データベース・ソフトでも読み込めるように配慮されている。

以上，多少煩雑な説明になってしまったが，処理自体は自動化されており，ユーザー自身がとくに手を煩わす必要はないようにしている。こうした処理の工夫により，語彙統計では，約2Mバイト程度のデータが一括処理できるようになった。我々のデータベースでは，伝説，KHMのそれぞれについて一括処理が可能であり，実用上は十分と言えるのではなかろうか。なお，TEDDYのもう一つの機能である用例検索の場合は，大量の配列を必要とするソーティングのような処理はしないので，扱えるデータ量に制限はなく，いわばディスク容量に依存するということになる。

以下，TEDDYによる処理例を，語彙統計，用例検索に分けて紹介していくことにする。

3.2 語彙統計

3.2.1 ワード・リスト

処理例2から**4**は，ワード・リストの処理例である。例に見られるようにワード・リスト処理では，テキスト中の全語彙を集計して，語彙の使用状況をリストアップすることができる。出力形式は次の三つである。

- 正順ワード・リスト（**処理例2**）
  ……アルファベット順のリスト
- 逆順ワード・リスト（**処理例3**）
  ……逆引き辞典の配列に見られるような語尾からのアルファベット順リスト

・頻度順ワード・リスト（**処理例 4**）

……頻度の多い単語から少ない単語へと並べたリスト

処理例では省略されているが，各リストの末尾には，語彙の TOKEN（テキスト中の単語の延べ総数）および TYPE（重複したものをカウントしない種類としての単語の総数）が出力される。

ワード・リストの作成にあたって，精確なデータを得るためには，少なくとも次の三点を考慮することが必要である。

1） 文頭の大文字を小文字に変換する処理
2） 分離動詞およびハイフンの処理
3） 人称変化形や格変化形を原形に戻す処理

1)の大文字→小文字変換に関しては，TEDDY では，変換を指示する制御文字をあらかじめデータ中に書き込んでおき，これをプログラムで処理するようにしている。この制御文字としてはアスタリスク［＊］を用い，これを変換すべき文頭の大文字の直前に付加している。制御文字を使用せず，この処理をプログラム側で自動化する方法も考えられるが，ドイツ語の場合は，本来大文字で始まる名詞が文頭に立つケースも多く，名詞か否かを識別させるために，大規模な辞書が必要である。また名詞の語彙の総数は固有名詞を加えると無限大ということになり，とても実用になるものではない。

2)の分離動詞およびハイフンの処理というのは，テキスト中で分離しているドイツ語の分離動詞を一語として処理したり，また der Ein- und Ausgang のようにハイフンで結ばれている語句から，Eingang という文字列を取り出したりする処理のことである。これらの処理のために TEDDY では，制御文字（初期値は＋）を使用している。こうした処理は目的によっては必要ない場合もあろうが，必要な場合は次の手順でデータを作成する。

処理例２　正順ワード・リスト

ALPHABEICAL ORDER
FILE NAME= M026.TXT

NO.	WORD	COUNT
1	ab	3
2	abends	1
3	aber	15
4	ableiten	1
5	abzog	1
6	ach	1
7	alle	2
8	allein	1
9	allerliebsten	1
10	alles	3
11	als	5
12	alte	2
13	Alte	1
14	alten	1
15	alter	1
16	am	2
17	an	2
18	anderer	1
19	anders	1
20	anfangen	1
21	anfing	3
22	anklopfte	1
23	ankomme	1
24	anlegen	1
25	ansah	1
26	antat	1
27	Antwort	1
28	arme	1
29	aß	1
30	atmen	1
31	auch	3
32	auf	4
33	aufmach	3
34	aufmachten	1
35	aufs	1
36	aufschlug	1
37	aufschneiden	1
38	aufsetzte	1
39	aufsprang	1
40	aufstand	1
41	aufstehen	1
42	aufwachte	1
43	Augen	3
44	aus	2
45	aussah	1
46	bald	1
47	Bauch	1
48	Bäume	1
49	begegnet	1
50	begegnete	1
51	bei	2

