

「コンピュータにない漢字」の利用について
——「大規模文字セット」とマクロによる、データの再利用と共有——

福 田 忍

(北海道教育大学岩見沢校非常勤講師)

About use of the Chinese character which is not in a computer
—Reuse and sharing of data by the large-scale character set and the Macro program

HUKUDA Sinobu

はじめに

研究のため、あるいは教材作成のため、古文や漢文をコンピュータに入力する際、常に問題になるのは、利用したい漢字がコンピュータに存在しないことである。一例を挙げれば、筆者が『論語義疏』¹を全て入力したデータベースを作成した際には、全体約 125000 文字の内、800 回あまりにわたって、JIS 第一・第二水準²外の文字が出現した。

ことは古典に限らない。しばしば例として挙げられるところではあるが、森鷗外はパーソナルコンピュータ上では通常、森鷗外となってしまう。国語審議会が 2000 年末に答申した「表外漢字字体表」においては、「鷗」を「簡易慣用字体」として容認しているが、一方で、学校教育においては「鷗」を用いることを追認している³。

¹ 六朝・梁の皇侃（488～545）が、諸家の注釈を集めて著した『論語』の注釈書。

² 正式には JIS X 0208 と呼ばれる日本の文字コードの規格。6350 字余りの漢字を収録している。この範囲の文字は、ほとんどのコンピュータで使用可能である。

³ 「表外漢字字体表」前文には次のようにある。「学校教育用の教科書に使用される表外漢字は、常用漢字表の制定以前から、人名用漢字は別として基本的にいわゆる

両者を併存したことには批判もあるかもしれないが、学校教育において安易に略字を使用すべきでないことについては大方の賛同が得られるところであろう。

後述するように、漢字が不足していることに関する多くの不満は現在解消しつつあり、実は「鷗」も多くのパーソナルコンピュータで利用可能となっている。しかし、コンピュータに存在しない文字はまだ多く、多くの教育者・研究者の悩みの種となっているのが現状であろう。そこで本稿では、近年利用されるようになった、「大規模文字セット」「多文字環境」等と呼ばれる、多数の漢字を利用できる環境を作り出すソフトウェア（以下、「大規模文字セット」と呼ぶ）について、その利点と問題点を分析し、こうしたソフトウェアを、なるべく問題点が少ない形で利用するための方法について紹介する。

1 パーソナルコンピュータで利用できる文字

使用できない文字について考える前に、現在のパーソナルコンピュータで利用できる文字について概観しておこう。なお、現在パーソナルコンピュータの OS（基本ソフト）としては、マイクロソフト社の Windows およびアップル社の Mac OS が大きなシェアを持っているため、以下は主にこの二つについて述べる。

Windows と Mac OS はいずれも長い歴史を持ったソフトウェアであるが、それぞれの一般用途向けの最新版である Windows XP と Mac OS X は、両者とも内部の文字処理に、ユニコードと呼ばれるコード体系を使用している¹。

ユニコードは、業界団体であるユニコードコンソーシアムが制定したコード体系であり、現在（2002 年末）の最新バージョンである 3.2 では、7 万字あまりの漢字を収録している²。従って、理論的にはこれらの OS においては漢字 7 万字が利用できることになるが、ことはそう簡単ではない。その理由のひとつは、7 万字すべてを含むフォント（画面の表示や印刷に使用される書体）が必ずしも提供されていないこ

康熙字典体が用いられている。従って、表外漢字字体表によって現行教科書の漢字字体が変更されることはない。（4 その他関連事項（1）学校教育との関係）」

¹ Windows は 95 以降、Mac OS は 8.5 以降がユニコードを使用している。

² ユニコードコンソーシアムのホームページ <http://www.unicode.org/>において、その詳細を見ることができる。

とである。現時点では Windows・Mac OS いずれも、標準状態ではユニコードのバージョン 2 において定義された、2 万字程度の漢字が実装されているに過ぎない。

2 万字「程度」とは言っても、もちろん膨大な数であり、多くの文書においては、この範囲でこと足りるだろう。事実、先程例に挙げた「鷗」はこの範囲に含まれている。しかし、漢籍においては 2 万字の範囲を超える文字は決して珍しくない。例えばこれも先程例に挙げた『論語義疏』においては、この範囲を逸脱する文字が 14 種類出現する。

