

# AH Formatter V6.2

Mar 7, 2014 石野恵一郎

## 1. 数式組版

### 2. 多言語組版

## 1. 数式組版

AH Formatter は、MathML を使って数式を組版することができます。MathML 描画エンジンは、AH Formatter V6.2 で完全に書き換えられました。このとき、MathML の対応バージョンも V6.1 までの 2.0 から最新の 3.0 へ更新されています。MathML 3.0 では、追加や廃止された属性などがありますが、初等数学用の要素群の追加が機能的には大きな違いとなります。

$$\begin{array}{r} 424 \\ +33 \\ \hline \end{array}$$

### 1.1 MathML の概要

MathML は XML の応用です。W3C で規格化されており、以下のように勧告されています（MathML 3.0 (2nd Edition) は勧告ではありません）。

1. MathML 1.0 – 7 April 1998
2. MathML 2.0 – 21 February 2001
3. MathML 2.0 (Second Edition) – 21 October 2003
4. MathML 3.0 – 21 October 2010
5. MathML 3.0 (2nd Edition) – 11 February 2014 – Proposed Edited Recommendation

AH Formatter は、MathML 3.0 に対応しています。MathML 3.0 で廃止された MathML 2.0 の要素属性にも対応しています。ごく新しい MathML 3.0 (2nd Edition) には対応していません。

MathML の主要な要素は、プレゼンテーションマークアップに属する要素群とコンテンツマークアップに属する要素群に大別されます。プレゼンテーションマークアップは、数式を構成するために必要なすべての要素を含んでおり、プレゼンテーションマークアップだけで MathML で表現できるすべての数式を表現できます。コンテンツマークアップは、プレゼンテーションマークアップの構造を知らなくても、数式を数学的構造を維持して簡単にマークアップすることを目的に設計されています。表示するときはプレゼンテーションマークアップに翻訳されて表示されますが、MathML の仕様では、どのように翻訳するのかを厳密に規定しているわけではなく、多くは表示系の実装に委ねられていました。AH Formatter ではコンテンツマークアップのための外部の情報を利用することはできません）

次の式はプレゼンテーションマークアップの簡単な例です。

$\sin x$

```
<mi> sin </mi>
<mo> &ApplyFunction; </mo>
<mi> x </mi>
```

同じ式をコンテントマークアップで表すと次のようにになります。

$\sin x$

```
<apply>
  <sin/>
  <ci> x </ci>
</apply>
```

また、コンテントマークアップでは、次のように引数によって括弧が自動的に補われたり、さまざまな調整が行なわれます。その反面、自由度はありません。

$\sin(x + y)$

```
<apply>
  <sin/>
  <apply>
    <plus/>
    <ci> x </ci>
    <ci> y </ci>
  </apply>
</apply>
```

## 1.2 MathML で利用するフォント

数式に利用するフォントは、専用にデザインされたものが望ましいでしょう。STIX Fonts (The Scientific and Technical Information eXchange) というフリーのフォントがあります。このフォントは、アメリカ物理学協会 (AIP)、アメリカ化学会 (ACS)、アメリカ数学会 (AMS)、アメリカ電気電子学会 (IEEE)、アメリカ物理学会 (APS)、エルゼビア社 (Elsevier) が参加する STI Pub companies が MathML 用に作成したものです。AH Formatter はこれをデフォルトとして想定しています。このフォントの斜体の字形は次のようなものです。

$x y z$

Windows には、Cambria Math という数式用フォントが同梱されています。STIX フォントがインストールされていないとき、AH Formatter はこのフォントをデフォルトとして使用します。斜体は次のような字形をしています。

$x y z$

数式でよく使う丸い感じの字形を使いたいことがあるでしょう。TeX で使われている BaKoMa というフリーのフォントは、次のような斜体の字形をしています。

$x y z$

AH Formatter では、MathML のマークアップに変更を加えることなく、一部の文字に異なるフォントを容易に割り当てすることができます。例えば、斜体の英小文字に BaKoMa を割り当てて使うには、設定ファイルで次のように指定するだけです。

```
<variant-font mathvariant="italic">
  <font-entry unicode-range="U+0061-007A" fontfamily="cmmi10" mathvariant="normal"/>
</variant-font>
<variant-font mathvariant="bold-italic">
  <font-entry unicode-range="U+0061-007A" fontfamily="cmmbi10" mathvariant="normal"/>
</variant-font>
```

これによって、次のように数式を表現できます。この例では、 $\sin$  が STIX で、 $x$  が BaKoMa になっています。

$$\sin x$$

また、Design Science 社の MathType Fonts (Euclid Fonts) も丸い感じの字形を持つフォントです。このフォントはグリフが下寄りにデザインされているため、マイナス記号などを式の中心にそろえるためには、設定ファイルでの調整が必要となります。

### 1.3 プrezentationマークアップの実際

ここでは、代表的なプレゼンテーションマークアップをごく簡単に紹介します。ただし、正確な仕様を解説するものではありませんし、すべての要素を紹介するものではありません。

MathML のルート要素は `<math>` です。名前空間は <http://www.w3.org/1998/Math/MathML> です。したがって、MathML の XML は次のように書きます。

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  ...
</math>
```

数式は、文章中にセンテンスの一部として記述する場合と、パラグラフを分けて独立なブロックとして記述する場合があります。前者をインラインモード、後者をブロックモードと呼びます。ここでは、インラインモードは割愛して、ブロックモードを仮定して説明します。ブロックモードの MathML は次のように書きます。

