Unicode と XSL によるアラビア語の組版

2004 年 3 月 1 日、2005 年 1 月 3 日改訂

アンテナハウス株式会社

Table of Contents

はじめに 3

Unicode のアラビア文字 4

アラビア文字（U+0600 から U+06FF） 4

アルファベット 5

アラビア文字の拡張 5

主要 3 言語のアルファベット 6

アラビア文字の字形処理 8

アルファベットの接続による字形変化 8

XSL Formatter の字形処理 9

アラビア文字のリガチャ 10

母音記号の処理 11

母音記号とは 11

Unicode での扱い 12

Unicode アラビア文字の結合記号 12

主要な母音記号 12

その他の文字 14

句読点 14

文の終わりを示すもの 14

文の論理的区切りを示すもの 14

括弧類、引用符、その他 14

その他 15

数字 15

ジャスティフィケーション 16

双方向性 17

双方向性とは 17

方向整形コード 17

基本的な表示アルゴリズム 18

定義 18

双方向文字特性 19

埋め込みレベルの解決 20

解決したレベルの並び替え 23

形状（shaping） 24

準拠 24

参考資料 25

はじめに

この文書では、Unicode と XSL-FO を使ってアラビア語を組版する方法について検討する。多言語組版研究会のテキストとして用意するものなので、整理した結果を述べるだけでなく、問題点と思われること、不明な点についても記述する。

将来的には、Unicode と XSL-FO を用いてアラビア語、あるいは、アラビア語と日本語、英語などの多言語混植の組版を行おうとするユーザに対して、整理した知識を提供することを目的としたい。しかし、現時点ではそこまでの知識は著者にないので、あくまでも将来の課題である。(1)

Thomas Milo: *Authentic Arabic* によるとアラビア文字の世界は次の３つに分けることができる。

1. アラビア。ネイティブ・スピーカーが歴史的な言語と文字を使い続けているオリジナルのアラビアである。これは、地理的にはアラビア半島に一致する。
2. アラビア語を話す世界。アラビア語が他の言語と文字を置き換えた地域からなる。メソポタミア、レバント、北アフリカを含む。
3. アラビア文字を使う世界。自分達の歴史的な言語を使い続けているイスラム国家。この人たちは、イスラム文化が浸透するにつれて、自分達の書き方をアラビア文字の応用に置き換えた。

三番目の世界の言語にはペルシャ語（Farsi）、ウルドゥ語（Urdu）、Sindhi、Pashto、Kurdish、Kashmiri、Bauchi、Kazakh、Lahnda、Berber、Malay などがある。

Kamal Mansour によるとアラビア文字を使う主要 3 言語は次の通りである。

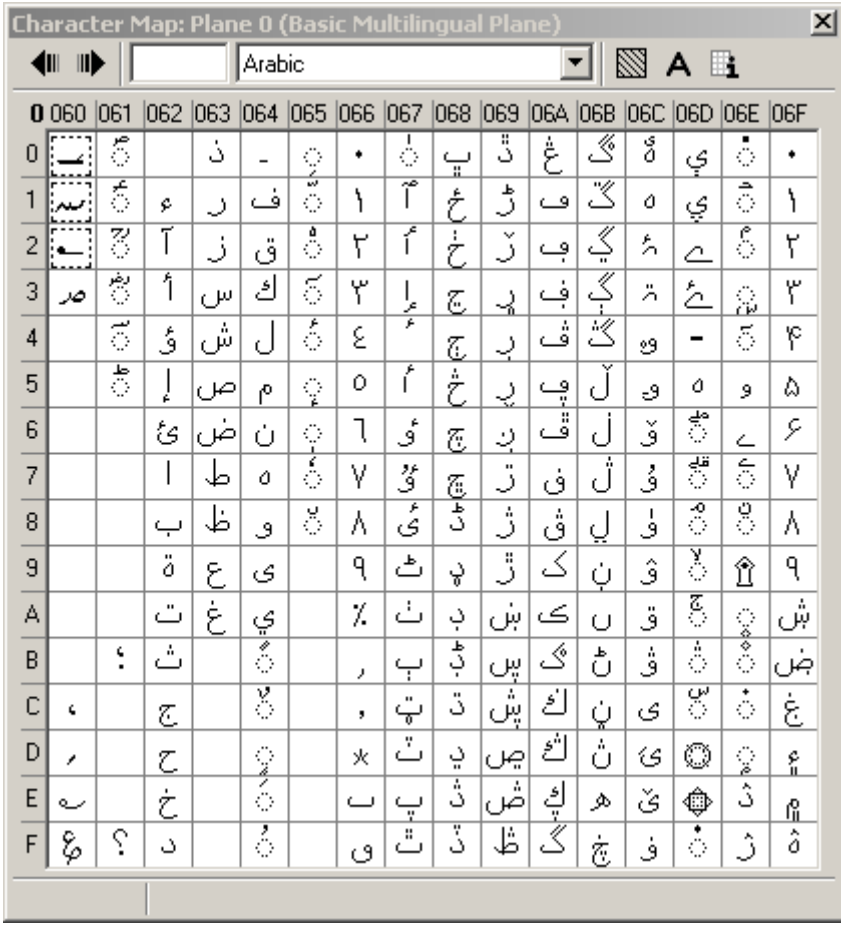
言語	人口	地域
アラビア語	2 億 5 千万人以上（アラブ・イスラム学院の Web ページによると日常的に話す人は 1.5 億人となっている）。	国連公用語のひとつ。サウディアラビア、クウェート、イラクなどアラブ連盟加盟国。
ペルシャ語	7 千万人以上	イラン・イスラム共和国
ウルドゥ語	1 億 8 千万人以上	パキスタン

(1)なお、この文書の前提となる知識としては、XML、Unicode の概要、XSL-FO が必要である。これらについては説明しない。

Unicode のアラビア文字

アラビア文字（U+0600 から U+06FF）

アラビア文字は隣の文字とどのように結合するかによって字形が異なるが、U+0600 から U+06FF の範囲では文字の意味毎にコード・ポイントを与えている。Unicode の文字コード表に掲載されているのは単独形で、Unicode3.2 では 208 文字、同 4.0 で 19 文字追加されて 227 文字となった。



Unicode 4.0 U+0600 から U+06FF

コード番号	説明
U+060C	カンマ
U+060D	日付区切り（Arabic Date Separator）
U+061B	セミコロン（Arabic Semicolon）
U+061F	疑問符（Arabic Question Mark）
U+0621 - U+064A	アルファベット（ISO/IEC 8859-6 Arabic Alphabet と同じ順序で並べている）
U+064B - U+0652	ISO 8859-6 による Points
U+0653 - U+0655	結合用の Maddah と Hamza
U+0656 - U+0658	その他結合用記号
U+0660 - U+0669	Arabic-Indic 数字
U+066A - U+066C	%記号、小数点記号、千単位の区切り
U+0671 - U+06D3	Extended Arabic（ほとんどがアラビア語以外で使う文字）
U+06F0 - U+06F9	Eastern Arabic-Indic 数字

この他、Unicode には次の 2 箇所にアラビア文字が定義されている。

アラビア文字表示形式 A（U+FB50 から U+FDFF）

この部分は互換性のために表示形（字形）を符号化したものであり Unicode4.0 の発行時点（2003 年 4 月）で、この表示形式すべてを実装したものはない。