Windows XP においては、マイクロソフト社より Office XP Proofing Tools と称するソフトウェアが提供されており、インストールすることで、ユニコードで定義されたすべての文字を収録したフォントを使用できる。これによって、通常の漢籍の利用においては、文字の不足に関する不満はかなり解消される。『論語義疏』においても、この範囲を逸脱する文字は出現しない。しかし、ユニコードの文字は日本・中国・台湾・韓国の文字を、字形の差異を包括する形で統合する建前であったため、日本の人名・地名の表記や、写本などのデータにおいて問題となる異体字については、7 万字の範囲でも、必ずしも十分ではない。また、市販のソフトウェアであるため、当然のことながら導入にはコストがかかる。専門の研究者はいざ知らず、教員や学生にとっては、数文字のために投資を行うのはためらわれるだろう。

さて、こうして現在の多くのパーソナルコンピュータでは、2 万字（場合によっては 7 万字）あまりの漢字が利用できるのだが、ここにさらに大きな問題がある。それは、ユニコードが必ずしも OS やアプリケーションを超えて利用できないことである。例えば Windows XP 上のワードプロセッサである Microsoft Word で「森鷗外」と入力した文書を作成し、それを Mac OS X 上の同じ Microsoft Word で開くと、「鷗」の部分は表示されない¹。

さらに例を挙げれば、Windows XP の標準テキストエディタである「メモ帳」と、Mac OS X の標準テキストエディタである TextEdit は、いずれもユニコードに対応しているため、両者の間で「鷗」を含んだテキスト文書をやりとりできる。しかし、この文書を Windows の旧バージョンである Windows95 上の「メモ帳」で正しく表示

¹ 「鷗」に当たる部分に繁体字中国語のフォントを指定することで表示／印刷可能である。

することはできない。結局のところ現時点で、本当に安全に OS やアプリケーションを越えて利用できる日本の漢字は、相変わらず従来の JIS 第一・第二水準のみだということになる¹。

要するに、現在の OS は、多くの漢字を利用する潜在能力は有しているものの、その実装はまだ不完全であり、内部的には同じコード体系を使用してはいても、OS やアプリケーションを越えてデータを再利用あるいは共有することには多くの問題が存在している。やがてはユニコード、もしくは他のより優れたコード体系が、完全な形で各 OS に実装され、全てのアプリケーションがそれに対応して、利用者がなんら考えることなく多くの漢字を利用し、かつ共有できるようになるのが理想であろうが、現実の教育や研究に携わる者は、とにかく今実際に存在するものを利用して最大限の効果を上げなければならない。コンピュータに無い漢字（以後、「外字」と呼ぶ）を利用する技術が必要なゆえんである。次節ではいくつかの外字の利用法とその特徴・問題点について考察してみる。

2 外字の利用方法

2. 1 ユーザー定義外字による外字への対処とその問題点

パーソナルコンピュータが一般に利用されるようになった当初から、外字に対処するため、OS の一部やワードプロセッサなどのアプリケーションは、ユーザーが外字を作成する機能を提供してきた。コードの、文字が割り当てられていない領域にユーザー独自の文字を定義する機能である。これによって一応は希望する文字をコンピュータ上で利用できるようになるわけであるが、この方法には深刻な問題点が存在していた。

まず、当然のことながら外字を作成する作業が必要となる。特に、美しく印刷できるような文字を作成するには非常な労力を必要とすることが多い。第二に、外字は通常必要に応じて、いわばゆきあたりばつたりに作成されるため、大量に外字を作成している場合、同一の文字が複数作成される等の混乱を来しがちである。この

¹ 実はこれでも厳密には安全ではない。JIS X 0208 には 1978 年版と 1983 年版があり、文字の追加、字形の変更、コードの入れ換えが行われているからである。ただし現在はほとんど 1983 年版が使用されている。

場合同じ文字に複数のコードが割り当てられることになるため、当該文字を検索した際、一度に全てが検索されないという不都合が発生する。また外字は統一された順序を持たない。従って、データの並べ替えを行う際に問題が発生する¹。

