

Unicode の *character* 概念に関する一考察

師 茂樹*

2004年2月19日

1 はじめに

文字を書くという行為は、コンピュータを使った活動の中でも中心的、根本的な位置を占めている。最近では(私が今書いているような)通常の文章を書くことも重要な用途となったが、より古く、より本質的なのはプログラムを書くということである。コンピュータで文書を書くための諸システムの歴史をどの時点から始めるかについては議論はあるが、最初の高級言語と言われる John Mauchly の Short Order Code が 1949 年に考案されたというから、これをひとつの画期と考えることができるかもしれない*1。いずれにせよ、コンピュータ上の書記活動の歴史は、コンピュータ自体の歴史—控え目に言ってもパーソナル・コンピュータの歴史—に匹敵するほどの長い歴史を持っていると言っても言い過ぎではないのではなからうか。

さて、1967 年に ASCII コードが制定されて以来、標準化された符号化文字集合による文字処理・テキスト処理が行われてきた。ASCII の後、多くの国でそれぞれのローカルな規格が制定されたが、国内の需要を満たし切れないための不満、あるいはインターネットの普及などによる情報交換の国際化が進むと、複数の規格を様々な形で同時に矛盾なく利用する方法が考案されたり (ISO 2022、Mule など)、逆にそれらを統合する動きがあったり (Unicode、ISO/IEC 10646 など)、あるいはベンダが外字として拡張したり (IBM 拡張、NEC 拡張など)、独自に文字集合を構築するプロジェクト (e 漢

字、今昔文字鏡、GT 明朝など)が見られるようになった。現在、コンピュータ上の多くのシステムにおいて Unicode が採用されるようになり、一時の混乱、批判はなりを潜めたかのようなのであるが、現在 JIS X 0213 や GB 18030*2のようなローカルな規格が制定され続けていることからわかるように、根本的な解決には至っていない。

本稿では、現在もっとも普及している Unicode において定義されている *character**3という概念について考察することによって、終息する様子のない符号化文字集合をめぐる諸問題の原因の一端を追求したい。この *character* は、単純に日本語訳をすれば「文字」ということになるが、我々の日常的な文字の感覚とは多少異なった抽象的な定義をされており、それが Unicode の設計の根幹に据えられている。しかもこの考え方は、Unicode のみに限定されるものではなく、他の多くの符号化文字集合規格や、Unicode と正反対のアプローチをしているときられる TRON コードなどにも、(明言されることはないものの) ほぼ共通して見られるものであると考えられるのである。しかしながら、それが批判的に検討されることは、これまであまりなかったように思われる。

なお、本稿では、最新版の Unicode Version 4.0 ([2]) を考察の対象とする。

* 花園大学 (s-moro@hanazono.ac.jp)

*1 ちなみに Fortran の発表は 1957 年である。

*2 Unicode の上位集合として制定された GB18030 は、ローカル規格から見た Unicode の評価が垣間見えて興味深い。[8] 参照。

*3 以下、イタリック体で *character* と書く場合には Unicode の定義する「character」を指す。

2 *character* の定義

2.1 10 の原則

Unicode における *character* の定義およびそれに関連する情報は、主に第 2 章の “10 Unicode Design Principles” ([2], p. 14) にまとまって見られる。

1. Universality
2. Efficiency
3. Characters, Not Glyphs
4. Semantics
5. Plain Text
6. Logical Order
7. Unification
8. Dynamic Composition
9. Equivalent Sequences
10. Convertibility

多少なりとも Unicode による処理に取り組んだ者にとっては自明のことであろうが、この原則は必ずしも十全に守られているわけではなく、またこれまでの開発の歴史の中で何度となく変更されている。例えば 1 番目の「Universality」は、Version 3.2 以前では「Sixteen-bit character code」であった ([9] 参照)。また、拡張漢字 B の制定において Unification の原則に変節が見られたという川幡太一氏の指摘もある ([5])。