アラビア文字表示形式 B（U+FF70 から U+FEFF）

このブロックには、アラビア文字判別記号を空白または TATWEEL（Kashida）と合成した形、基本アラビア文字の文脈による派生形、LAM-ALEF リガチャを含む。既存の標準と古い実装との互換性のためのものであり、U+0600 から U+06FF の文字で置き換えできる。U+FEFF はアラビア文字ではなく、ゼロ幅空白または BOM（バイト順マーク）である。

アルファベット

アラビア文字は子音アルファベットで記述するシステムで文章は右から左に書くが、数字は左から右に書く双方向の書き方である。アラビア文字のアルファベットは、3 層構造になっている。即ち基本の文字、判別点（diacritic dot）及び母音記号（vacalisation mark）である。母音記号は頻繁に使うわけではなく特別な利用に限られる。

なお、この 3 層で音を表すほかに、カリグラフィーのテキストでは、4 番目の飾りの記号がある。これは芸術的な創造物で、個々のカリグラファの流儀に属するものである。ミニチュアの文字を含み、必ずしも本文のテキストと意味的な関係があるわけではない。

アラビア文字の拡張

アラビア語

アラビア語の文字数は lam-alef リガチャと hamza を除いてアルファベットは 28 文字である。[(2)] 28 種類の文字の中で、基本形は 18 種類である。[(3)] 次の図のように基本形の上または下に 1 個から 3 個の判別点（diacritic dot）を付加することで、基本形を共有する文字ができる。判別点は文字の一部であって独立したコード・ポイントを与えられてはいない。9 世紀に、アラビア語特有の発音を区別するために、元になったアラム文字に判別点が追加された。[(4)]

アラビア語以外への拡張

アラビア文字はアラビア語以外の言語の表記にも使われる。その際、言語特有の発音を表記するために文字が拡張された。文字の拡張の二つの原則は、追加の判別点とミニチュア文字を使う方法である。

判別点はもともとアラビア語の発音を明確に表記するために導入されたが、アラビア語以外の発音を表記するために文字を追加する方法としても使われた。また、ミニチュア文字は、もともとはアラビア語の母音の表記や子音の拡張のために導入されたものだがアラビア語以外の言語を表記するために同じ方法が使われた。幾つかの非アラビア語では既存のアラビア文字のミニチュア文字を基本文字の上に乗せて新しい音を表記する文字を作った。[(5)]

次はアラビア文字 reh（ラー）を基本文字とする 4 つの文字の例である。左から reh（U+0631）、rreh（U+0691）、zain（U+0632）、jeh（U+0698）。reh と zain のみアラビア文字である。rreh はウルドゥ語、jeh はペルシャ語とウルドゥ語で使われる。

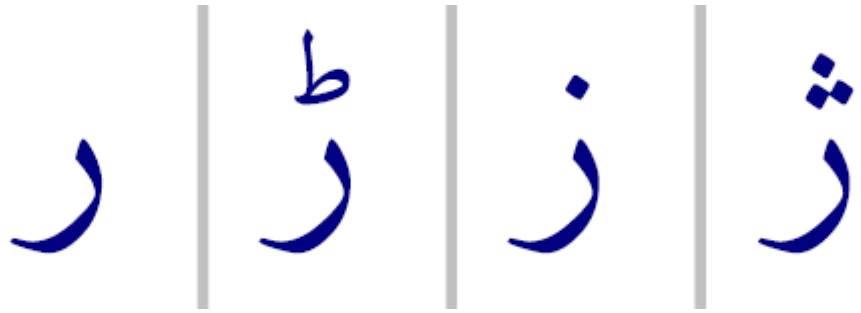
---

(2)「アラビア文字を書いてみよう読んでみよう」では、hamza と ta marbuta を含めて 30 文字としている。

(3)Authentic Arabic: a case study の Intorductio p.5 には、14 の基本形で 30 の子音を表すとある。

(4)Arabic Typography a comprehensive sourcebook p.87。但し、Authentic Arabic の説明は少し異なる。

(5)Arabic Typography a comprehensive sourcebook p.88。但し、Authentic Arabic では他に、間隔（Gap）、変形（Variation）を挙げている。



主要 3 言語のアルファベット

Unicode のアラビア文字には、アラビア語以外の言語を表記するために拡張された文字も一緒に符号化されている。

アラビア文字を使う 3 つの主要言語、アラビア語、ペルシャ語、ウルドゥ語について比較すると、ペルシャ語ではアラビア語にない音をもつため、アラビア語の文字を拡張した。ウルドゥ語では更に数多い音を表すための拡張がなされて文字の数が増えている。

文字を拡張する際に、Kaf 系の文字、Heh 系の文字、Yeh 系の文字は色々なバリエーションができたが、Unicode ではそれらのバリエーションが一緒に符号化されているため推奨されない文字がある。

Kamal Mansour は、新聞、詩、教科書、文献などを作成する際に使うための実用的なサブセットについて推奨している。次の表は 3 つの言語で使用する推奨アルファベットである。並び順はウルドゥ語の順番である。

Unicode	単独形	名称	日本語名称	アラビア語	ペルシャ語	ウルドゥ語	Hamza との結合有無	aspirate 形有無（ウルドゥ語）
U+0621	ء	HAMZA	ハムザ	○	○	○		
U+0627	ا	ALEF	アリフ	○	○	○	○	
U+0628	ب	BEH	バー	○	○	○		○
U+067E	پ	PEH			○	○		○
U+0629	ة	MARBUTA	ター・マルブータ	○	○	○		
U+062A	ت	TEH	ター	○	○	○		○
U+0679	ٹ	TTEH				○		○
U+062B	ث	THEH	サー	○	○	○		
U+062C	ج	JEEM	ジーム	○	○	○		○
U+0686	چ	TCHEH			○	○		○
U+062D	ح	HAH	ハー	○	○	○		
U+062E	خ	KHAH	ハー	○	○	○		
U+062F	د	DAL	ダール	○	○	○		
U+0688	ڈ	DDAL				○		○
U+0630	ذ	THAL	ザール	○	○	○		
U+0631	ر	REH	ラー	○	○	○		
U+0691	ڑ	RREH				○		○
U+0632	ز	ZAIN	ザーイ	○	○	○		
U+0698	ژ	JEH			○	○		
U+0633	س	SEEN	スィーン	○	○	○		

Unicode	単独形	名称	日本語名称	アラビア語	ペルシャ語	ウルドゥ語	Hamza との結合有無	aspirate 形有無（ウルドゥ語）
U+0634	ش	SHEEN	シーン	○	○	○		
U+0635	ص	SAD	サード	○	○	○		
U+0636	ض	DAD	ダード	○	○	○		
U+0637	ط	TAH	ター	○	○	○		
U+0638	ظ	ZAH	ザー	○	○	○		
U+0639	ع	AIN	アイン	○	○	○		
U+063A	غ	GHAIN	ガイン	○	○	○		
U+0641	ف	FEH	ファー	○	○	○		
U+0642	ق	QAF	カーフ	○	○	○		
U+0643	ك	KAF	カーフ	○	○	○		○
U+06A9	ک	KEHEH			○	○		○
U+06AF	گ	GAF			○	○	○	○
U+0644	ل	LAM	ラーム	○	○	○		
U+0645	م	MEEM	ミーム	○	○	○		
U+0646	ن	NOON	ヌーン	○	○	○		
U+06BA	ں	GHUNNA				○		
U+0648	و	WAW	ワーウ	○	○	○	○	
U+0647	ہ	HEH	ハー	○	○	○	○	
U+06D5	ہ	AE			○	○	○	
U+06BE	ھ	HEH				○		
U+0649	ى	ALEF MAKSURA		○	○	○		
U+064A	ي	YEH	ヤー	○	○	○	○	
U+06CC	ی	FARSI YEH			○	○		

