外字と異体字

京都大学人文科学研究所
安岡孝一

1　はじめに

コンピュータによる漢字データ処理の歴史は、1978 年の日本工業規格 JIS C 6226 (現 JIS X 0208) の制定に遡る。JIS C 6226 は、ASCII とその拡張法である ISO 2022 の流れを汲み、全ての文字に漢字コードという一意なコードを割り振ることで、漢字を扱う手法である。この直後に、国産初のワードプロセッサが開発され、コンピュータ上での漢字は全て漢字コードを用いて処理する、という考え方が一般的となった。

このような漢字コードを用いた漢字データ処理においては、ある「漢字集合」を最初に規定し、その「漢字集合」の各漢字に一意なコードを割り振ることによって、それらの漢字を扱うことになる。すなわちコンピュータ上における、漢字の入力・出力・検索いずれの局面においても、漢字コードというものによって必要な漢字を処理することになるのである。

しかし、ある文書において必要とされる漢字が、必ずしもこのような「漢字集合」に全て含まれているとは限らない。実際の文書において必要とされる漢字の全集合が、現時点では不明である以上、どのような「漢字集合」を規定しても、足りない漢字というものが常に存在しうるのである。そのような「漢字集合」に含まれていない漢字、すなわち通常の漢字コードで表すことのできない漢字、これが「外字」である。

これまでの「外字」処理は、漢字コードの空き領域を用いる方法が一般的であった。すなわち必要とする「外字」を、漢字コードの空いている場所に適宜割り当ててやる方法である。しかしながら、この方法では、漢字コードのどの場所にどの「外字」を割り当てるかは、その「外字」を使う人それぞれが勝手に定めてしまうため、同じ「外字」であっても文書によって違う漢字コードに割り当てられている、という不都合が起こってしまうのである。つまり「外字」の出力はできても、検索は不可能というのが実態であった。

これに対し本稿では、漢字コードの空き領域に「外字」を割り当てるのではなく、漢字コードで既に使用されている領域に「外字」を割り当てる手法を提案する。この手法により、従来は検索することができなかった「外字」を、その「外字」の「異体字」を用いて検索することが可能となる。以下、次章で本稿のアイデアのガイドライン、第 3 章で PDF (Portable Document Format) [1] による実際例について述べる。

2　異体字の漢字コードを用いた外字処理手法

手始めに、次ページ右上図に示すような文書を、何がしかの DTP ソフトで作ることを考えてみよう。

「橋」という漢字には、例えば Shift_JIS なら<8bb4>という漢字コードが割り当てられているので、Shift_JIS 系の DTP ソフトでは、<8bb4>を用いて「橋」という漢字を扱うことになる。これは文書中に現れるどの「橋」でも同じで、次ページ右上の文書における 2 つの「橋」のいずれも、<8bb4>という漢字コードで扱われることになる。

また、そうすることによって、<8bb4>という漢字コードで、「橋」という漢字を検索することが可能になっているのである。

では「高」はどうであろうか。「高」という漢字には、例えば Shift_JIS なら<8d82>という漢字コードが割り当てられているので、Shift_JIS 系の DTP ソフトでは、<8d82>を用いて「高」という漢字を扱うことになる。これは「高」であっても「**高**」であっても同じで、実際、この文書における「高」と「**高**」は、普通の DTP ソフトであれば、同一の漢字コードであらわされているはずである。これによって、「高」でも「**高**」でも、同一の<8d82>によって検索することが可能となっているのである。この結果、右の文書上で「高橋」という文字列を検索すると、「高橋」と「**高**橋」の両方にヒットすることになるわけである。

これを実装の面から考えると、「高」と「**高**」という、アウトライン上は全く異なる 2 つの字形が、<8d82>という 1 つの漢字コードで処理されていることになる。すなわち、現時点の DTP ソフトにおいてもすでに、<8d82>という漢字コードに、複数の字形を対応させることが可能になっているのである。では、これを応用して、<8d82>に「髙」という字形を対応させることができれば、その結果はどうなるだろうか?