処理例3　逆順ワード・リスト

REVERSE ALPHABETICAL ORDER
FILE NAME= M026.TXT

NO.	WORD	COUNT
1	da	14
2	ja	2
3	ab	3
4	gab	1
5	hab	1
6	lieb	1
7	Leib	2
8	ob	1
9	zuleid	1
10	bald	1
11	Wald	7
12	Hand	1
13	niemand	1
14	stand	3
15	aufstand	1
16	Kind	1
17	sind	2
18	geschwind	1
19	und	63
20	wird	6
21	habe	2
22	halbe	1
23	Haube	2
24	glaube	1
25	Stube	3
26	gerade	3
27	beide	1
28	Hände	1
29	stände	1
30	finde	1
31	Kinde	1
32	Stunde	1
33	Viertelstunde	1
34	Freude	1
35	Wege	3
36	Vorhänge	2
37	lange	2
38	ginge	1
39	mitbringe	1
40	junge	1
41	schwache	1
42	Flasche	1
43	die	38
44	sie	10
45	wie	12
46	kranke	1
47	Klinke	2
48	alle	2
49	Schule	1
50	ankomme	1
51	arme	1

処理例4　頻度順ワード・リスト

FREQUENCY ORDER
FILE NAME= M026.TXT

NO.	WORD	COUNT
1	und	63
2	die	38
3	der	27
4	Großmutter	26
5	er	25
6	Rotkäppchen	25
7	es	22
8	du	20
9	so	20
10	den	19
11	das	17
12	Wolf	17
13	ich	16
14	aber	15
15	in	15
16	nicht	15
17	da	14
18	daß	14
19	dem	13
20	sich	13
21	ihm	12
22	wie	12
23	ein	11
24	zu	11
25	sie	10
26	dich	9
27	hatte	9
28	was	8
29	eine	7
30	Wald	7
31	für	6
32	große	6
33	Haus	6
34	ihr	6
35	kann	6
36	vom	6
37	wenn	6
38	wird	6
39	als	5
40	besser	5
41	dachte	5
42	ging	5
43	hast	5
44	ist	5
45	noch	5
46	sagte	5
47	Türe	5
48	war	5
49	wieder	5
50	wollte	5
51	auf	4

A. ドイツ語の分離動詞

（例）＊Ich +stehe sehr früh auf+.

＊Ich bitte Sie, ihn im Büro an+zu+rufen.

（文頭の＊は大文字→小文字変換記号）

B. ハイフンでつながれている語句

（例） der Ein- und Aus++gang

hin- und her++tanzen

A の上の方の例では，文中で分離している二つの要素のうち，基礎動詞の方には直前に＋記号を，前綴の方には直後に＋記号を付加し，単語切り出し処理の段階でこの＋記号を目印に二つの要素を合成する。下の zu 不定詞の場合は，例えば an+zu+rufen から zu と anrufen を取り出す。この場合例えば，分離動詞の辞書を作成して処理を自動化する方法が考えられる。名詞の場合と比べると語彙に限りがあるため，よほど可能性のある方法であるが，やはり辞書の作成が大変な作業になるため，現時点では導入していない。仮に辞書が作成できたとしても，実際のテキストにあたってみると，辞書ではカバーし切れない微妙な場合も出て来ると思われるので，手作業がやはり一番安全で確実な方法であろう。B のハイフンでつながれている語句の場合は，後の合成語の切れ目に ++ を挿入して，これを目印に Eingang と Ausgang, あるいは hintanzen と hertanzen という文字列を取り出すようにしている。

ところで分離動詞にはもう一つ面倒な問題がある。それは分離動詞をどのような基準で識別すればよいかという問題である。我々のグリム•データベースでは，その内容の性格上，上の hin や her あるいはこれに前置詞等が結びついた hinab, heran, hinein, herauf など，さらには fort, weiter, heim といった方向規定詞が動詞の意味を補強する機能をもって使われている例が頻出する。精密な文法的議論はここでは省くとしても，果たしてこれ

らを分離前綴として扱うべきかは，検討の余地があると思われる。一例として heim を取り上げてみよう。

(a) Als sie *heim* kam, fragte die Maus ‚wie ist denn dieses Kind getauft worden?' (M002.TXT)

(b) Der König nahm es auf seinen Arm, trug es auf sein Pferd und ritt mit ihm *heim*, ... (M003.TXT)

(c) Darauf läutete er die Glocke, ging *heim*, legte sich, ohne ein Wort zu sagen, ins Bett und schlief fort. (M004.TXT)

(d) ‚macht mir auf, Kinder, euer liebes Mütterchen ist *heim*gekommen und hat jeden von euch etwas aus dem Wald mitgebracht.' (M005.TXT)

(e) Nicht lange danach kam die alte Geiß aus dem Wald wieder *heim*. (M005.TXT)

(f) Nachdem sie das letzte betrachtet hatte, dankte sie dem Kaufmann und wollte *heim*, ... (M006.TXT)

(g) ‚ei, da führt er die Königstochter vom goldenen Dache *heim*,' (M006.TXT)