注[(6)]

[(6)]Unicode のコード表では U+06D5（AE）は Uighur, Kazakh, Kirghiz となっている。

アラビア文字の字形処理

アルファベットの接続による字形変化

アラビア文字のアルファベットには大文字と小文字の区別はない。アラビア文字は筆記体で書くので、単語の中の文字は可能な限り結合する。アラビア文字は文字の接続状態により initial、medial、final、単独形（isolated、free standing）の 4 つの字形をもつ。

アラビア語の場合、アルファベット 28 文字の中で 22 個は 4 つの位置に応じた 4 つの字形をもち、位置に応じて異なる字形を表示する。しかし、ا（Alef）、د（Dal）、ذ（Thal）、ر（Reh）、ز（Zain）、و（Waw）の 6 種類をベースとする文字は、単独形か、final の 2 種類の字形のみをもつ。Aleph Maqsura（U+0649 ى）, Teh Marbuta（U+0629 ة）という変形があり、これは単独形か final 形をもつ。アラビア語を表記するには、29 種類しかない文字に 130 のグリフセットが必要である。(7)

2 つの字形を持つ文字は単語の中でも次の文字に繋がらない。このため、単語の内部に小さな空白ができることがある。単語間にはより広い空白を置くがひとつの文字からなる単語や句（lexcal unit）は次の単語と結合する。(8)

単語の単位は、カリグラフィーの書法に根をもつ。単語の区別は明確な単語間空白のみではなく次の単語の下に走る尻尾によることが多い。ラテン文字とは違って、文字の間の接続は水平とは限らないし、全ての文字が同じベースラインに並ぶとは限らない。ある文字の接続は垂直方法になり、文字が下から上に積み重なり、斜めの積み重なる多層のベースラインの並びを形成する。特に、元祖 Kufi 形式で書いたもの：Naskh、Ruqaa スタイルでは明確である。(9)

アラビア語アルファベット（フォントは Arabic Typesetting を指定）

上から独立形、initial、medial、final

ابتثجحخدذرزسشصضطظعغفقكلمنهوي

ابتثجحخدذرزسشصضطظعغفقكلمنهويـ

ابتثجحخدذرزسشصضطظعغفقكلمنهويـ

ابتثجحخدذرزسشصضطظعغفقكلمنهويـ

上は独立形で文字を並べたもの。下は単語の形。

الـتـمـويـل

التمويل

単語の中に後ろに繋がらない文字があるとき、単語の途中で切れる。

جامع حديد ذلك عربي زينة

単語の最後の文字の前に後ろに繋がらない文字があるとき最後の文字が独立形になる。

باب غيرك مشغول

(7)Arabic Typography a comprehensive sourcebook p. 100

(8)The world's writing, p.559 による。

(9)Arabic Typography a comprehensive sourcebook p. 99

XSL Formatter の字形処理

XSL Formatter はアラビア文字の接続による字形変化は組版エンジンが自動的に処理をする。従って、ユーザは字形変化について気にする必要はない。これは Unicode の Cursive 接続に関する仕様を使って行う。具体的には次のように行う。

Unicode の文字データベース

Unicode ではすべてのアラビア文字を次表の 6 つの結合クラスに分類している。詳細なデータは Unicode 文字データベースの ArabicShaping.txt にある。

結合クラス	記号	内容
右結合	R	ALEF, DAL, THAL など
左結合	L	なし
両結合	D	BEH, TEH, THEH, JEEM など
結合を引き起こす文字	C	ゼロ幅結合子（ZeroWidthJoiner U+200D）と TATWEEL（U+0640)
非結合	U	ゼロ幅非結合子（ZeroWidthNonJoiner U+200C）と他の結合クラスに含まれると明記されているものを除く全ての空白文字
透明	T	Transparent characeter。全ての非間隔文字と多くのフォーマット制御文字

結合による字形の入れ替え方法

両結合、左結合、結合を引き起こす文字を集約したものが「右結合を引き起こすクラス」、両結合、右結合、結合を引き起こす文字を集約したものが「左結合を引き起こすクラス」である。

アラビア文字の字形は単独形、右結合形、左結合形、両結合形の 4 つのタイプになる。実際に表示する字形の決定方法は例えば次のような規則で表される。

1. 右結合クラス文字 A の右側に右結合を引き起こす文字がある場合、文字 A の字形は右結合形となる。
2. 両結合クラスの文字 B の右側に右結合を引き起こす文字が**あり**、左側に左結合を引き起こす文字が**ある**場合、文字 B の字形は両結合形となる。
3. 両結合クラスの文字 B の右側に右結合を引き起こす文字があり、左側には左結合を引き起こす文字が**ない**場合、文字 B の字形は右結合形となる。
4. 両結合クラスの文字 B の右側に右結合を引き起こす文字が**なく**、左側に左結合を引き起こす文字がある場合、文字 B の字形は左結合形となる。
5. それ以外は単独形となる。例えば非結合クラスの文字は、隣の文字との結合を遮断する。

幅を持たずに結合のみに影響を与える文字として、ゼロ幅非結合子（ZeroWidthNonJoiner U+200C）とゼロ幅結合子（ZeroWidthJoiner U+200D）がある。

アラビア文字とアラビア文字の間に、空白を置けば空白文字は非結合文字なので、上のルール 5 でアラビア文字の字形は単独形になる。アラビア文字の前後をゼロ幅非結合子で囲えば単独形になる。文字の前にゼロ幅非結合子、文字の後ろにゼロ幅結合子を置けば initial 形になり、文字の前にゼロ幅結合子、文字の後ろにゼロ幅非結合子を置けば final 形になる。

ゼロ幅非結合子とゼロ幅結合子による hah（U+062D）の字形変化

```
<fo:block-container writing-mode="rl-tb"  inline-progression-dimension="30mm" >
<fo:block font-size="18pt" font-family="Tahoma">
&#x200c;&#x062D;&#x200c; &#x200c;&#x062D;&#x200d; &#x200d;&#x062D;&#x200d;
&#x200d;&#x062D;&#x200c; </fo:block>
</fo:block-container>
```

右から順に独立系、initial、medial、final になる（Arabic Typesetting フォント）。

ﺣـﺤﺣ

アラビア文字のリガチャ

アラビア文字を手書きする場合はリガチャが頻繁に起きる。ل（Lam）とا（Alef）のリガチャﻻは手書きでも印刷でも必須とされている。次の例は Arabic Typesetting フォントによる。

ل+ا⟵لا، ك+ل+ا+م⟵كلام، ا+ل+أ+و+س+ط⟵الأوسط

Lam と Alef 以外のリガチャと文脈依存形はフォントとアプリケーションに依存するオプションである。

「Authentic Arabic: a case study」によると、「アラビア文字のリガチャに関しては業界内に誤った考え方がもたれている。ラテン文字のリガチャは、少数の問題のある文字の組み合わせを美的に解決する方法である。しかし、アラビア文字の場合は文字の接続は例外ではなく規則である。適切にデザインしたアラビア文字では各文字は、あらゆる連続する組み合わせ毎に異なる外見となる。このため、小さなコード空間では表せない。Unicode のアラビア文字では最初に 2,000 のリガチャが提案されたが、政治的に 400 に減らされた。これが、Arabic Representation Forms A である。しかし、このためフォントの開発者は縮小された 400 のリガチャだけサポートすれば良いという誤った概念、それで権威付けされるという問題を生んでいる。]

例えば、「Writing Arabic」には、Kaf が次に続く文字によっては二重文字になって字形が変わることが乗っている。

A. kaaf

(i) Before ا, ب (m. and f.), ل (m. and f.).