左図が予想される結果である。この文書において「高」にも「髙」にも同じ<8d82>という漢字コードが用いられていれば、「高橋」という検索文字列に対して、「高橋」と「髙橋」の両方がヒットするはずである。またそれは「髙」という「外字」に対して、その「異体字」を用いた検索が、現実に可能となることを意味している。

では、実際の DTP の世界では、本当にこのようなことが可能なのであろうか? 次章では PDF を例に見ていくことにしよう。

3　異体字の漢字コードを用いた外字処理手法の実際例

PDF においては、全ての漢字に CID という漢字コードが振られており、内部ではそれを用いて漢字処理をおこなっている [2]。日本語向けには Adobe-Japan1 [3] という CID 集合が準備されており、JIS X 0208 の文字が全て含まれている。また、日本語フォントとしては、Ryumin-Light と GothicBBB-Medium という 2 つのフォントが組み込みフォントとして準備されており、通常はこれらのフォントを用いて、文書を作成することになる。上の表に、881、1710、2036 という CID に対する、

CID	881	1710	2036
Ryumin-Light	と	橋	高
GothicBBB-Medium	**と**	**橋**	**高**

Ryumin-Light と GothicBBB-Medium の字形を示す。ただし日本語 PDF では、文字の指定にこれらの CID を直接は用いず、右に示す Identity-H、H、RKSJ-H、EUC-H、UniJIS-UCS2-H のような**エンコーディング**を用いることになっている。Identity-H は CID を 2 バイトでそのままあらわしたものであり、H はいわゆる JIS コード、RKSJ-H は Shift_JIS、EUC-H は EUC-JP、UniJIS-UCS2-H は Unicode を示している。なお、Adobe Acrobat 等での PDF 中の文字列検索は、内部的には CID 集合名 (Adobe-Japan1) と CID との組み合わせでおこなわれており、エンコーディングや使用しているフォントには無関係である。

CID	881	1710	2036
Identity-H	<0371>	<06ae>	<07f4>
H	<2448>	<3636>	<3962>
RKSJ-H	<82c6>	<8bb4>	<8d82>
EUC-H	<a4c8>	<b6b6>	<b9e2>
UniJIS-UCS2-H	<3068>	<6a4b>	<9ad8>

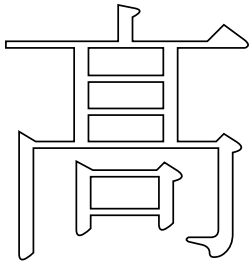
ここまでを踏まえて、まずは前ページ右上の文書を PDF で作成してみたところ、付録 1 のようになった。PDF 中、文字列を出力しているのは

```
/F1 100 Tf <07f4 06ae 0371> Tj
/F2 100 Tf <07f4> Tj
/F1 100 Tf <06ae> Tj
```

の部分であり、F1 に対するフォント Ryumin-Light とエンコーディング Identity-H の指定は 4 0 obj と 3 0 obj の部分で、F2 に対するフォント GothicBBB-Medium とエンコーディング Identity-H の指定は 7 0 obj と 6 0 obj の部分で、それぞれおこなっている。この結果、まずは Ryumin-Light で<07f4 06ae 0371>すなわち「高橋と」が出力され、次に GothicBBB-Medium で<07f4>すなわち「**高**」が出力され、最後に Ryumin-Light で<06ae>すなわち「橋」が出力されるわけである。また、この PDF を Adobe Acrobat で表示し、その中で「高橋」という文字列を検索したところ、「高橋」の部分と「**高**橋」の部分の両方にヒットした。

では、この PDF を改良して、前ページ左図の文書を作成するには、どうしたらいいだろう? つまるところ、GothicBBB-Medium の代わりに、CID 2036 に「髙」が入っているようなフォントを使うことができればよいのである。PDF には、組み込みフォント以外のフォントを、フォントのインストールなしに使用するための機構として、**フォント埋め込み**というものが準備されているので、それによって必要なフォントを PDF 内に埋め込めばよい。PDF 内にフォントを埋め込んでおけば、その PDF をどこに持っていったとしても、「髙」が必ず出力されるはずなのである。

この方針にしたがって、実際に作成してみたのが、付録 2 の PDF である。F2 に対するフォントとして、Hashigo-Taka というフォントを 7 0 obj で指定しており、さらにフォント中の字形を 10 0 obj の部分で定義している。CID 2036 の字形の定義には右に示すアウトラインデータを使用したが、PDF の埋め込みフォントには Compact Font [4] と Type 2 Charstring [5] というものを使用せねばならず、一目で見てわかるようなものになっていないのが残念である。この PDF を Adobe Acrobat で表示してみたところ、見事、前ページ左図が得られた。また、この PDF 中で「高橋」という文字列を検索したところ、当然のことながら「高橋」の部分と「髙橋」の部分の両方にヒットした。すなわちこれによって、本稿の目的である**「異体字」で検索できる「外字」**が、PDF 上では実現可能であると確認できたわけである。