上の(a)から(g)の heim を含む例のうち，(d)は過去分詞形であるが，明らかに分離前綴として使われているのに対し，(a)の副文中の heim は定形から分離して書かれているので，独立した副詞として用いられていることになる。他の(b)(c)(e)(f)(g)は主文中に使われている例であるが，(a)と(d)の表記上のゆれを考えると，一義的に分離前綴または独立副詞のどちらかに分類することは困難であろうと思われる。(f)の例は定形が話法の助動詞 wollen であり，普通の解釈は助動詞の本動詞的用法で heim は独立副詞ということになるのであろうが，heimwollen という見出し語が国内の独和辞典（例え

ば木村・相良『独和辞典』）にも収められており，問題がますますややこしくなってくる。

一つの解決策としては，このような言語の事実そのものではなく解釈にかかわるような部分には手をつけないという行き方が考えられるであろう。つまり人為的にテキストに識別記号を付けて分離動詞を取り出すのではなく，原典の状態をそのまま生かして機械的に単語をリスト・アップする方法である。もちろんこれではデータとして不精確ということもあろうが，その部分について精確なデータが必要な場合は，後述する用例検索の機能などを利用し，用例を逐一ピックアップしてデータの再吟味を行えばよい。要はユーザーの利用目的に応じた使い方の工夫が現実的ではないかと思われる。この点を考えて TEDDY では，この分離動詞・ハイフン処理の機能を解除することもできるようにしている。

最後に 3)の問題について少しだけふれておく。人称変化形や格変化形のような変化語形を原形に戻した形でワード・リストを作成する機能（lemmatisation）は，例えば Butler（1985）の指摘[4]を待つまでもなく，不可欠とも言える重要な機能である。しかしこれも出現語形をすべてマークするような大規模な辞書が必要であるなど技術的になかなか難しく，TEDDY では対応していない。今後の検討課題の一つである。現実的な解決策としては，ワード・リスト処理したデータを，他のリレーショナル・データベース（例えば dBASE III）に読み込み，その中で原形のフィールドを追加して個々のレコードに原形情報を手作業で書き込んだ上，これをキー・インデックスとして処理させるということが考えられる。いずれにせよ計算機は万能ではなく，まだまだ人手を煩わす仕事が多くあるということも事実である。

3.2.2 語彙統計表

OCP には，STATS というコマンドがあって，語彙の度数分布などの統計を出力することができる。TEDDY でも，これとほぼ同等の処理を実現

することができる。**処理例5**がそれである。この種の表は，一つのテキストについてだけ作成しても，あまり有効な使い道はなさそうであるが，複数のテキストのデータを比較すると，いろいろとおもしろい語彙の傾向がわかってくる。例えば頻度1の単語の比率の高いテキストは，語彙のバラエティが豊富であること，逆に低いテキストはバラエティに乏しく，場合によっては単調あるいは平易といった傾向を推測することが可能であろう。

表中の TYPE/TOKEN RATIO というのは，先ほどふれた語彙の TYPE を語彙の TOKEN で割った数値であるが，この数値を比較した場合も上と同じような傾向を読み取ることができる。つまりこの数値が高いほど語彙のバラエティが豊富であることになる。この最大値は1.0であるが，この場合は，実際には有り得ないことであるが，テキストに現れる単語がすべて異な

処理例5　語 彙 統 計 表

STATISTICS
FILE NAME= M026.TXT

頻度(A)	度数(B)	(A*B)	語彙数	単語数	語彙%	単語%	(A*B)%
1	312	312	312	312	65.82	25.45	25.45
2	74	148	386	460	81.43	37.52	12.07
3	24	72	410	532	86.50	43.39	5.87
4	14	56	424	588	89.45	47.96	4.57
5	12	60	436	648	91.98	52.85	4.89
6	8	48	444	696	93.67	56.77	3.92
7	2	14	446	710	94.09	57.91	1.14
8	1	8	447	718	94.30	58.56	0.65
9	2	18	449	736	94.73	60.03	1.47
10	1	10	450	746	94.94	60.85	0.82
11	2	22	452	768	95.36	62.64	1.79
12	2	24	454	792	95.78	64.60	1.96
13	2	26	456	818	96.20	66.72	2.12
14	2	28	458	846	96.62	69.00	2.28
15	3	45	461	891	97.26	72.68	3.67
16	1	16	462	907	97.47	73.98	1.31
17	2	34	464	941	97.89	76.75	2.77
19	1	19	465	960	98.10	78.30	1.55
20	2	40	467	1000	98.52	81.57	3.26
22	1	22	468	1022	98.73	83.36	1.79
25	2	50	470	1072	99.16	87.44	4.08
26	1	26	471	1098	99.37	89.56	2.12
27	1	27	472	1125	99.58	91.76	2.20
38	1	38	473	1163	99.79	94.86	3.10
63	1	63	474	1226	100.00	100.00	5.14