Examples:-

1. كَاتِبٌ　2. كَفٌّ　3. كَلَّفَ

كاتب　كفّ　كلّف

kaatibun.　ka kaffin.　kallafa.

(ii) Before the remaining letters.

Examples:-

5. كَتَبَ　6. كُحْلٌ　7. كَذِبٌ

كتب　كحل　كذب

kataba.　kuḥlun.　kaðibun.

母音記号の処理

母音記号とは

セム系の子音言語では、書き方は、話し方の模倣ではなく、音や意味の判断は大いに文脈に繋がっている。書き方というのはコード化であり、読み手が鍵を知っていることを前提とする。

母音記号（vocalization marks アラビア語では Tashkil）はテキストの中で文字の上か下に付けて短母音、二重母音、子音の声門閉鎖音化を示す。母音記号は古代シリア語から導入され変遷した。Arabic Typography a comprehensive sourcebook によると、現在は、7 つの基本形に基づく 11 種類がある。

Fathah（U+064E）

短母音 a を表す。常に文字の上に配置する短い斜線

Tanwin Fathah（U+064B）

短母音 an を表す。常に文字の上に配置する短い二重斜線。Unicode では、Arabic Fathatan という名前がついている。

Kasrah（U+0650）

短母音 i を表す。常に文字の下に配置する短い斜線

Tanwin Kasrah（U+064D）

短母音 in を表す。常に文字の下に配置する短い二重斜線。Unicode では、Arabic Kasratan という名前がついている。

Dammah（U+064F）

ミニチュアの Waw（U+0648 و）。短母音 u を表す。常に文字の上に置く。

Tanwin Dammah（U+064C）

ミニチュアの二重 Waw（وو）。短母音 un を表す。常に文字の上に置く。Unicode では、Arabic Dammatan という名前がついている。[10]

Sukun（U+0652）

ミニチュアの o で短母音が無いことを示す。常に文字の上に置く。

Shaddah（U+0651）

ミニチュアの Sin（U+0633 س）の尻尾の無い形。子音が二重化されストレスがあることを示す。全ての短母音記号（6 種類）と組み合わせることができる。短母音記号が Shadda と組み合わせる時、Shadda を文字として短母音記号を Shadda の上下に組み合わされ配置する。ふたつの組み合わせがセットで常に文字の上に置かれる。[11]

Hamza（U+0654、U+0655）

ミニチュアの Ayn（U+0639 ع）尻尾の無い形。声門閉鎖音（glottal stop）を示す。母音記号としては、必ず、長母音すなわち Alef（a）、Waw（u）、Yeh（i/y）と組み合わされる。単語の中間では、長母音の後ろで声門閉鎖音を示すために文字の上に置かれる。Alef の下に置かれる時は、短い声門 i 母音（a short glottal *i* vowel）を示す。大きな Hamza は単独形で単語の末尾に置かれ文字と考えられ、別の文字コード U+0621 ء を与えられている。

Maddah（U+0653）

二重母音を表す。Alef の上に置かれて、a の音を伸ばすことを示す。

Waslah

Sad（U+0635 ص）の尻尾の無い形から由来する。常に Alef の上に置かれて音価を持たないことを示す。

---

[10] ArabicTypography では Waw ふたつになっている。しかし、Unicode では一筆書きの字形になっている。

[11]「Harakat is broken in Qt」の議論を読むと、Shadda と Kasra をセットにする場合、Kasra を Shadda の下に置き、Shadda と Kasra を共にベース文字に上に置くのと、Shadda をベース文字上、Kasra をベース文字の下に置くのは両方許されるようである。

Unicode での扱い

Unicode アラビア文字の結合記号

Unicode の仕様書には、次のように記述されている。「母音記号あるいは他の記号は「harakat」と呼ぶ結合記号（Combining Mark）で符号化される。[(12)]」（Unicode 4.0 p.196）

アラビア文字の中で結合記号として定義されているのは次のものである。

分類	Unicode 番号	意味
Honorifics	U+0610 ～ U+0614	Unicode4.0 で追加された文字である。人の状態を表現する句を意味し、アラビア文字を書く世界では広く使われている。大部分は宗教上の意味をもつ。これらの記号は、ひとつの基底文字に結合するのではなく、単語レベルの結合文字である。名前の中の文字の形とカリグラフィのスタイルに依存して、名前の中のどこかの一つの文字に適用される。Unicode の正規化アルゴリズムはこのような単語レベルの結合文字を単語の最後に移すことはしないことに注意せよ。
Koranic annotation sign	U+0615	Unicode4.0 で追加された文字である。イラン、パキスタンで発行されるコーランの中で休止位置を示す。
Points from ISO 8859-6	U+064B ～ U+0652	母音記号 Fathatan, Dammatan, Kasratan, Fatha, Damma, Kasra, Shadda, Sukun
Combination madda hamza	U+0653 ～ U+0655	Madda above, Hamza above, Hamza below
Other combination marks	U+0656 ～ U+0658	Subscript Alef, Inverted Damma, Noon Ghunna
Point	U+070	Superscript Alef
Koranic annotation signs	U+06D6 ～ U +06DC, U+06DF ～ U+06E4, U+06E7 ～ U+06E8, U+06EA ～ U+06ED	コーラン用の記号

主要な母音記号

Arabic Typography a comprehensive sourcebook の挙げる 11 種類の母音記号のポート・ポイントは次の表の通りである。Hamza、Madda は Unicode では母音記号は結合記号（Combining Mark）として単独のコード・ポイントが与えられている。しかし、Waslah 記号には単独で文字コードを与えていない。U +0610 は似ているが、名前が「Arabic Sign Sallallahu Alayhe Wasallam （神の平和と祝福を彼の上に)」となっていて意味が異なる。また、Tahoma フォントには、Hamza、Madda はグリフがない。

Unicode	Arial	Arabic Typesetting	Times New Roman	名称
U+064B	ً	ً	ً	Fathatan タンウイーン・ファトハ
U+064C	ٌ	ٌ	ٌ	Dammatan タンウイーン・ダンマ
U+064D	ٍ	ٍ	ٍ	Kasratan タンウイーン・カスラ
U+064E	َ	َ	َ	Fatha ファトハ
U+064F	ُ	ُ	ُ	Damma ダンマ
U+0650	ِ	ِ	ِ	Kasra カスラ
U+0651	ّ	ّ	ّ	Shadda シャッダ

[(12)]Harakat の意味はよくわからない。ここでは母音記号などの記号を結合記号として符号化することとして解釈する。

Unicode	Arial	Arabic Typesetting	Times New Roman	名称
U+0652	◌ْ	◌ْ	◌ْ	Sukun スクーン
U+0655	◌ٕ	◌ٕ	◌ٕ	Hamza Below 下につくハムザ
U+0653	◌ٓ	◌ٓ	◌ٓ	Madda マッダ
U+0654	◌ٔ	◌ٔ	◌ٔ	Hamza Above 上につくハムザ

Alef+Hamza（上、下）、Alef+Maddah、Waw+Hamza、Yeh+Hamza は、組み合わせ済み文字として単独のコード・ポイントもある。また、Alef+Waslah の組み合わせは単独の文字としてのみ定義されている。