しかし、様々な規格において、設計原則の説明にこれほど紙数を割いている例はほとんどなく、またこの原則 (の理念) は他の規格に対して大きな影響を与えていることから、これを考察することは大きな意義があるのではないと思われる。

本稿において考察の対象とするのは、主にこの中の「Characters, Not Glyphs」「Semantics」「Logical Order」「Unification」である。

2.2 抽象的な *character*

character について端的に説かれているのは、第 3 番目の原則「Characters, Not Glyphs」である。ここでは、*character* について以下のように説明されている。

The Unicode Standard draws a distinction between *characters* and *glyphs*. Characters are the abstract representations of the smallest components of written language that have semantic value. (...) Characters represented by code points. (...) The Unicode Standard deals only with character codes. ([2], p. 15)

この説明からまずわかることは、Unicode で言われる *character* が、視覚的な表象の差異を捨象した抽象的なものとして規定されている点、そして書記言語の中の最小構成部品とされている点であろう。この *character* にひとつのコードポイントが対応することになる。

この中、特に前者は、後に見る他の原則の前提となるものであり、Unicode の設計思想を考える上で非常に重要である。これに関連して、他所では、抽象的な文字を表す用語として *Abstract Character* が定義されており ([2], p. 64)、*character* の意味の一部であるとされる ([2], p. 1365)。 *Abstract Character* の定義は以下の通りである。

Abstract Character: A unit of information used for the organization, control, or representation of textual data.

- When representing data, the nature of that data is generally symbolic as opposed to some other kind of data (for example, aural or visual). Examples of such symbolic data include letters, ideographs, digits, punctuation, technical symbols, and dingbats.
- An abstract character has no concrete form and should not be confused with a *glyph*.
- An abstract character does not necessarily correspond to what a user thinks of as a “character” and should not be confused with a *grapheme*.
- The abstract characters encoded by

the Unicode Standard are known as Unicode abstract characters.

- Abstract characters not directly encoded by the Unicode Standard can often be represented by the use of combining character sequences.

すなわち、Unicode における *character* は、グリフなどの視覚的、具象的、物理的な実体*4を持つ文字ではなく、「抽象的な文字」であるとされる。この「抽象的な文字」という考え方はわかりにくいですが、上の引用で「symbolic」という言葉が見えることから考えると、文字の記号的な機能を実体的に捉えたものであると思われる。確かに、我々は視覚的に多少の差がある文字があったとしてもそれを同じ文字として読むことができるので、その意味ではこのような抽象的な文字の考え方も我々の文字観と矛盾するものではない。

この *character* の記号性は、先に引いた原則中に「semantic value」を持つと説明されていることに通ずる。この「semantic value」については、4 番目の原則において説明がある。

Characters have well-defined semantics. Characters property tables are provided for use in parsing, sorting, and other algorithms requiring semantic knowledge about the code points. The properties identified by the Unicode Standard include numeric, spacing, combination, and directionality properties (...). Additional properties may be defined as needed from time to time. ([2], pp. 17–18)

各 *character* の「semantic value」もしくは「semantics」は、具体的には文字プロパティテーブルにおいて規定されている。現在のところ、数字の持つ数値、文字の占める幅、組み合わせて用いる文字

*4 場合によっては、ディスプレイに表示されたりプリンタで印字された物理的に存在するグリフを「グリフイメージ」とし、それ以前のデータとしてのグリフと区別することができる。

か否か、書記方向で変化する振る舞い方などが定められており、これは今後も増加されるとのことであるが、文字の「semantics」としてはいささか物足りない印象があることは免れえない。

ここで注目すべきは、「semantics」が「well-defined」とされている点ではないかと思う。これはすなわち、「semantics」として適切でないもの、曖昧なものを排除していることに他ならない。言い換えればこれは文字にとって何が本質的で何が本質的でないか、ということをあらかじめ規定しているということであろう。このような難しい問題に取り組んでいるのであれば、プロパティテーブルの貧弱さも納得がいく。