TYPE=　474　TOKEN=　1226　TYPE/TOKEN RATIO= 0.387

っていることを意味する。

参考までに，**処理例 6～10** に，「いばら姫（Dornröschen）」の手稿版（1810），初版（1812），第 2 版（1819），決定版（1857）のそれぞれのデータをあげておく。テキストの長さを TOKEN の数で計ると，順に 395，844，1097，1216 で版を追うごとに長くなっていることがわかる。また各データの TYPE/TOKEN RATIO を比較すると，順に 0.565，0.405，0.420，0.401 となっている。一般に TYPE/TOKEN RATIO の数値は，テキストが長くなるほど低くなることが知られている[5]が，上の各数値は漸減ではなく，デコボコがある点で変わった傾向を示していると言える。この傾向をどう解釈するかは観察者に任せるべき問題である。一つの解釈を示すと，二番目の，つまり初版（1812）の数値 0.405 が，漸減という一般的な傾向から見ると，落ち込んでおり，一種の谷間を示していると言えると思う。一般的な傾向としては，この数値がその前後の値 0.565 と 0.420 の間に収まるはずである。つまり初版が，他の版（とくに後の第 2 版および決定版）と比べて，変わった文体の傾向を示しているということである。どのように変わっているかは，他のデータも踏まえて総合的に検討しなければならない問題であるが，大ざっぱには次のように考えることができよう。先ほど TYPE/TOKEN RATIO

処理例 6　語彙統計表（「いばら姫」手稿版（1810））

```
STATISTICS
FILE NAME= M019_10.TXT
```

頻度(A)	度数(B)	(A*B)	語彙数	単語数	語彙%	単語%	(A*B)%
1	160	160	160	160	71.75	40.51	40.51
2	31	62	191	222	85.65	56.20	15.70
3	10	30	201	252	90.13	63.80	7.59
4	8	32	209	284	93.72	71.90	8.10
5	5	25	214	309	95.96	78.23	6.33
6	1	6	215	315	96.41	79.75	1.52
7	4	28	219	343	98.21	86.84	7.09
10	1	10	220	353	98.65	89.37	2.53
12	2	24	222	377	99.55	95.44	6.08
18	1	18	223	395	100.00	100.00	4.56

TYPE= 223　TOKEN= 395　TYPE/TOKEN RATIO= 0.565

処理例 7　語彙統計表（「いばら姫」初版（1812））

STATISTICS
FILE NAME= M050_12.TXT

頻度(A)	度数(B)	(A*B)	語彙数	単語数	語彙%	単語%	(A*B)%
1	205	205	205	205	59.94	24.29	24.29
2	72	144	277	349	80.99	41.35	17.06
3	24	72	301	421	88.01	49.88	8.53
4	12	48	313	469	91.52	55.57	5.69
5	5	25	318	494	92.98	58.53	2.96
6	1	6	319	500	93.27	59.24	0.71
7	6	42	325	542	95.03	64.22	4.98
9	5	45	330	587	96.49	69.55	5.33
10	1	10	331	597	96.78	70.73	1.18
11	1	11	332	608	97.08	72.04	1.30
12	2	24	334	632	97.66	74.88	2.84
13	1	13	335	645	97.95	76.42	1.54
14	1	14	336	659	98.25	78.08	1.66
15	2	30	338	689	98.83	81.64	3.55
21	1	21	339	710	99.12	84.12	2.49
23	1	23	340	733	99.42	86.85	2.73
41	1	41	341	774	99.71	91.71	4.86
70	1	70	342	844	100.00	100.00	8.29

TYPE= 342　TOKEN= 844　TYPE/TOKEN RATIO= 0.405

処理例 8　語彙統計表（「いばら姫」第 2 版（1819））

STATISTICS
FILE NAME= M050_19.TXT

頻度(A)	度数(B)	(A*B)	語彙数	単語数	語彙%	単語%	(A*B)%
1	309	309	309	309	67.03	28.17	28.17
2	78	156	387	465	83.95	42.39	14.22
3	24	72	411	537	89.15	48.95	6.56
4	11	44	422	581	91.54	52.96	4.01
5	3	15	425	596	92.19	54.33	1.37
6	6	36	431	632	93.49	57.61	3.28
7	7	49	438	681	95.01	62.08	4.47
8	2	16	440	697	95.44	63.54	1.46
9	1	9	441	706	95.66	64.36	0.82
10	2	20	443	726	96.10	66.18	1.82
11	2	22	445	748	96.53	68.19	2.01
12	2	24	447	772	96.96	70.37	2.19
13	5	65	452	837	98.05	76.30	5.93
14	1	14	453	851	98.26	77.58	1.28
16	2	32	455	883	98.70	80.49	2.92
20	2	40	457	923	99.13	84.14	3.65
22	1	22	458	945	99.35	86.14	2.01
27	1	27	459	972	99.57	88.61	2.46
53	1	53	460	1025	99.78	93.44	4.83
72	1	72	461	1097	100.00	100.00	6.56