Unicode	単独形	名称	意味
U+0622	آ	ALEF WITH MADDA ABOVE	a の長母音
U+0623	أ	ALEF WITH HAMZA ABOVE	
U+0624	ؤ	WAW WITH HAMZA ABOVE	
U+0625	إ	ALEF WITH HAMZA BELOW	
U+0626	ئ	YEH WITH HAMZA ABOVE	
U+0671	ٱ	ALEF WASLA	コーランで使うアラビア文字（Koranic Arabic）

その他の文字

句読点

アラビア語の句読点のシステムはラテン文字と共通である。現代のアラビア語で使われている句読点の記号は、ラテン文字の鏡像イメージである。主としてフランス語の句読点システムに従っている。(Arabic Typography a comprehensive sourcebook p.104)

文の終わりを示すもの

ピリオド

ベースライン上の点。考えまたはセンテンスの終わりを示す。

疑問符

ベースライン上の鍵状と点のマーク。右へ向いている（U+061F ؟）。質問を示す。

感嘆符

ラテンと同じ文字で用法も同じ。

省略記号

ラテンと同じ文字で用法も同じ。

文の論理的区切りを示すもの

コロン

ラテンと同じ。時間の区切り（15:30、1:15:30 など）にも用いる。

コンマ

二つの短いセンテンスの区切りを示す。Dammah との混同を避けるため上にはねる形状をもつ（U+060C ، ARABIC COMMA）。数字は、ラテン文字と同じように左から右へ書くので通常のコンマ（Decimal Comma）を数値の区切りとして使う。アラビア語では、Decimal Comma は数値の小数点として使い、千単位では空白を空ける。

セミコロン

ラテン文字を 180 度回転した形で上にはねる（U+061B ؛ ARABIC SEMICOLON）。

ダッシュ

アラビア語のタイプライターのダッシュは en ダッシュで em ダッシュの半分の長さ。Kashida として単語の中の特定の文字の接続を伸ばすためにも使う（Unicode では Kashida は、U+0640 TATWEEL として別のコードが与えられている）。

括弧類、引用符、その他

丸括弧

ラテンと同じ

矩形括弧

トップからボトムまでの垂直線と交わる短い水平線

引用符

Damma と区別するため特別にデザインされた文字を使う。ベースラインの上に置き、トップは Alef の高さ。開始引用符のトップが太く、ボトムは細くなる特別なデザイン。16 世紀に発明され、17 世紀まで頻繁に使われたが、今は、Gillmet を使う傾向がある。

二重小角括弧（Guillemets）

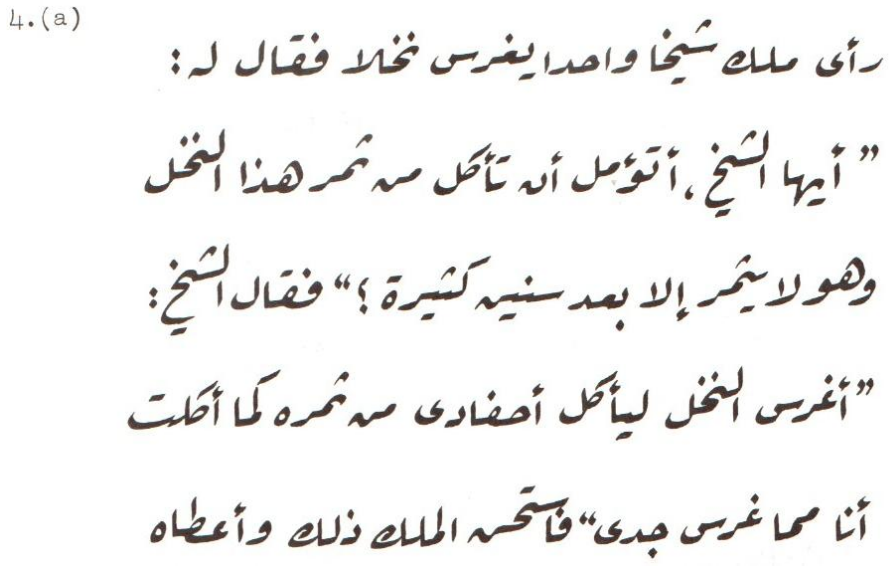
ベースラインの上にある二重のミニチュアの角括弧

括弧類はラテンと同じものであるが、テキストの進行方向に依存してグリフの選択が必要である。

その他

Virgule

逆スラッシュに似ている形状。or（例：and/or）、and or（例：black/white）、per（例：days/week）、分数（例：1/2）、改行、日付（例：11/1/1965）あるいは分離の記号として色々な所に使う。



句読点の使用例（「Writing Arabic」p.115）

数字

アラビア文字の範囲には、Arabic-Indic 数字（U+0660-U+0669）と Eastern Arabic-Indic 数字（U+06F0-U+06F9）が定義されている。

	0	1	2	3	4	5	6	7	8	9
Arabic-Indic	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
Eastern Arabic-Indic	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹

ジャスティフィケーション

次は、「Authentic Arabic: a case study」による。

ラテン文字ではジャスティフィケーションの方法に、無差別に微小な空白を入れるグローバルな方法と言語別に異なる規則に基づいて単語をハイフネートする特殊な方法の 2 種類ある。

イスラムのカリグラフィでは Keshideh という方法がある。これは、ペルシャとオットマン・トルコの言葉で「伸ばす」という意味。Keshideh は、ハイフンが言語に依存するようにタイプフェース依存である。Keshideh はある種の文字の組み合わせに対して他よりも優先度を高くする複雑なルールに基づいて配置する。このルールはカリグラフィのスタイルによって異なる。この結果はアラビア筆記体の種類によって異なる特徴的なものである。言い換えれば、Keshideh は微小な空白ではなく、ハイフネーションと同等である。Keshideh は、通常、単語または文字の複合物の中で一箇所以下。文字の中にはそれ自身で伸びた形状をもつものがあり、その場合は Keshideh は入らない。

アラビア語 1
axf:text-kashida-space="0%"

توفر العلامات الذكية الحساسة تجاه النصوص مدخلا سريعا للمعلومات المطلوبة عن طريق تنبيهك إلى الإجراءات المهمة مثل خيارات تنسيق المعلومات الملصقة، هناك جوانب جديدة لجزء المهام تجعلك قادرا على الحصول على الأدوات التي تحتاجها بضغطة واحدة على الفأرة وتساعدك على إيجاد الملفات وتنسيق المحتويات.

アラビア語 11
axf:text-kashida-space="100%"

توفـر العلامــات الذكيـة الحساســة تجـاه النصـــوص مدخــلا سريعـا للمعلومــات المطلوبـــة عـن طريــق تنبيهــك إلـى الإجـراءات المهمـة مثل خيـارات تنسيق المعلومـات الملصقة، هناك جوانب جديدة لجـزء المهام تجعلك قادرا على الحصول على الأدوات التي تحتاجها بضغطة واحدة علـى الفأرة وتساعدك على إيجاد الملفات وتنسيق المحتويات.