そして第三に、これが最も大きな問題であるが、外字を利用しているデータは、アプリケーション、あるいはマシン（もちろん OS も）を越えて利用することができない。すなわち、あるワードプロセッサで外字を含んだ文書を作成しても、それを他のアプリケーション、例えばデータベースに移入するには、通常、文字情報を含まないテキストデータに変換しなければならない。その際、外字の情報は消えてしまうのが普通である。

OS のレベルで外字を設定できる場合は、この問題は回避できることもあるが、別のマシンにそのデータを移行する場合には、相手のマシンに同じ外字が定義されている場合を除いて、外字の情報は失われてしまう。さらに、マシン間で外字の定義が異なっていれば、作成したマシン以外でデータを利用する際、データの作成者が意図した文字とは全く異なる文字が、表示あるいは印刷されることになるだろう。

つまりユーザー定義の外字という方法を使う限り、折角のデータを他のアプリケーションで再利用したり、不特定多数の利用者間で共有したりすることが極めて困難なのである。インターネットの普及等により、データの公開・共有が活発化し、その重要度が益々高まっている中、これはまことに由々しき事態と言わなければならない。

2. 2 大規模文字セットの登場

ユーザー定義の外字に代わって数年前から脚光を浴びるようになったのが、パーソナルコンピュータ上で多数の漢字を使用可能にする、「大規模文字セット」「多文字環境」等と称せられるソフトウェア（フォントのセット）である。そのしくみと利点・問題点については後述するとして、まず現在利用できるソフトウェアを紹介すると、代表的なものに、「今昔文字鏡」「e 漢字」「GT 書体」などが存在する。

利用できるパーソナルコンピュータのシステムとして、「今昔文字鏡」は Macintosh

¹ ただし、JIS コード自体、文字配列の規則は統一されていない（第一水準は字音順、第二水準以上は部首画数順）。

と Windows に対応し、「GT 書体」は Macintosh と Windows に加えて、超漢字¹にも対応している。「e 漢字」は特定のシステムに依存せず、後述するように、現在はもっぱらインターネット上で外字を表示するのに用いられている。パーソナルコンピュータ上で文書の作成に使用するならば、「今昔文字鏡」もしくは「GT 書体」を用いることになる。

試みにこれらを利用して外字を使用してみよう。なお、こうした文字の利用法はソフトウェアによって異なるが、基本的には何らかの形（プログラム、電子的なコード表など）で希望の文字を検索し、ワードプロセッサなどに当該文字をコピーする方法を用いる。

GT 書体による

確かに外字（この場合はユニコードによる二万字の定義を超えた文字）が表示／印字できている。

「今昔文字鏡」は約 12 万字、「GT 書体」は約 6 万 7 千字を収録し、いずれも多く
の異体字を包摂している。従って、これらを利用すれば、ほとんどユーザー定義の外字を作成する必要はなくなるだろう。また、データを交換する双方が同じ「今昔文字鏡」なり「GT 書体」なりをインストールしているという条件付きではあるが、JIS 外の文字を含んだファイルをやりとりすることもできる。そうした意味で、これらのソフトウェアは非常に有益なものと言える。また、これらはいずれも、インターネット等を通じて公開され、個人が原則として無償で利用することを認めている。この点も、大きな利点であると言えよう。

以下、各製品について簡単に紹介する。

2. 3 大規模文字セットの各製品

「今昔文字鏡」は（株）エーアイ・ネットが開発・制作した、パーソナルコンピュータ（Windows および Macintosh）上で使用される文字フォントおよび、目的の文字を検索するためのソフトウェアである。現在、文字鏡研究会により、（株）エー

¹ 国産の OS である TRON に準拠したシステム。TRON は多種類の文字を利用することが可能なように設計されている。

イ・ネットの許諾のもとにインターネット (<http://www.mojikyo.org/>) を通じて公開されており、また、紀伊国屋書店から『今昔文字鏡単漢字 10 万字版』が、また同じく紀伊国屋書店から『パソコン悠悠漢字術 2002—今昔文字鏡徹底活用—』が出版されている。