TYPE= 461　TOKEN= 1097　TYPE/TOKEN RATIO= 0.420

処理例 9　語彙統計表（「いばら姫」決定版（1857））

STATISTICS
FILE NAME= M050.TXT

頻度(A)	度数(B)	(A*B)	語彙数	単語数	語彙ﾊﾟｰｾﾝﾄ	単語ﾊﾟｰｾﾝﾄ	(A*B)ﾊﾟｰｾﾝﾄ
1	312	312	312	312	63.93	25.66	25.66
2	86	172	398	484	81.56	39.80	14.14
3	38	114	436	598	89.34	49.18	9.38
4	12	48	448	646	91.80	53.13	3.95
5	7	35	455	681	93.24	56.00	2.88
6	1	6	456	687	93.44	56.50	0.49
7	4	28	460	715	94.26	58.80	2.30
8	4	32	464	747	95.08	61.43	2.63
9	2	18	466	765	95.49	62.91	1.48
10	3	30	469	795	96.11	65.38	2.47
11	1	11	470	806	96.31	66.28	0.90
12	1	12	471	818	96.52	67.27	0.99
13	4	52	475	870	97.34	71.55	4.28
14	1	14	476	884	97.54	72.70	1.15
15	1	15	477	899	97.75	73.93	1.23
16	2	32	479	931	98.16	76.56	2.63
17	2	34	481	965	98.57	79.36	2.80
18	2	36	483	1001	98.98	82.32	2.96
22	1	22	484	1023	99.18	84.13	1.81
24	1	24	485	1047	99.39	86.10	1.97
34	1	34	486	1081	99.59	88.90	2.80
54	1	54	487	1135	99.80	93.34	4.44
81	1	81	488	1216	100.00	100.00	6.66

TYPE= 488　TOKEN= 1216　TYPE/TOKEN RATIO= 0.401

の値が高いほど語彙のバラエティが豊富であることを述べた。初版でこの値が落ち込んでいる，すなわち基準より低いということは，これとは逆に，相対的に語彙のバラエティに乏しいということを意味する。つまり初版では，使われている語彙の種類が少なく，その意味では，あまり色彩感のない書き方になっていると考えてよい。よく知られているように，初版は文体的な技巧をあまり凝らさず，オリジナルの語りの名残りをとどめたいわば素朴な文体であるのに対し，第2版以降は，もっぱら弟のヴィルヘルムが絶えず手を加え，「読むメルヒェン」として洗練させてきた。上の数値は，この事実を数字の面から裏付ける一つの例と考えられるのではないかと思う。

3.3 用例検索

用例検索では，指定したキーワードをテキスト中から検索し，その出現箇所をセンテンス単位またはパラグラフ単位で出力させる。センテンス単位の場合，ピリオド［.］，疑問符［?］，感嘆符［!］をレコード区切り記号としているが，ユーザー独自の区切り記号を新たに加えることもできる。パラグラフ単位では，改行コードを区切りとする。**処理例 10** はセンテンス単位で，**処理例 11** はパラグラフ単位で検索した一例である。MS-DOS の FIND コマンドや各種 GREP のようなパターン検索ツールと違うところは，単にパターン•マッチしたレコードを抜き出し，数えるだけでなく，指定したキーワードが一つのレコードに複数個ある場合，そのすべてを拾い出し，キーワードの出現総数をカウントできる点である。これは単なる用例収集にとどまらず，語彙調査を行う場合，データ中の全該当箇所とその文脈，総数を出力させる必要があると考えたためである。

キーワードの指定は，単語レベル（キーワードの両端が空白か句読記号）の他，両端にハイフン［-］を付けることによって形態素レベル（接辞や語中文字列）の検索を指定することができる。すなわち er という文字列を例にとると，

er　（単語の検索）
er-　（単語＋接頭辞の検索）
-er　（単語＋接尾辞の検索）
-er-　（単語＋接辞＋語中文字列の検索）

のような指定が可能である。キーワードは，一度に 10 個まで指定することができる。複数のキーワードを指定する場合は，**処理例 11** に見られるように各キーワード間をコンマ［,］でつなぐ。