2.3 Unification と script

さて、このような *character* の定義に基づいて、有名な Unification*5の原則が成立している。

The Unicode Standard avoids duplicate encoding of characters by unifying them within scripts across languages; characters that are equivalent are given a single code. (...) Avoidance of duplicate encoding of characters is important to avoid visual ambiguity. ([2], pp. 19–20)

ここで言われているのは、複数の「language」を含む「script」単位での文字の統合である。これについては次のような説明がある。

Script. A collection of symbols used to represent textual information in one or more writing systems. ([2], p. 1377)

すなわち、複数の言語 (language) = 書記システム (writing system) で共通して用いられる記号のまとまった集合を「script」と呼んでいるようである。例えば、英語やフランス語はラテン文字という script を用いているので統合された x を用いるが、数式などで用いる x は script が異なるため統合さ

*5 Unification の訳語には「統合」「包摂」の両方があるが、本稿では「統合」を用いる。

れない、という具合である。また、日本語はひらがな、カタカナ、漢字、ラテン文字、その他記号などの複数の script が組み合わされた書記システムである。ただし、何をどの script に属するかについては明確な根拠を見出せず、恣意的な印象がある。

逆に言えば、Unicode においては script を超えた統合はしていない。したがって、Unicode の批判者としてとりあげられることの多い TRON コードとの違いは、乱暴に言えば統合のレベルが script レベルか language レベルかの差にすぎないとも言える。

ところで、ここで「visual ambiguity」と言われていることからわかるように、先の原則で *character* から分離されていたグリフなどの視覚的な表象は曖昧なものとして捉えられている*6。これは、先に「semantics」のところ指摘した「well-defined」の考え方に通ずる。すなわち、*character* とは、あらゆる曖昧なものを排除した文字の本質というべきものとして規定されているのである。

2.4 理論上のテキスト≒音声言語

では、Unicode の言う抽象的な文字によって構成されるテキストは、どのようなものであろうか。7 番目の原則では、これについて説明されている。

Unicode text is stored in *logical order* in the memory representation, roughly corresponding to the order in which text is typed in via the keyboard. In some circumstances, the order of characters differs from this logical order when the text is displayed or printed. (...) For the most part, logical order corresponds to *phonetic order*. ([2], pp. 18-29)

ここでは、Unicode の文字によって構成されるテキストが、メモリ上において「論理的な順序 (logical

order)」によって並べられなければならない、とされている。そしてこの「論理的な順序」とは、キーボードから入力する順序、もしくは音声言語において発話される際の順序 (phonetic order) に近いものであると言う。この「論理的な順序」と(ある状況において)対立するのは、画面表示や印刷などにおける視覚的な文字の順序である。アルファベット/英語や漢字/中国語の場合、両者が対立することはないが、デーヴァナーガリ/ヒンドゥー語などの場合には、子音と母音の見た目の位置が通常の文字の流れと逆になるなどの対立が発生することがある。これは、先の原則中にあつた *character* とグリフとの分離と重なる内容である。

音声言語的な順序を「logical」と見なす背景には、一般に広く受け入れられている文字言語に対する音声言語の先行性があるのであろう*7。

3 *character* 概念の背景

3.1 離散的で確定的な「文字」

では、以上のような *character* 概念の背景には、文字に対するどのような考え方があるのだろうか。

文字処理をある種の知識処理と考えれば、ドレイファスが AI の分析のなかで述べる「存在論的前提」は、*character* 概念の抱える問題を鋭く指摘するものであろうと思われる。

デジタル・コンピュータに入るすべての情報はビットの形をとらなければならないため、心のコンピュータ・モデルは次のことを前提する。世界に関連するすべての情報、知的振舞いの産出にとって本質的なものすべては、原理的には状況に依存しない確定的要素の集合として分析可能でなければならない。この前提を一言で言えば次のようになる。