「今昔文字鏡」には大修館『大漢和辞典』所収の漢字（約 5 万字）以外に、ユニコードの文字、さらには甲骨文字、西夏文字なども収録されている。また漢字ではないが、変体仮名のフォントも提供されており、国文学関連のデータ作成に便利である。

続いて、「e 漢字」は、京都大学人文科学研究所の勝村哲也氏らによって作成された文字セットで、ユニコードのフォントをはじめとして、『大漢和辞典』の親字、『康熙字典』の親字などを収録したフォントを公開している (<http://nohara.u-shimane.ac.jp/ekanji/index.html>)。「e 漢字」の特徴は、コンピュータの機種に依存しないフォントデータを提供していることであり、様々な機種が接続するインターネットのホームページ上で外字を表示する際などに利用されている。

さて、最後に「GT 書体」であるが、これは、東京大学の田村毅氏をプロジェクトリーダーとする、日本学術振興会未来開拓学術研究推進事業「マルチメディア通信システムにおける多国語処理研究プロジェクト」による成果である（書体名の G は学術振興会、T は東京大学の頭文字）。

当該プロジェクトは、「世界のありとあらゆる文字をコンピュータで処理できるようにしよう」（田村氏による¹）というものであり、その一環として日本語の漢字の電子化を行い、2000 年末に 66773 文字からなるフォントのセット（Windows、Mac OS、TRON 用）および、全ての文字を網羅したコードブックを、ホームページ (<http://www.l.u-tokyo.ac.jp/GT/index.html>) を通じて公開している。

「GT 書体」の特徴は、『大漢和辞典』をはじめとする漢字字典の文字を集積しただけでなく、通常漢字字典に収録されない、日本における異体字や国字の類を多く採集していることであり、漢籍関係のみならず、国文学における利用、あるいは JIS で表せない人名や地名への対応に気を配った文字セットとなっている。また、賛否

¹ 「多言語の中の日本語 日本語の中の多言語 漢字六万四千字のフォントセット公開に向けて」 (http://www.l.u-tokyo.ac.jp/KanjiWEB/01_01.html)

両論のある所かもしれないが、文字の重要な情報として画数を考慮しており、画数がはっきり分かるような文字のデザインになっている。

現在の所、「GT 書体」は後発であることもあって、既に数年前からインターネットや書籍を通じて広く利用されてきた「今昔文字鏡」の後塵を拝している感は否めない。収録文字数も、現時点では「今昔文字鏡」が「GT 書体」をはるかに上回っている。

しかし「GT 書体」には「今昔文字鏡」に対する極めて大きな優位点がある。それは、「今昔文字鏡」の利用に際しては、著作権者である（株）エーアイ・ネットの権利を侵害しないための様々な制限が存在するのに対して、（財）日本学術振興会が著作権を有する「GT 書体」には、利用における制限が非常に少ないことである。

一例を挙げれば、「今昔文字鏡」は、それを使用するためのプログラムやマクロ（特定のアプリケーション上で動作するプログラム）を作成することに制限を設けている¹。一方、「GT 書体」の使用許諾条件は、2002 年 12 月現在では以下の五条を定めているのみである²（文言は筆者が変更している）。

1. 著作権者は日本学術振興会であること。
2. フォントと漢字データは非営利的な目的にのみ使用すること。
3. 改変を行ったものを配布しないこと。
4. 対象の機種以外に移植する場合は届け出ること。
5. 使用で生じた障害には、著作権者は責任を負わないこと。

従って、「GT 書体」についてはプログラムやマクロを作成することも、それを配布することも自由である。知的財産権を守るのは当然のことだが、教育・学術に関わるものにはこうした緩やかさが重要だと考えるのは筆者だけであろうか。

いずれにしても筆者は、後述するように現在の大規模文字セットの利用には様々な問題点が存在し、それを補うにはマクロやプログラムが不可欠だと考えているため、現状では「GT 書体」を選択せざるを得ない。