双方向性

双方向性とは

次の例では、日本語の中にアラビア文字の数字と文字・記号を埋め込んだ。この中で２番はアラビア文字の%記号が数字の右側に表示されている。これは、３番のように表示する必要がある。2 の%記号は、左から右へ既述する文字に続いて出現するため 3 番のように表示させるには、%記号の前に[右から左へ書く文字]であることを示すマークを入れて方向を制御する必要がある。このような問題を双方向性 BIDI という。

数字と文字

左から右に書く文脈での数字と文字
1. 日本語の 598B をアラビア語では٥٩٨بと書く。
2. 日本語の 64%をアラビア語では ٦٤٪と書く。（方向制御なし）
3. 日本語の 64%をアラビア語では٪٦٤と書く。（方向制御）

右から左に書く文脈での数字と文字
日本語では 598B←٥٩٨ب .1
日本語では 64%←٪٦٤ .2

双方向性はアラビア文字のみではなく、ヘブライ語のように右から左へ書き進める言語を処理するときにも共通の課題である。この節では双方向性について説明し、Unicode の双方向アルゴリズム（The Bidirectional Algorithm: UAX#9）について簡単に紹介する。

メモリ上の順序は論理順という。テキストを水平に表示するとき、大抵の言語は左から右に表示する。これに対し、アラビア文字やヘブライ文字は右から左に書く。Unicode の文字データベースには、書き進める方向（方向特性値）が定義されていて、大抵の場合、それで表示する方向が決まるが、左から右に書く文字と右から左に書く文字が混在した時、文字を表示する方向に曖昧さが生まれる。これを解決するための方法が双方向アルゴリズムである。

方向整形コード

暗黙の文字の方向性だけでは複雑にいり組んだテキストの表示方向を決定できない状況に備えて、Unicode は最小限の方向整形コード・セットを定義している。方向整形コードは表示の順序を決定するためのみに使用するためのものであり、それ以外の面では無視されねばならない。

方向整形コードには文字に定義された暗黙の双方向性に基づくアルゴリズムを変更する機能をもつ明示的なコード、および、暗黙の順序コードである right-to-left と left-to-right マークがある。これらのすべてのコードの効果は段落の中でのみ有効で、段落の区切で終了する。

また、方向特性には強いタイプと弱いタイプがある。左から右、右から左というような方向特性は強いタイプである。一方、数字につけられる方向特性は弱いタイプである。

分類		コード	値	名称	説明
明示的で強い方向性の文字	埋め込み	RLE	U+202B	Right-to-Left Embedding	続く文字を右から左へ埋め込まれたものとして扱う
		LRE	U+202A	Left-to-Right Embedding	続く文字を左から右へ埋め込まれたものとして扱う
	上書き	RLO	U+202E	Right-to-Left Override	続く文字を強い右から左方向文字として扱う
		LRO	U+202D	Left-to-Right Override	続く文字を強い左から右方向文字として扱う

分類		コード	値	名称	説明
	終了	PDF	U+202C	Pop Directinal Format	双方向性の状態を、直前の LRE、RLE、RLO、LRO の状態に戻す
暗黙で弱い方向マーク。範囲は局所的である。方向特性をもつ文字と同等に扱われるが、文字とは違い表示されない（幅がゼロの文字)。		RLM	U+200F	Right-to-Left Mark	Right-to-Left ゼロ幅マーク
		LRM	U+200E	Left-to-Right Mark	Left-to-Right ゼロ幅マーク

埋め込みコードの使用例。次のようにアラビア語の文章中に、英語の引用をするときに使う。

```
論理順  ARABIC [LRE]"english sentence"[PDF] ARABIC

表示          →→                  ←←
       CIBARA "english sentence" CIBARA
```

上書きコードは、パーツ番号の中のラテン文字や数字を強制的に右から左に表示する時などに使う。

```
論理順  ARABIC [RLO]PN-123[PDF] ARABIC

表示                    ←←
       CIBARA 321-NP CIBARA
```

基本的な表示アルゴリズム

Unicode BIDI の処理は、テキストのストリームをインプットして、次の 3 段階の処理を行う。

1. インプット・テキストを段落に分解する。
2. テキストの埋め込みレベルを解決する。文字の双方向特性と Unicode の方向整形コードを使って、解決済みの埋め込みレベルを決定する。
3. テキストを行に区切った後で、行単位で解決済みの埋め込みレベルを使ってテキストを表示のために並び替える。

処理単位は段落であって、他の段落の影響を受けることはない。

規範的定義とルール

番号	節
BDn	定義
Pn	段落レベル
Xn	明示的なレベルと方向
Wn	弱いタイプ
Nn	中立のタイプ
In	暗黙のレベル
Ln	解決したレベル

定義

- BD1. 双方向文字特性（bidirectional character types）は、各 Unicode 文字に割り当てられた値である。
- BD2. 埋め込みレベルは、テキストがどれだけ深くネストしているか、そのレベルのデフォルトの方向を示す。最小値はゼロ、最大値は 61。
- BD3. 現在の埋め込みレベルのデフォルトの方向を埋め込み方向といい、偶数なら L、奇数なら R である。例えば英語の段落では英語の埋め込みレベルは 0、それに埋め込まれたアラビア語のテキストは 1、さらにその中に埋め込まれた英語のテキストは 2 となる。上書きされていない限り、英

語や数のように右から左へ書くテキストは奇数になり、アラビア語（数字を除く）のように左から右へ書くテキストは偶数になる。

- BD4. 段落の埋め込みレベルはその段落のデフォルトの方向を決めるレベル。
- BD5. 段落の埋め込みレベルを、段落の方向という。あるいは、基本的な方向である。
- BD6. 方向上書き状態は、明示的な方向制御によって文字の方向特性が再設定されるかどうかを決定する。①中立、②文字が R に再設定される right-to-left、③文字が L に再設定される left-to-right、の 3 つの状態がある。
- BD7. レベルの連続は、同じ埋め込みレベルの文字の最長の列である。

次の例では、大文字は右から左へ書く文字、小文字は左から右へ書く文字である。空白は中立で THE CAR の間の空白は、周囲の文字の方向に影響を受ける。

```
Memory:           car is THE CAR in arabic
Character types:  LLL-LL-RRR-RRR-LL-LLLLLL
Resolved levels:  00000001111111000000000
```

中立の文字の回りに適切な方向マークを挿入することで中立な文字のレベルを変更できる。これが方向マークの使い方である。

双方向文字特性

Unicode 文字データベースには、文字の双方向特性が規定されている。要約すると次の表の通り。

分類	タイプ	説明	範囲
強い	L	左から右	LRM、アルファベット、シラビック、漢字、欧州・アラビア文字以外の数字
	LRE,LRO	左から右	LRE,LRO
	R	右から左	RLM、ヘブライ語のアルファベットと主要な句読点
	AL	右から左	アラビア文字、Thaana、シリア文字のアルファベット、それらの文字の主要な句読点
	RLE,RLO	右から左	RLE,RLO
弱い	PDF	方向性の回復	PDF
	EN	欧州の数字	欧州の十進数、東方 Arabic-Indic 系の十進数
	ES	欧州の数字区切り	スラッシュ
	ET	欧州の数字終了子	+記号、－記号、温度、通貨記号
	AN	アラビア数字	Arabic-Indic の十進数、アラビア語の十進・千単位の区切り
	CS	共通の数字区切り	コロン、コンマ、フル・ストップ、No-Break-Space
	NSM	非間隔記号	Unicode 文字データベースで Mn（非間隔記号）、Me（包含記号）とマークされた文字
	BN	中立の境界	上で明示された以外の整形、制御記号
中立	B	段落区切り	段落区切り、改行機能、上位の区切り（表のセルなど）
	S	セグメント区切り	Tab
	WS	ホワイトスペース	空白、数字用空白（Figure Space）、行の区切り、FF、一般的な句読点の空白
	ON	その他中立	その他の全ての文字。Object Replacement Character を含む。

この他、説明で使用する略称は次の通り。

記号	説明
N	中立あるいは区切り（B、S、WS、ON）

記号	説明
e	埋め込みレベルの方向に一致する方向タイプ（L か R)
sor	レベルの連続の前の位置に割り当てる方向タイプ（L か R)
eor	レベルの連続の後に割り当てる方向タイプ（L か R)