存在するものは、互いに論理的に独立した

*6 記号論においても、例えば記号作用の内部で起こる価値 (日本を意味する日の丸に付随して起こるポジティブもしくはネガティブな価値判断など) については議論されるものの、具体的な記号そのものの価値 (A 君が描いた赤い丸は醜いなど) についてはほとんど議論されないようである。

*7 これに関連することとして、Gelb が提唱する「theory of writing」を指摘したい。Gelb は、文字の歴史を絵画的なものからアルファベットのなものへの進化と捉えた上で、音声言語に限りなく近い IPA の発音記号的なものになっていくのが「書記の理論」であるとしている ([1])。この Gelb の理論はデリダによって批判されている ([3])。

事実の集合である。([4], p. 276)

言うまでもなくコンピュータが離散機械である以上、そこで扱われるデータ、概念、知識などもまた離散的でなければならず、コンピュータ処理による恩恵を受けたいのであればそのような離散性を前提にモデル化するのは当然である。ドレイファスが批判するのは、コンピュータ処理そのものではなく、「道徳的、知的、実際的な不明確さを取り除く」ことを目標とした結果、世界は離散的要素に分析可能であるとの結論に至る「西洋文化に埋め込まれた哲学的伝統」である ([4], p. 365)。

先に見たように、*character* もまた「書記言語の最小部品 (the smallest components of written language)」とされ、曖昧さを排除した—コンテキストの違いに左右されない—ものであるとされていることから考えて、この「伝統」の延長線上にあると言ってよいのではないかと思われる。

コンテキストに曖昧性がないというのは、音声言語の特徴でもある。すなわち、書かれたテキストがその物質性によって時空を超え、それが書かれた時点のコンテキストとは異なるコンテキストにおいて読まれることがある (現代人が古典を読むように) のとは異なり、音声言語は発話時のコンテキストを出ることはないからである。その意味で、*character* の列によるテキストが音声言語的な「logical order」であるという Unicode の規定は示唆的である。ジャック・デリダがプラトン以来の西洋哲学を「音声中心主義」「ロゴス中心主義」と批判し、そこで隠蔽され続けてきたエクリチュールの独自性と復権を唱えたこと ([3]) は、*character* が西洋哲学の「伝統」を継承しているという上の予想の傍証となろう*8。

3.2 字書の形而上学

これまでの議論を通じてわかるように、*character* はコンテキストに応じて様々に変化する文字に先行

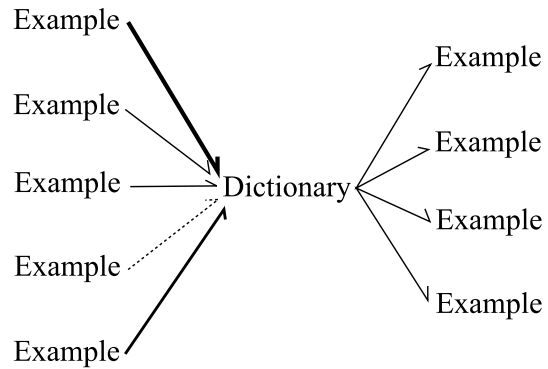


図1 用例と字書の関係

して存在する、言わば文字の多様性、多義性を支える基盤である。一方、デリダが主張したエクリチュールの理論とは、文字言語の多義性に先行する散種 (dissémination) 性である。すなわち、文字が複数のコンテキストを通過することで用例が生じ、その結果文字が各コンテキストに多重に所属することによって、多義性が事後的に獲得される、ということである。この問題をわかりやすくするために、字書と用例との関係について考えてみたい (図1)。