¹ 「「今昔文字鏡フォント」および「今昔文字鏡フォントを利用するためのソフトウェア」のための使用許諾契約書」第三条第五項による。

² <http://www.l.u-tokyo.ac.jp/GT/license.html>

3 大規模文字セットによる多文字環境の仕組みと問題点

さて、話はやや前後するが、先程も見たように、現在流通している大規模文字セットを使用すると、確かに外字を利用することができる。なぜ大量の文字が扱えるのだろうか。

その秘密はソフトウェアのフォント指定の機能にある。現在、文字を扱う多くのアプリケーション（ワードプロセッサ、表計算など）が、一文字単位で文字の書体を指定できるようになっている。この機能は通常は、このように強調する際にゴシック体にするような利用法のためにあるわけだが、「今昔文字鏡」や「GT 書体」はこの機能を、書体ではなく文字そのものを切り替えるのに利用しているのである。先程の例、

GT 書体による 𪛗

の場合、「𪛗」には GT2000-09 というフォントが割り当てられている。試みにフォントを標準の明朝体に変えてみよう。

GT 書体による 渴

「渴」は JIS 第一水準の文字である。先程の無闇に画数の多い文字の正体は、一般的な JIS で定義された文字であったのだ。

さらに、同じ「GT 書体」であっても、異なる番号のフォントに変更してみよう。

GT 書体による 𪛘

GT2000-08 というフォントを指定してみた。これも正体は先程の「渴」に過ぎない¹。

ここで、通常の OS 上で大規模文字セットを利用することの問題点が理解できるだ

¹ このような切り替えを可能とするため、「今昔文字鏡」では Mojikyo M101 から Mojikyo M203 までの 33 個、「GT 書体」では GT2000-01 から GT2000-11 までの 11 個のフォントが提供されている。試みに明朝体の「𪛘」に「GT 書体」の 11 個のフォントを順に割り当ててみる。

ろう。

まず、これらはアプリケーションのフォント設定の機能を利用して見てくれの文字を切り替えている。従って、複数のフォントを使い分けることのできるアプリケーションでなければ、多数の漢字を使用することはできない。ワードプロセッサは通常こうした機能を有しているが、データベース、表計算、テキストエディタなどにはこのような機能を持たないアプリケーションも多い。さらに、あるアプリケーションで大規模文字セットが利用できたとしても、他のアプリケーションにデータを移行する際には、既に述べたように、通常、文字の属性を含まないテキストにしななければならない、この際フォントの情報は消えてしまう。従って、「鱺」を送ったつもりでも、相手のアプリケーションでは「渴」としか認識されない。他のマシンにデータを移行する際、そのマシンに同じ大規模文字セットがインストールされていなければ、意図した文字を送ることはできないのはもちろんである。

また、見出しなど、強調したい部分のフォントを地の文とは変えることはしばしば行われることだが、ここに大規模文字セットの文字が存在していたら、それは異なる文字となってしまう。

さらに厄介なのは、検索や置換を行う場合である。通常、検索・置換は、書体の情報を含まない文字そのもの（コンピュータの内部的なコード）に対して実行される。従って、「GT 書体」の「鱺」を検索した場合、「渴」も発見されてしまう。存在していれば、もちろん「𪗇」も検索対象となる。要は「渴」に対して大規模文字セットのフォントが指定されて他の文字に見えているものは全て検索の対象となってしまう。同様に置換を実行すれば、これらは全て他の文字に変わってしまうだろう¹。

以上を要するに、「今昔文字鏡」や「GT 書体」などの大規模文字セットの導入により、利用者は外字を作成する苦勞からは確かに解放され、同一の大規模文字セットをインストールしているという条件付きながら、外字を含むデータを他人とやりとりすることも可能となった。しかし、アプリケーションを越えてデータを再利用

¹ 検索・置換における問題を回避する方法がないわけではない。アプリケーションによっては、検索や置換の対象にフォントの属性を含めることができる。従って、「鱺」のみを検索・置換したい場合は、同時にフォント属性として GT2000-09 を指定すれば、「𪗇」や「渴」は実行の対象とならない。

あるいは共有することは相変わらず困難であり、さらに利用者は、ひとつのコードに複数の文字が割り当てられることに起因する問題を新たに抱え込むことになったのである。

4 大規模文字セットの問題点の回避

さて、上記の問題を回避するにはどうすれば良いだろうか。もちろん TRON のように最初から大規模文字セットを取り込んだ OS を使用するという解決方法もありうる。しかし、Windows や Mac OS などの既存のシステムには多くのソフトウェアの蓄積が存在し、簡単に移行することはできないのが普通だろう。また、TRON は多くの文字を使用するために独自のコードを使用しており、他のマシンとデータの共有が困難であるという点についてはなんら解決とならない。