埋め込みレベルの解決

この解決処理では、文字の方向特性と明示的コードを用いて、解決済みのレベルのリストを作成する。次の 5 つのステップから成る。

1. 段落レベルの決定
2. 明示的な埋め込みレベルと方向の決定
3. 弱いタイプの決定
4. 中立のタイプの決定
5. 暗黙の埋め込みレベルの決定

段落レベル

- P1. テキストを段落に分割する。段落区切りは前の段落と一緒にする。段落毎に以下のルールを適用する。
- P2. 各段落の中で、L、AL、R のいづれかである最初の文字を探す。LRE、LRO、RLE、RLO のタイプの文字は無視する。これらのコードは段落の既定値の方向とは反対方向を設定するために用いられることが多いからである。
- P3. もし、P2 で AL または R の文字が見つかれば、段落の埋め込みレベルを 1 とし、それ以外の時は 0 とする。但し、もし上位のプロトコルで段落レベルを指定しているならば、P2、P3 のルールを適用する必要はない。

明示的なレベルと方向

次の X1 から X9 のルールを適用して、埋め込みコードまたは上書きコードから明示的なレベルを決定する。

明示的な埋め込み

- X1. 現在の埋め込みレベルを段落の埋め込みレベルに設定する。方向上書き状態を中立とする。文字に順次、X2 から X9 を適用して処理する。
- X2. RLE があったら、より大きな奇数の埋め込みレベルの中で最小値を計算し、現在の埋め込みレベルと方向上書き状態を記憶し、新しい埋め込みレベルを現在の埋め込みレベルとしてセットする。方向上書き状態を**中立**にリセットする。例えば、現在の埋め込みレベル 0 なら 1 、レベル 1,2 なら 3、レベル 3,4 なら 5 となる。
- X3. LRE があったら、より大きな埋め込みレベルのなかで最小の偶数値を計算し、現在の埋め込みレベルと方向上書き状態を記憶し、新しい埋め込みレベルを現在の埋め込みレベルとしてセットする。方向上書き状態を**中立**にリセットする。例えば、現在の埋め込みレベル 0、1 なら 2、レベル 2,3 なら 4、レベル 4,5 なら 6 となる。

明示的な上書き

明示的な上書きコードは、明示的な埋め込みコードと同様に埋め込みレベルをセットし、同時に、影響を受ける文字の方向特性を、上書き方向に変更する。

- X4. RLO があったら、より大きな奇数の埋め込みレベルの中で最小値を計算し、現在の埋め込みレベルと方向上書き状態を記憶し、新しい埋め込みレベルを現在の埋め込みレベルとしてセットする。方向上書き状態を **right-to-left** にリセットする。
- X5. LRO があったら、より大きな埋め込みレベルのなかで最小の偶数値を計算し、現在の埋め込みレベルと方向上書き状態を記憶し、新しい埋め込みレベルを現在の埋め込みレベルとしてセットする。方向上書き状態を **left-to-right** にリセットする。
- X6. RLE、LRE、RLO、LRO、PDF 以外の全てのタイプについて、現在の文字のレベルを、現在の埋め込みレベルにセットする。また、方向上書き状態が中立であれば、文字の方向特性は元のままとする。方向上書き状態が中立でない時は、現在の文字の方向特性を、方向上書き状態にリセットする。

方向上書き状態が中立であれば、アラビア文字は AL、ラテン文字は L、中立の文字は N のままであるが、方向上書き状態が R であれば文字の方向特性を R に、方向上書き状態が L であれば文字の方向特性を Lにする。

埋め込みまたは上書きの終了

現在の明示的なコードの影響範囲を終了するためのコードが一つある。また、全てのコードと記憶された状態は段落の終了で元に戻る。

- X7. PDF に対して、対応する埋め込みまたは上書きコードを決定する。対応する妥当なコードがあれば、最後に記憶された埋め込みレベルと方向上書き状態を回復する。
- X8. 段落の終わりで全ての明示的方向埋め込み、方向上書きは完全に終了する。段落の分割子は埋め込みには含めない。
- X9. RLE、LRE、RLO、LRO、PDF と BN コードを削除する。ゼロ幅ジョイナー、非ジョイナーは、隣接する文字の形状変化に影響する。但し、オリジナル状態での隣接である。
- X10. 次のルールは、同じレベルの文字の連続毎に適用する。各連続に対して、sor（start-of-level-run）と eor（end-of-level-run）のタイプを L か R に決定する。これは、おのおのの境界の上の二つのレベルの大きな方に依存する（段落の先頭または最後では基本埋め込みレベルと比較する）。もし、高い方のレベルが奇数であれば R で、それ以外は L である。ふたつの隣接するレベルの連続の先行する eor と後ろの sor は同じ値となる。

```
Levels: 0  0  0  1  1  1  2
Runs:  <-------><-------><-->
         (1)       (2)    (3)
Run 1 はレベル 0、sor は L、eor は R
Run 2 はレベル 1、sor は R、eor は L
Run 3 はレベル 2、sor は L、eor は L
```

弱いタイプの解決

弱いタイプについては一度に一つのレベルの連続毎に解決する。レベルの連続の境界で、境界の反対側のタイプが必要な時は、sor または eor を使う。非間隔記号は前の文字に基づいて決定する。

- W1. 各非間隔記号（NSM）を調べて、NSM のタイプを先行する文字のタイプに変更する。NSM がレベルの連続の先頭にあるときは、sor を使う。

```
sor が R と仮定する
AL NSM NSM => AL AL AL
sor NSM    => sor R
```

続けて、テキストを数字のために解析する。欧州の数字区切り、欧州の数字終了子、共通の数字区切りの方向タイプを、欧州の数字テキスト、アラビア文字の数字テキスト、あるいは他の中立テキストにセットする。テキストは既に方向上書きによってタイプを変更されているかもしれない。その時は、数字としては解析しない。

- W2. 欧州の数字が出現する毎に、最初の強いタイプ（R、L、AL、sor）が見つかるまで逆方向に探す。AL が見つかったら、欧州の数字のタイプをアラビア文字の数字に変更する。

```
AL EN    --> AL  AN
AL N EN  --> AN N AN
sor N EN --> sor N EN
L N EN   --> L N EN
R N EN   --> R N EN
```

- W3. 全ての AR を R に変更する。
- W4. ふたつの欧州の数字の間の欧州のひとつの区切り記号を欧州の数字に変更する。ふたつの同じタイプの数字の間のひとつの共通の区切り記号は、そのタイプに変更する。

```
EN ES EN  --> EN EN EN
EN CS EN  --> EN EN EN
AN CS AN  --> AN AN AN
```

・W5. 欧州の数字に隣接する欧州の終了子の並びを欧州の数字に変更する。

```
ET ET EN  --> EN EN EN
EN ET ET  --> EN EN EN
AN ET EN  --> AN EN EN
```

・W6. それ以外の場合、区切り記号、終了子は中立に変更する。

```
AN ET    --> AN ON
L ES EN  --> L ON EN
EN CS AN --> EN ON AN
ET AN    --> ON AN
```

・W7. それぞれの欧州の数字の各インスタンスから逆方向に強いタイプ（R、L、sor）が見つかるまで探す。もし、L が見つかったら、欧州の数字のタイプを L に変更する。

```
L N EN   --> L N L
R N EN   --> R N EN
```