字書と用例のどちらが先かと問われれば、言うまでもなく用例が先である。しかしながら、字書の内部においては常に用例は親字の後に続いている。言わば親字の多義性を表すものとして用例が挙げられているのである (正字、異体字などの分類も同様である)。そして我々は字書を使って読み書きをし、時に我々は字書をもとに誰かの用例を「間違いである」とすることもある。言い換えれば、書記言語をめぐる我々の活動の多くは、図1の右半分に対応するものである場合が多い。

しかしながら—そして当然のことながら—字書に載っていない用例に遭遇することは往々にしてある。中国では新生児の名前をつけるために新たな漢字が作る習慣が続いているし、古文献を扱う文献学者は日常的に字書以前の用例と格闘している。言わば、図1の左端が増える場合である。このような字書の外側、字書以前で起こっていることに対して、現在の符号化文字集合モデルでは対応することができない。しかし、これらの用例に基づいて新しい字

*8 [7]において、*character* 概念とアリストテレスの本質主義との類似性、およびデリダのエクリチュール理論に基づく符号化文字集合モデルに対する批判について述べたので参照されたい。

書が作られれば、やはり用例と親字との先行性が逆転してしまうのである。

character 概念が依拠しているのも、この字書のヒエラルキーであると言ってよいのではなかろうか。そして、Unicode がその開発において、ローカルコードのみならず康熙字典などの権威ある古典を典拠としているのも、*character* の性質を端的に表していると思われる。

4 まとめにかえて

以上、雑駁ではあったが、Unicode の *character* についてその一端を考察した。

コンピュータが英語圏で発明され、コンピュータによる書記活動もまたアルファベットによって始まって、そしてそれが全世界に広がったことを考えれば、文字コードの概念が *character* という形に昇華して言ったのは必然であるようにも思われる。

確かに、限られたコンピュータ資源の中で文字処理を行おうとするのであれば *character* のような抽象化は効率的であるし、それ以前に文字の一面を表現する上で自然な選択肢であるようにも思われる。しかしながら、現在のようにコンピュータ資源、ネットワーク資源が潤沢となった状況においては、より文字の実体に近い、用例中心の文字処理モデルがあってもよいのではなかろうか。筆者も参加している CHISE プロジェクト^{*9}が提案する Chaon モデルは、符号化文字集合モデルを克服する新しい文字モデルとして大いに意義があると思っている ([7])。本稿の考察と併せて、ご叱正いただければ幸いである。

参考文献

- [1] Ignace J. Gelb. *A Study of Writing*. University of Chicago Press, revised edition, 1963.
- [2] The Unicode Consortium. *The Unicode Standard, Version 4.0*. Addison-Wesley, Boston,

2003.

- [3] ジャック・デリダ. 根源の彼方へ グラマトロジーについて. 現代思潮社, 1972. 足立和浩訳, 原著 1967.
- [4] ヒューバート・L. ドレイファス. コンピュータには何ができないか — 哲学的人工知能批判—. 産業図書, 東京, 1992. 黒崎政男・村若修訳, 原著 1979.
- [5] 川幡太一. 新 ISO/IEC 10646 と Unicode の漢字を検証する. 漢字文献情報処理研究, Vol. 2, , 2001.
- [6] 守岡知彦ほか. CHISE Project. 漢字文献情報処理研究, Vol. 4, , 2003.
- [7] 師茂樹. Surface or essence: Beyond the coded character set model. 京都大学 21 世紀 COE プログラム「漢字文化の全き継承と発展のために」, 書体・組版ワークショップ (2003 年 11 月) のための論文. 近日刊行予定.
- [8] 師茂樹. GB18030 とは何か 大陸の戦略. 漢字文献情報処理研究, Vol. 2, , 2001.
- [9] 師茂樹. Unicode 4.0. 漢字文献情報処理研究, Vol. 4, , 2003.

^{*9} <http://www.kanji.zinbun.kyoto-u.ac.jp/projects/chise/>、<http://cvs.m17n.org/chise/>、<http://mousai.as.wakwak.ne.jp/projects/chise/>。[6] 参照