Windows および Mac OS において、上述の問題を回避しつつ大規模文字セットを利用する試みとして、クリスティアン・ウィッテアン氏が作成した「IRIZ 漢字ベース」¹が存在する。「IRIZ 漢字ベース」は Windows および Mac OS 双方のワードプロセッサ上で動作するマクロであり、文書中にテキストで記述された CNS コード（台湾政府が制定した文字コードで、48000 字余りを収録）を、該当する文字の画像に変換する。

コードはアルファベットと数字によるテキストデータであるから、OS やアプリケーションに依存しない。従ってももちろん上述の大規模文字セットにまつわる問題は発生しない。データの共有や他のアプリケーションでの利用はこの状態で行い、印刷などの限られた場面でのみ、マクロによって変換を実行し、OS やアプリケーションに依存する実際の文字を使用する。OS やアプリケーションに依存する部分と依存しない部分を切り分けることによって問題を回避したわけである。コードの状態では人間にとっての閲覧性が良くないという欠点はあるものの、これは巧みな解決策であると言えよう。

そこで筆者は、「IRIZ 漢字ベース」のアイディアに基づいて、「GT 書体」のコード

¹ 「電子達磨」4号（花園大学禅学研究所，1995）所収。また、花園大学禅学研究所のホームページ（<http://www.ijjnet.or.jp/iriz/irizhtml/kanjibas/cefintroj.htm>）に解説が掲載されている。

と該当文字とを相互に変換する Microsoft Word 用のマクロを作成した。以下、その詳細について記述する。

5 マクロによる「GT 書体」の利用

筆者が作成したマクロは前述のように、Microsoft Word の文書中に通常のテキスト文字によって記入された「GT 書体」のコードを実際のフォントに変換する機能を提供する。また、逆に実際のフォントをコードに変換する機能を提供する。具体的には、「GT098a89;」という文字列を、「鱺」に変換し、また逆に「鱺」を「GT098a89;」に変換する。なお、Microsoft Word は Windows、Mac OS 双方に対応した版が提供されており、筆者のマクロは両方で動作することを確認している。

繰り返しとなるが、この方式では、コードは単なるテキストにすぎないため、OS やアプリケーションを越えてやりとりすることが可能である。例えば、Windows で作成したデータを Mac OS に輸入することも、その逆も全く問題は発生しない。また、作成したデータを表計算ソフトやデータベースなど、他のアプリケーションに移行することも容易である。

具体的な使用例として、筆者は自宅では Macintosh (Mac OS X) を使用し、図書館での調べ物などの際には Windows XP を搭載したノートパソコンを持参する。処理能力に限りのあるノートパソコンでは、軽快なエディタを使用してテキストを入力する。もちろんこの際、外字（筆者は JIS 第一・第二水準以外の漢字をすべて「外字」としている）は「GT 書体」のコードを記入する。帰宅後、データを Macintosh に転送して Microsoft Word に読み込み、必要であればマクロによってコードを実際のフォントに置換し、閲覧やプリントアウトを行う。その後、他の OS やアプリケーションでの使用を考えて、再びマクロによってコードの状態に戻しておく。

また、翻訳やデータ入力などの共同作業を行う際には、全員が共通に使用できる文字の範囲を確認し、それを逸脱する文字はコードを記入することに取り決めておく。各参加者は安心して使い慣れたマシンやアプリケーションを使用することができる。同様にインターネットなどでデータを公開する際も、使用する文字の範囲を明記し、それを超えるものについてはテキストでコードを記入している旨を断っておく。

以上を要するに、テキストによる「GT 書体」コードの記述と、それを処理するマ

クロの組み合わせにより、外字の使用にまつわる諸々の問題を回避し、アプリケーション間、あるいは異機種・異 OS 間で、データの再利用・共有を図ることが可能となる。以下、筆者作成のマクロの使用法について解説する。