処理例10　用 例 検 索（センテンス単位）

```
RESULTS OF KEYWORD SEARCH
FILENAME= M026.TXT

KEYWORD= Wald
================================================================================
--------------------------------------------------------------------------------
    1 Die Großmutter aber wohnte draußen im Wald, eine halbe Stunde
      vom Dorf.
      (Page 175)
--------------------------------------------------------------------------------
    2 Wie nun Rotkäppchen in den Wald kam, begegnete ihm der Wolf.
      (Page 175)
--------------------------------------------------------------------------------
    3 'Noch eine gute Viertelstunde weiter im Wald, unter den drei
      großen Eichbäumen, da steht ihr Haus, unten sind die Nußhecken,
      das wirst du ja wissen,' sagte Rotkäppchen.
      (Page 176)
--------------------------------------------------------------------------------
    4 Da ging er ein Weilchen neben Rotkäppchen her, dann sprach er
      'Rotkäppchen, sieh einmal die schönen Blumen, die ringsumher
      stehen, warum guckst du dich nicht um? ich glaube, du hörst gar
      nicht, wie die Vöglein so lieblich singen? du gehst ja für dich
      hin, als wenn du zur Schule gingst, und ist so lustig haußen in
      dem Wald'.
      (Page 176)
--------------------------------------------------------------------------------
    5  Rotkäppchen schlug die Augen auf, und als es sah, wie die
      Sonnenstrahlen durch die Bäume hin- und hertanzten und alles
      voll schöner Blumen stand, dachte es 'wenn ich der Großmutter
      einen frischen Strauß mitbringe, der wird ihr auch Freude
      machen; es ist so früh am Tag, daß ich doch zu rechter Zeit
      ankomme,' lief vom Wege ab in den Wald hinein und suchte Blumen.
      (Page 177)
--------------------------------------------------------------------------------
    6 Und wenn es eine gebrochen hatte, meinte es, weiter hinaus
      stände eine schönere, und lief darnach, und geriet immer tiefer
      in den Wald hinein.
      (Page 177)
--------------------------------------------------------------------------------
    7  Da waren alle drei vergnügt; der Jäger zog dem Wolf den Pelz ab
      und ging damit heim, die Großmutter aß den Kuchen und trank den
      Wein, den Rotkäppchen gebracht hatte, und erholte sich wieder,
      Rotkäppchen aber dachte 'du willst dein Lebtag nicht wieder
      allein vom Wege ab in den Wald laufen, wenn dirs die Mutter
      verboten hat.'
      (Page 179)
--------------------------------------------------------------------------------
  TOTAL= 7 WORDS IN 7 SENTENCES
```

2.4で述べたようにグリム•データベースにはレコード番号が付加されていない。用例の出現位置の表示は，ページ単位で行う。このためには，あらかじめ原典のページ終了位置に当たるデータ•ファイル上の箇所に，ページ

処理例11　用 例 検 索（パラグラフ単位）

```
RESULTS OF KEYWORD SEARCH
FILENAME= M026.TXT

KEYWORD= Jäger,Pelz
================================================================================
--------------------------------------------------------------------------------
    1  Wie der Wolf sein Gelüsten gestillt hatte, legte er sich wieder
       ins Bett, schlief ein und fing an überlaut zu schnarchen. Der
       Jäger ging eben an dem Haus vorbei und dachte 'wie die alte Frau
       schnarcht, du mußt doch sehen, ob ihr etwas fehlt.' Da trat er
       in die Stube, und wie er vor das Bette kam, so sah er, daß der
       Wolf darin lag. 'Finde ich dich hier, du alter Sünder,' sagte
       er, 'ich habe dich lange gesucht.' Nun wollte er seine Büchse
       anlegen, da fiel ihm ein, der Wolf könnte die Großmutter
       gefressen haben, und sie wäre noch zu retten: schoß nicht,
       sondern nahm eine Schere und fing an, dem schlafenden Wolf den
       Bauch aufzuschneiden. Wie er ein paar Schnitte getan hatte, da
       sah er das rote Käppchen leuchten, und noch ein paar Schnitte,
       da sprang das Mädchen heraus und rief 'ach wie war ich
       erschrocken, wie wars so dunkel in dem Wolf seinem Leib!' Und
       dann kam die alte Großmutter auch noch lebendig heraus und
       konnte kaum atmen. Rotkäppchen aber holte geschwind große
       Steine, damit füllten sie dem Wolf den Leib, und wie er
       aufwachte, wollte er fortspringen, aber die Steine waren so
       schwer, daß er gleich niedersank und sich totfiel.
       (Page 178)
--------------------------------------------------------------------------------
    2  Da waren alle drei vergnügt; der Jäger zog dem Wolf den Pelz ab
       und ging damit heim, die Großmutter aß den Kuchen und trank den
       Wein, den Rotkäppchen gebracht hatte, und erholte sich wieder,
       Rotkäppchen aber dachte 'du willst dein Lebtag nicht wieder
       allein vom Wege ab in den Wald laufen, wenn dirs die Mutter
       verboten hat.'
       (Page 179)
--------------------------------------------------------------------------------
  TOTAL= 3 WORDS IN 2 PARAGRAPHS
```