中立のタイプを解決

ひとつの同一レベルの連続毎に、中立タイプを解決する。レベルの連続の境界で、境界の反対側のタイプが必要な時は、sor または eor を使う。

次の段階では、中立の方向の解決である。この段階で中立はすべて R か L になる。一般的に、中立は取り巻くテキストの方向を取る。矛盾が生じた時は、埋め込み方向を取る。

・N1. 両側のテキストが同じ方向を取っているなら、中立の並びはそれを取り巻くテキストの方向を取る。欧州の数字とアラビアの数字は、R であるかのように取り扱う。レベルの連続の境界では、sor と eor を使う。

```
R N R   --> R R R
L N L   --> L L L
R N AN  --> R R AN
AN N R  --> AN R R
R N EN  --> R R EN
EN N R  --> EN R R
```

・N2. 残りの中立は全部埋め込み方向を取る。

```
eor が L、sor が R と仮定する。

L N eor --> L L eor
R N eor --> R e eor
sor N L --> sor e L
sor N R --> sor R R
```

中立で区切られて、方向連続に埋め込まれた数字のリストは、連続の順序になる。

```
Strage : he said "THE VALUE ARE 123, 456, 789, OK".
Display: he said "KO ,789 ,456 ,123 ERA SEULAV EHT".
```

上の例では、数字の間のカンマと空白は、数字を無視して取り囲むテキスト（大文字なので、right-to-left）の方向となる。カンマは、数字の解析の項で示したように両側が数字に囲まれないので数字の一部とは見なさない。しかし、もし、隣接する left-to-right の並びがあれば、欧州の数字はその方向を取る。

```
Strage : he said "IT IS A bmw 500, OK."
Display: he said ".KO ,bmw 500 A SI TI"
```

暗黙のレベルの解決

最終段階では、解決した文字のタイプに基づいてテキストの埋め込みレベルが増加するかもしれない。right-to-left のテキストは常に奇数レベルになり、left-to-right と数字のテキストは常に偶数のレベルになる。付け加えると、数字のテキストは、かならず、パラグラフレベルよりも高いレベルになる。

- I1. 偶数の埋め込み方向の文字すべてについて、R のものは一段階あげ、AN または EN のものは二段階上げる。
- I2. 奇数の埋め込み方向の文字すべてについて、L、EN または AN のものは一段階上げる。

要約すると次の表のとおり。

タイプ	埋め込みレベル	
	偶数	奇数
L	EL	EL+1
R	EL+1	EL
AN	EL+2	EL+1
EN	EL+2	EL+1

解決したレベルの並び替え

段落の改行処理を施す。

段落を行単位に分割する改行処理は双方向アルゴリズムのカバー範囲外である。もし、文字の形状処理が関係する場合は、更に複雑になる。論理的には次のステップがある。

- テキストのレベルを双方向アルゴリズムに従って決定する。
- 文字をコンテキストに従って、字形にする。埋め込みレベルを考慮してミラーリングすること。
- 論理的な順序で、これらの字形の幅を足していき、改行位置を決定する。
- 各行に対して、次の L1 から L4 のアルゴリズムを適用する。
- 行の文字に対応する字形をその順番に表示する。

次は行単位で正しい表示順にする。

- L1. 各行において次の文字の埋め込みレベルを段落の埋め込みレベルにリセットする。なお、文字のタイプは、元の文字のタイプである。
  1. セグメント・セパレータ
  2. パラグラフ・セパレータ
  3. セグメント・セパレータまたはパラグラフ・セパレータに先行する空白文字の並び
  4. 行の終わりの空白文字の並び

  これによって、末尾の空白は、行の見かけ上の最後に出現することになる。
- L2. テキストの中の最もレベルの高い文字から最もレベルの低い奇数の文字まで、文字の連続する並びをそのレベルまたはより高いレベルのものをまとめて逆転する。これは、だんだんにより長い部分文字列を逆転することになる。次の例では、最初と 3 番目の例の段落の埋め込みレベルを 0、2 番目と 4 番目のそれを 1 に仮定する。

```
例 1 （埋め込みレベル=0）
Memory:           car means CAR.
Ressolved levels: 00000000001110
Reverse level 1:  car means RAC.
```

```
例１（埋め込みレベル=1）
Memory:              car MEANS CAR.
Ressolved levels:    22211111111111
Reverse level 2:     rac MEANS CAR.
Reverse levels 1-2:  .RAC SNAEM car
```

```
例１（埋め込みレベル=0）
Memory:              he said "car MEANS CAR."
Ressolved levels:    000000000222111111111100
Reverse level 2:     he said "rac MEANS CAR."
Reverse levels 1-2:  he said ".RAC SNAEM car"
```

```
例１（埋め込みレベル=1）
Memory:              DID YOU SAY 'he said "car MEANS CAR."'?
Ressolved levels:    11111111111122222222244433333333333211
Reverse level 4:     DID YOU SAY 'he said "rac MEANS CAR."'?
Reverse levels 3-4:  DID YOU SAY 'he said ".RAC SNAEM car"'?
Reverse levels 2-4:  DID YOU SAY '"rac MEANS CAR" dias eh"?
Reverse level 1-4:   ?'he said ".RAC SNAEM car"' YAS UOY DID
```

- L3. right-to-left の基底文字に適用した結合記号は、この時点で基底文字に先行している筈である。もし、最終の表示において結合文字が基底文字の後ろに来ることを期待するならば、記号と基底文字の順序を入れ替える必要がある。(詳細は、Unicode 仕様書 5.14 非間隔記号のレンダリングを参照 ※原文は 5.14 になっているが、誤り?)。
- L4. 4.7 節の Mirrored で指定される鏡像の特性をもつ文字は、もし、解決された方向が R であれば、鏡像の字形で描かねばならない。例えば、U+0028 left parenthesis は、Unicode では開き括弧と解釈される。解決したレベルが偶数であれば"("で表示されるが、奇数ならば")"で表示される。

形状（shaping）

双方向アルゴリズムを使った後で shaping を適用する。shaping は同じ方向の連続の範囲の文字に限定される。

準拠

双方向アルゴリズムは、right-to-left 文字の暗黙の意味の一部を指定している。方向性の解釈を置き換える上位レベルの規約がないとき、これらの文字を解釈するシステムは、レンダリングの際には、暗黙の双方向アルゴリズムと同じ結果にならなければならない。

参考資料

The Unicode Consortium: *The Unicode Standard Version 4.0* (Addison Wesley, 2003)

Mark Davis: *The Bidirectional Algorithm* (http://www.unicode.org/reports/tr9/tr9-11.html)
Date: 2003-04-17

T. F. Mitchell: *Writing Arabic (A Practical Introdcution to the Ruq'ah Script)* (Oxford University Press: 1953)

Huda Smitshujizen AbiFares: *Arabic Typography a comprehensive sourcebook* (London: Saqi Books, 2001)

Peter T. Daniels, William Bright: *The world's writing* (Oxford University Press, 1996)

Thomas Milo: *Authentic Arabic: a case study* (20th International Unicode Conference, Washington DC)

Kamal Mansour (Agfa | Monotype): *Guidelines to Use of Arabic Characters* (24th Internationalization & Unicode Conference, Atlanta, GA September 2003)

Mohammed Yousif <mhdyousif at gmx dot net>: *Harakat is broken in Qt* http://lists.arabeyes.org/archives/developer/2003/August/msg00006.html
Mon, 4 Aug 2003 10:08:06 +0300

本田孝一、師岡カリーマ・エルサムニー: *アラビア文字を読んでみよう書いてみよう* 白水社
2000 年 10 月