5. 1 マクロのインストール

本稿に付録したソースリストを、Microsoft Word の Visual Basic Editor を使用して入力する。もしくは、筆者のホームページ (<http://homepage1.nifty.com/chinaclassic/macro/macrotile.html>) からインストーラーをダウンロードし、インストーラーの内容に従ってインストールする。なお、本マクロは、Windows においては Word 2000、Macintosh においては、Word 2001 および Word X で動作することを確認している。

5. 2 マクロの使用法

1. 文書中に、GT+フォント番号+シフト JIS コード+; の形式で、利用したい「GT 書体」の文字を記述する。右は GT フォントプロジェクトから提供されているコードブックの例である。二段目の GT09 と三段目の 8A89 を続けて記述する。(アルファベットは小文字でも可)。最後の「;」を忘れないこと。なお、この時使用するのは1バイト文字 (いわゆる半角文字) でなければならない。



066773
GT09
8A89



例) GT 書体による GT098A89;

2. 「ツール」メニューから「マクロ」を選択し、マクロ・CodetoGT を起動する。インストーラーを使用してインストールした場合は、「ツール」メニューをクリックし、「GT 書体」にカーソルをあわせ、現れた「コードから GT 書体へ」を選択する。文書中に挿入されたコードが、すべて該当する「GT 書体」の文字に置換される。

例) GT 書体による 龠

3. 「ツール」メニューから「マクロ」を選択し、マクロ・GTtoCode を起動する。インストーラーを使用してインストールした場合は、「ツール」メニューをクリックし、「GT 書体」にカーソルをあわせ、現れた「GT 書体からコードへ」を選択すると、文書中の「GT 書体」の文字が、すべて該当するコードに置換さ

れる。

例) GT 書体による GT098a89;

なお、Windows で利用する場合は、「GT 書体」のコードを検索する際に、乙部巖己氏作成の「GT コード検索」アプリケーションを利用することができる。部首と画数から、「GT 書体」の文字コードを検索することが可能である。さらに同アプリケーションで、「設定」メニューの「クリップボードへのコピー方法」を選択すると現れる、「文字情報の出力形式」メッセージボックスに、「GT^N^S;」と入力し、「OK」をクリックすると、以後、希望の文字を選択してコピーを行うと、「GT 書体」のコードが、本マクロで使用する形式でクリップボードにコピーされる。そのまま Microsoft Word 文書にペーストすれば良いので大変便利に使用できる。「GT コード検索」アプリケーションは、乙部氏のホームページから入手できる。(<http://ms326.ms.u-tokyo.ac.jp/otobe/hobby/GTcode.html>)

5. 3 マクロ使用上の注意

本マクロプログラムによってコードを実際のフォントに変換した後の文書の使用に対する注意点は、以下の通りである。これらは基本的に、既に述べた、「大規模文字セット」に共通する問題である。もちろん、テキストでコードが記述されている段階ではこれらの問題は発生しない。フォントへの変換は最終的な印刷・閲覧のみに止めておくことが望ましい。

1. 変換後のフォントの変更

前述のように、「GT 書体」は、ひとつの文字コードに対して複数のフォントを指定することにより、多数の文字の表示や印刷を可能にしている。そのため変換後にフォント名を変更してしまう（全文選択をした上で、フォントを変更する等）と、意図した文字が表示・印刷できなくなる。この場合、「GT 書体」からコードに変換するマクロを使用しても、もはやコードに戻すことはできない。十分注意のこと。

2. 検索と置換

これも前述のように、先程の例で言えば、「鱺」を検索・置換すると、同じコードを持つ「渴」も一緒に検索・置換されてしまう。「鱺」のみ、あるいは「渴」のみを検索・置換したい場合は、条件にフォント属性を加えれば良い。

3. 標準表示・アウトライン表示での使用

Microsoft Word の標準表示・アウトライン表示では、高速な表示を行うために、

文字フォントの指定が画面に反映されないようにすることができる（下書きフォント）。このオプションを指定している場合には、文字はすべて下書き用のフォントによって表示されるため、GT 書体の文字は正しく表示されない。なお、フォントの指定自体が無効となっているわけではないので、プリントアウトは正しく行われる。