区切り識別記号（初期値は$）を書き込んでおくことが必要で，この数をプログラムがカウントして該当ページを表示させるようになっている。ページは1ページからカウントされるが，次の書式で開始ページを指定することもできる。

（例）　%$56%

この例のように，開始ページを指定する場合は，$に続けてページを数字で書き込み，その全体を%で囲めばよい。この%記号はスキップ範囲を指定する制御文字で（4章参照），本来はデータ・ファイル中で，処理の対象か

ら除外したい部分（例えば表題など）を指定するのに用いられる。**処理例 1** に見られるように，データ・ファイルの最初にこの指定をするのが一般的な使い方である。

このようにデータ・ファイル中に，とくにレコード番号のようなものがなくてもよいわけであるが，TEDDY ではレコード番号付きのデータにも対応している。この場合は通常レコードの末尾が改行コード（$0D_H + 0A_H$）で終わっているので，これをレコードの区切り記号として処理する。つまりパラグラフ単位の検索モードで処理するわけである。この場合，**処理例 10, 11** にあるような左端の用例番号は表示されず，代わりにユーザーの書き込んだレコード番号が表示される。

TEDDY の検索機能は，以上述べたように，該当する用例を単に拾い出すだけではなく指定キーワードを逐一マークしたり，さまざまなデータにフレキシブルに対応（上では述べなかったが，英文ワープロ WordStar の文書ファイルをそのまま処理できるような機能も備えている）できたり，出力形式についても簡単な操作で細かい指定ができるよう工夫を凝らしており，使いやすいソフトと自負しているが，まだまだ不完全で，数多くの問題が残されているのも事実である。

まず検索速度の遅さという問題があり，これは早急に何らかの改善をする必要があると考えている。速度に影響を与える要因としては，ハードウェア，ソフトウェア（処理系），アルゴリズムの問題といろいろあげられるが，とくに問題なのは，処理系とアルゴリズムであろう。処理系に関しては，現バージョンの開発言語である N88BASIC コンパイラが，コンパイラといってもいわゆる P コード方式であり，ネイティブ・コード方式に比べると，どうしても分が悪い（もっとも画面表示など一部の処理に関しては Quick BASIC や BASIC98 など他のネイティブ・コード・コンパイラに比べても抜群に高速であるというテスト結果もある[6]）。いずれにしても，この章の始めの方でふれたように，ドイツ語特殊文字の画面表示の問題が解決できた

段階で，他のより高速な処理系に移植することを考えている。

アルゴリズムの問題に関しては，いわば素人が試行錯誤的に作ったもので，こと速度効率についてはあまり自信がない。もっともこれまでの作成意図としては，速度効率よりも，上で述べたような細かい処理，ユーザー•インターフェイスのやさしさを最優先させてきたという事情もあって，検索以前の処理で時間をロスしている面もある。例えば，プログラムの流れに立ち入って言うと，データ•ファイルを読み込む間に，その都度あらかじめ指定した出力時の一行の文字数に合わせてデータを加工する。この際にワード•ラップ処理も同時に施される。この加工した一行ごとのデータを順番に検索にかける，といった処理をしているので，この加工のプロセスが速度効率を下げているという面がある。いかにも回りくどい処理であるが，一つの文字変数に許される最大文字数が 255 文字という N88BASIC の仕様上の制約から，どうしてもこうせざるを得ない事情もある。この他，出力時にもドイツ語画面表示ルーチンのようなスピードを落とすプロセスが含まれている。高速化のアルゴリズムは今後の検討課題であるが，応急の対応策としては，GREP のような高速のパターン検索ツールで，まず該当レコードを拾い出しておき，これを TEDDY にかけて細かい処理をさせるということも考えられよう。