4　おわりに

漢字コードで既に使用されている領域に、フォント埋め込みという手法で「外字」を割り当て、「外字」の「異体字」を用いて検索可能とする手法を提案した。また提案にしたがって、実際にPDFで例を作成し、本稿の手法が現実的に実現可能であることを示した。この結果は、本稿で提案するような「外字」が、実際にインターネット上で文字化けなく流通可能であり、またそれらをインターネット上で検索可能であることを示している。ただし、本稿のPDFはあくまで手作業で作成したものであり、Adobe Acrobat Distillerですら、このテのPDFを出力してはくれない。今後、本稿の手法を誰でもが簡単に使えるようなツールの開発が、必要となるだろう。

参考文献

[1] PDF Reference Second Edition — Adobe Portable Document Format Version 1.3, Addison-Wesley, March 2000.

[2] Adobe CMap and CID Font File Specification, Adobe Technical Specification #5014, October 1996.

[3] Adobe-Japan1-4 Character Collection for CID-Keyed Fonts, Adobe Technical Note #5078, August 2000.

[4] The Compact Font Format Specification, Adobe Technical Note #5176, March 2000.

[5] The Type 2 Charstring Format, Adobe Technical Note #5177, March 2000.

付録1　PDFリスト「高橋と**高**橋」

```
%PDF-1.3
% Hashigo-Taka Example 1 by Koichi Yasuoka

1 0 obj << /Length 89 >>
stream
BT
50 500 Td
/F1 100 Tf <07f4 06ae 0371> Tj
/F2 100 Tf <07f4> Tj
/F1 100 Tf <06ae> Tj
ET
endstream
endobj

2 0 obj <<
/ProcSet [/PDF /Text]
/Font <<
/F1 3 0 R
/F2 6 0 R
>> >> endobj

3 0 obj <<
```

```
/Type /Font
/Subtype /Type0
/BaseFont /Ryumin-Light-Identity-H
/Encoding /Identity-H
/DescendantFonts [4 0 R]
>> endobj

4 0 obj <<
/Type /Font
/Subtype /CIDFontType0
/BaseFont /Ryumin-Light
/FontDescriptor 5 0 R
/CIDSystemInfo <<
/Registry (Adobe)
/Ordering (Japan1)
/Supplement 0
>>
/DW 1000
>> endobj

5 0 obj <<
/Type /FontDescriptor
/Ascent 723
/CapHeight 709
/Descent -241
/Flags 6
/FontBBox [-170 -331 1024 903]
/FontName /Ryumin-Light
/ItalicAngle 0
/StemV 69
>> endobj

6 0 obj <<
/Type /Font
/Subtype /Type0
/BaseFont /GothicBBB-Medium-Identity-H
/Encoding /Identity-H
/DescendantFonts [7 0 R]
>> endobj

7 0 obj <<
/Type /Font
/Subtype /CIDFontType0
/BaseFont /GothicBBB-Medium
/FontDescriptor 8 0 R
/CIDSystemInfo <<
/Registry (Adobe)
/Ordering (Japan1)
/Supplement 0
>>
/DW 1000
>> endobj

8 0 obj <<
```

```
/Type /FontDescriptor
/Ascent 723
/CapHeight 709
/Descent -241
/Flags 6
/FontBBox [-170 -331 1024 903]
/FontName /GothicBBB-Medium
/ItalicAngle 0
/StemV 69
>> endobj

9 0 obj <<
/Type /Catalog
/Pages 10 0 R
>> endobj

10 0 obj <<
/Type /Pages
/Kids [11 0 R]
/Count 1
/MediaBox [0 0 612 792]
>> endobj

11 0 obj <<
/Type /Page
/Parent 10 0 R
/Resources 2 0 R
/Contents 1 0 R
>> endobj

xref
0 12
0000000000 65535 f
0000000053 00000 n
0000000192 00000 n
0000000268 00000 n
0000000400 00000 n
0000000584 00000 n
0000000758 00000 n
0000000894 00000 n
0000001082 00000 n
0000001260 00000 n
0000001311 00000 n
0000001395 00000 n
trailer
<<
/Size 12
/Root 9 0 R
>>
startxref
1478
%%EOF
```