4. テキストファイルへの書き出し

変換後の漢字を含む Microsoft Word 文書をテキストファイルに書き出すと、フォントの指定が無効になるため、意図する文字と異なってしまう。テキストファイルに書き出す前に、必ずマクロを使用してコードに戻す必要がある。

例) GT 書体による鱺 → GT 書体による渴

5. 変換後の書き足し

変換後の文字に続けて文字を入力すると、フォントの指定が「GT 書体」のままになっているため、表示が乱れることがある。表示が乱れた文字を選択し、標準で使用しているフォントに変更すれば、正常な表示に戻る。

例) GT 書体による鱺に揃けて蜻縮 → (「鱺に」以降のフォントを変更) → GT 書体による鱺に続けて入力

おわりに

以上、必要な漢字がコンピュータに無い場合の対処方法とその問題点について概括し、その問題を回避する方法として、テキストによるコードの記述とマクロによる処理を提案した。

もとよりこのような方法は、より良い文字環境が提供されるようになるまでの過渡的なものではあろう。しかし、個人の研究や教材作成のための比較的小規模なデータの作成・公開においては、最も現実的な解決方法であると考えられる。

なお、「アプリケーション間でのデータの再利用・共有」という命題を掲げながら、提供するマクロが Microsoft Word 用のみであるのは、自己矛盾のそしりを免れない。他のアプリケーションに対応したマクロの作成については、筆者の今後の課題としたい。

参考文献

大野悟他「10日でおぼえる Word VBA 入門教室」(株式会社翔泳社, 1999年8月)

樺島忠夫「「表外漢字字体表」とは何か」（大修館書店「月刊しにか」2001年2月号）

小林龍生他「インターネット時代の文字コード」（共立出版，2001年4月）

漢字文献情報処理研究会編「電腦中国学Ⅱ」（好文出版，2002年4月）

樺島忠夫「「表外漢字字体表」と JIS 漢字」（大修館書店「月刊しにか」2002年5月号）

伊藤英俊「ユニコード時代は来ている」（大修館書店「月刊しにか」2002年5月号）

付録 マクロのソースコード

```
Sub CodetoGT()
' 文書中の GT 書体コード (GT+フォント番号+フォント内シフト JIS コード+) を
' 実際の GT 書体に変換する。
'
    With Selection.Find
        .ClearFormatting
        .MatchAllWordForms = False
        .MatchSoundsLike = False
        .MatchFuzzy = False
        .MatchWildcards = True
        .Text = "GT[0-1][0-9]????;"
        .Forward = True
        .Wrap = wdFindContinue
    End With
    Do While Selection.Find.Execute
        With Selection
            FntNum = Mid(.Text, 3, 2)
            ChrNum = "&H" & Mid(.Text, 5, 4)
            If Val(FntNum) > 11 Or Val(FntNum) < 1 Then
                MsgBox "フォント番号が誤っています"
                Exit Sub
            End If
            If Val(ChrNum) > &HF040 Or Val(ChrNum) < &H889F Then
                MsgBox "フォント内のシフトコードが誤っています"
                Exit Sub
            End If
            FntNum = "GT2000-" & FntNum
            .Text = Chr(ChrNum)
            .Font.Name = FntNum
            .Collapse direction:=wdCollapseEnd
        End With
    Loop
    MsgBox "コード番号から GT 書体への変換を完了しました"
End Sub

Sub GTtoCode()
' 文書中の GT 書体を、コード (GT+フォント番号+フォント内シフト JIS コード+)
' に変換する。
'
    For i = 1 To 11
        With Selection.Find
            .ClearFormatting
            .MatchAllWordForms = False
            .MatchSoundsLike = False
            .MatchFuzzy = False
            .MatchWildcards = True
            .Format = True
            .Font.Name = "GT2000-" & Format(i, "00")
            .Text = "?"
            .Forward = True
            .Wrap = wdFindContinue
        End With
        Do While Selection.Find.Execute
            With Selection
                ChrCd = Hex(Asc(.Text))
                .Text = "GT" & Format(i, "00") & ChrCd & ";"
                .Font.Name = "MS 明朝"
                .Collapse direction:=wdCollapseEnd
            End With
        Loop
    Next
    MsgBox "GT 書体からコード番号への変換を完了しました"
End Sub
```