もう一つだけ問題をあげると，キーワード指定方法の強化という点がある。TEDDY でも，すでに見たように複数キーワードの指定やハイフンによるキーワード前後の文脈の指定，あるいは大小文字の区別の有無など必要最小限の機能は備わっているが，複数キーワード指定の場合では，OR 検索のみで AND 検索ができないなどの問題点がある。先にふれた GREP のように，必要に応じてさまざまな正規表現を使えるようにするのが今後の課題であると考えている。

4　制御文字のまとめ

TEDDY で用いられる各種制御文字について，以上でふれたものを含め，一括してここでまとめておくことにする。各項目で［　］内の記号は初期設定値であるが，TEDDY はグリム•データベースのみならず，他のデータベースでも汎用的に利用できるように配慮しているので，この初期設定値は変更できるようになっている。

1）　語彙統計で必要な制御記号

◇大文字の小文字読みかえ記号……［＊］

本来小文字ではじまる単語で，文頭等で大文字で表記されている単語は，その直前にこの制御文字を置くことによってプログラムが小文字に変換して処理する。

（例）＊Das ist ein Buch.
＊Er sagt: „＊Ich liebe sie.“
Kleider machen Leute.　（変換しない場合）

◇単語境界記号……［＃］

2 語以上の単語を 1 語として扱いたい場合に指定する。

（例）　#Mona Lisa#

◇分離動詞・ハイフン処理記号……［＋］

テキスト中で分離しているドイツ語の分離動詞を一語として処理したり，また der Ein- und Ausgang のようにハイフンで結ばれている語句から，Eingang という文字列を取り出す（使用例は 3.2.1 参照）。

2) 用例検索で必要な制御記号

◇ページ区切り……［$］

原典のページ終了位置に当たるデータ•ファイル上の箇所に，この文字を書き込んでおくことにより，プログラムがページ数をカウントして，検索データの該当ページが表示される（使用例は3.3参照）。

◇文 境 界……［/］

複数の文を一つの文として扱いたい場合に，指定範囲の前後にこの文字を書き込む。センテンス単位の検索をする場合，この文字で囲まれた範囲が一つのレコードとして認識される。

3) その他の制御記号

◇スキップ範囲……［%］

この文字で囲まれた部分は，プログラムがスキップし，処理の対象から除外する。囲む範囲に制限はなく，複数行にわたって指定することができる。またテキスト中の任意の位置に置くことができる。

5. 今後の展開

以上グリム•データベースとその利用法について，その処理プログラムTEDDY の機能にふれながら紹介してきた。TEDDY は本文中にも述べたように，まだまだ未完成なソフトである。とくに用例検索機能については今後大幅な改訂を予定している。

このようなテキスト処理プログラムで従来よくある機能に，例えばOCPにおけるようなインデックスやコンコーダンスの作成があるが，TEDDYでは今後ともこの機能に対応する予定はない。こうした処理は結局のところ書誌を作成するのが目的であると言えようが，我々のささやかなデータベー

スでも，例えば全語彙を網羅するようなコンコーダンスを作成するとなると，数千ページのヴォリュームになるであろう。このような印刷物の形での書誌は，出版の可能性も考えにくいし，利用形態としても不便で時代に合わないように思われる。計算機がセンターで一部の専門家だけにしか操作できなかった時代であれば，専門家が計算機処理したものを書誌の体裁で普及させるという方法がいわば必須であったのかも知れない。周知のように現在は，個人の研究室で，あるいは自宅で，オンラインまたは磁気媒体に記憶されたデータベースから，検索プログラムによって随時必要な情報を引き出すことができる時代である。今後一層こうした利用方法が普及することは間違いないと思われる。

〔注〕

1) 詳細および問題点は山田（1990）で述べているので，そちらを参照していただきたい。
2) 長瀬・西村（1986）参照。
3) データベースの各フィールドに当たる文字列をコンマで区切り，通常各文字列を引用符［”］で囲んだ形式のテキスト・ファイル。
4) Butler（1985），S. 14.
5) Butler（1985），S. 19.
6) 日経バイト 1989年5月号の記事「PC-9801用 BASIC 3製品の実行速度比較」（S. 193-198）参照。

〔参考文献〕

Butler, C.（1985）: Computers in Linguistics. Oxford.
長瀬眞理・西村弘之（1986）: コンピュータによる文章解析入門——OCPへの招待——（オーム社）
樋口忠治・篠原　武（1987）: テキストデータベース「トーマス・マン・ファイル」の完成と再編成について（九州大学大型計算機センター広報，Vol. 20, No. 6, S. 582-596）
山田善久（1990）: パソコンにおけるドイツ語特殊文字の取扱いと問題点（ドイツ語情報処理研究，第1号，S. 28-41）