付録2　PDFリスト「高橋と髙橋」


```
%PDF-1.3
% Hashigo-Taka Example 2 by Koichi Yasuoka

1 0 obj << /Length 89 >>
stream
BT
50 500 Td
/F1 100 Tf <07f4 06ae 0371> Tj
/F2 100 Tf <07f4> Tj
/F1 100 Tf <06ae> Tj
ET
endstream
endobj

2 0 obj <<
/ProcSet [/PDF /Text]
/Font <<
/F1 3 0 R
/F2 6 0 R
>> >> endobj

3 0 obj <<
/Type /Font
/Subtype /Type0
/BaseFont /Ryumin-Light-Identity-H
/Encoding /Identity-H
/DescendantFonts [4 0 R]
>> endobj

4 0 obj <<
/Type /Font
/Subtype /CIDFontType0
/BaseFont /Ryumin-Light
/FontDescriptor 5 0 R
/CIDSystemInfo <<
/Registry (Adobe)
/Ordering (Japan1)
/Supplement 0
>>
/DW 1000
>> endobj

5 0 obj <<
/Type /FontDescriptor
/Ascent 723
/CapHeight 709
/Descent -241
/Flags 6
/FontBBox [-170 -331 1024 903]
/FontName /Ryumin-Light
/ItalicAngle 0
/StemV 69
```

```
>> endobj

6 0 obj <<
/Type /Font
/Subtype /Type0
/BaseFont /Hashigo-Taka-Identity-H
/Encoding /Identity-H
/DescendantFonts [7 0 R]
>> endobj

7 0 obj <<
/Type /Font
/Subtype /CIDFontType0
/BaseFont /Hashigo-Taka
/FontDescriptor 8 0 R
/CIDSystemInfo <<
/Registry (Adobe)
/Ordering (Japan1)
/Supplement 0
>>
/DW 1000
>> endobj

8 0 obj <<
/Type /FontDescriptor
/Ascent 723
/CapHeight 709
/Descent -241
/Flags 6
/FontBBox [-170 -331 1024 903]
/FontName /Hashigo-Taka
/ItalicAngle 0
/StemV 69
/CIDSet 9 0 R
/FontFile3 10 0 R
>> endobj

9 0 obj <<
/Filter /ASCIIHexDecode
/Length 518
>>
stream
010000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000010
endstream
endobj

10 0 obj <<
```

```
/Subtype /CIDFontType0C
/Filter /ASCIIHexDecode
/Length 799
>>
stream
0100040200010101145045424846422b4861736869676f2d54616b6100010101
24f81bf81c8d0c1efb3efbdffa94fa1b051c07f50c22f7010ff70711f7040c25
f7f60c2400030101060c2141646f62654a6170616e324861736869676f2d5461
6b612d496e7465726e616c00000007f40000000002010102ea0ef813ae15baf7
866c076e9c8dadaa1ef72807af938f986e9d61a6186f7005fb760650aa05fb86
076f9c8cadaa1e8b07f74c04f786fb05fb86f70506fb5ffba215f807f924fbb5
07798282791e3c0681858b869586bf7f18998794817e1a818a9186951b9b8a98
919595089e9b94a1a81a8af7b5b1928e986e9d1961a66f7005fb1d068cf7e005
f781069c928a937b9842c2184f5005fb9c0689e1ad9b8c9571901950978cfb15
05fc24068d7405f784068dfbe005fb1e0651ae05fc46076f9c8cadaa1e8b07f8
61f8f2152dfb9ee9f79e07fb0a042efb9ee907f79e8a05f7830429fb9eedf79e
070e0001010117f81d0c261ea001ff8b8b1ea001ff8b8b0c0797f811127c9af8
ecbd068c0c11fa7c14
endstream
endobj

11 0 obj <<
/Type /Catalog
/Pages 12 0 R
>> endobj

12 0 obj <<
/Type /Pages
/Kids [13 0 R]
/Count 1
/MediaBox [0 0 612 792]
>> endobj

13 0 obj <<
/Type /Page
/Parent 12 0 R
/Resources 2 0 R
/Contents 1 0 R
>> endobj

xref
0 14
0000000000 65535 f
0000000053 00000 n
0000000192 00000 n
0000000268 00000 n
0000000400 00000 n
0000000584 00000 n
0000000758 00000 n
0000000890 00000 n
0000001074 00000 n
0000001280 00000 n
0000001873 00000 n
0000002772 00000 n
```

```
0000002824 00000 n
0000002908 00000 n
trailer
<<
/Size 14
/Root 11 0 R
>>
startxref
2991
%%EOF
```