```
<math xmlns="http://www.w3.org/1998/Math/MathML" display="block">
  ...
</math>
```

MathML の仕様書では、プレゼンテーションマークアップの要素は次のように分類されています。

- トークン要素
- 全体的なレイアウト
- スクリプトとリミット
- 表
- 初等数学

### 1.3.1 トークン要素

トークン要素は、数式中に現れる文字を表現します。識別子は $<\text{mi}>$ 、演算子は $<\text{mo}>$ 、数値は $<\text{mn}>$ という具合です。空白以外の文字は、トークン要素中にのみ含めることができます。空白以外の文字がトークン要素の外側に現れることはできません。トークン要素の外側の空白はすべて無視され、トークン要素中の先行後行する空白も無視されます。例えば、

```
<math> + </math>
```

は、

```
<math>+</math>
```

と同じです。

$<\text{mi}>$  や $<\text{mo}>$ などの要素は、意味付けはされていますが、内容にどのようなテキストでも書くことができます。しかし、数値として記述した $<\text{mn}>\pi</\text{mn}>$ と識別子として記述した $<\text{mi}>\pi</\text{mi}>$ とでは、表示のされ方が異なることがあります。

トークン要素では、太字や斜体を含むさまざまな文字の変形を`mathvariant`属性で行なうことができます。代表的な例を次に示します。



MathML では、文脈によって文字を変換して表示します。

- U+002D ⇒ U+2212 または U+2010
- U+0027 ⇒ U+2032

数式として書かれた U+002D (HYPHEN MINUS) は、正しく U+2212 (MINUS SIGN) として扱われます。

#### 識別子 $<\text{mi}>$

x や y の識別子は $<\text{mi}>$ で表現します。通常`mathvariant`の初期値は`normal`ですが、 $<\text{mi}>$ では、内容が1文字のときに`italic`が初期値となります。MathML の仕様書にはこれ以上のことは書かれていませんが、内容が数字などの場合は`italic`にしないのが通例のようです。AH Formatter では、内容が英字、ギリシャ文字、キリル文字で1文字のときに`italic`で表示するようにしています。(この文字種に関する条件は設定ファイルで調整できます)

#### 数値 $<\text{mn}>$

数値は $<\text{mn}>$ で表現します。

#### 演算子 $<\text{mo}>$

演算子は $<\text{mo}>$ で表現します。演算子には、多くの属性が定義されており、各演算子がどういう属性値を持っているかは演算子辞書に記述されています。演算子辞書の内容は MathML の仕様書に規定外として載っています。AH Formatter では、この内容をそのまま初期値として採用しています。外部から演算子辞書を指定して設定内容を変更することも可能です。

演算子は、 $-2$ のような前置（prefix）、 $3!$ のような後置（postfix）、 $2+3$ のような中置（infix）に分類されています。すべての演算子はこれらのどれかになります。マイナスなどは、複数種類の演算子が存在するということになります。この形式を明示していない場合、文脈からどれかが判断されます。form 属性で明示できます。

演算子の前後には、適切なアキが自動的に取られます。次の例は単純な式ですが、 $+$ の前後にアキが取られます。中置演算子と前置演算子でアキの付き方が異なります。このアキは lspace、rspace 属性で調整できます。

$$2 + 3$$

```
<mn> 2 </mn>
<mo> + </mo>
<mn> 3 </mn>
```

$$+ 3$$

```
<mo> + </mo>
<mn> 3 </mn>
```

次の例を見てみます。

$$a +_4 b$$

```
<mi> a </mi>
<msub>
  <mo> + </mo>
  <mn> 4 </mn>
</msub>
<mi> b </mi>
```

この例では、演算子  $+$  に適用されるアキが  $+$  の添え字の  $4$  の右側に付いています。これは、 $+_4$  が  $+$  の性質を引き継いだ演算子として振舞っています。このような演算子を装飾演算子（embellished operator）と呼びます。装飾演算子は、核となる 1 つの演算子がある条件を満たす構造の中にあるとき、その構造全体が演算子として振舞うというものです。（条件の詳細は MathML の仕様書を見てください）

括弧が分数を囲んでいるようなとき、演算子としての括弧は自動的に高さが調整されて高くなります。左の例では中心から上下対称になるように引き伸ばされていますが、中の例では対称性が維持されていないことに注意してください。右の例では伸張されていません。stretchy 属性と symmetric 属性の指定によって、どのように伸張するかを指定することができます。

$$\left(\frac{1}{x}\right) \left(\frac{1}{x}\right) \left(\frac{1}{x}\right)$$

```
<mo> ( </mo>
<mfrac>
  <mn> 1 </mn>
  <mi> x </mi>
</mfrac>
<mo> ) </mo>
```

インテグラルや  $\Sigma$  など、文字サイズを大きくする必要のある演算子が存在します。そのような演算子は、自動的に大きく描かれます。largeop 属性によって大きくするかどうかを指定できます。

$$\int_0^1 f(x)dx$$

```

<msubsup>
  <mo> &int; </mo>
  <mn> 0 </mn>
  <mn> 1 </mn>
</msubsup>
<mi>f</mi>
<mo> &ApplyFunction; </mo>
<mfenced>
  <mi> x </mi>
</mfenced>
<mi> d </mi>
<mi> x </mi>

```

### テキスト $<\text{mtext}>$

数学的意味のない文字列は  $<\text{mtext}>$  で表現します。

### スペース $<\text{mspace}>$

$<\text{mspace}>$  で内容のない矩形を作ることができます。それによって、任意のサイズのアキを作ることができます。また、大きさのない  $<\text{mspace}>$  は、行分割位置候補として挿入することができます。

## 1.3.2 全体的なレイアウト

全体的な数式の構造を表現するための要素群です。分数や根号などが含まれます。

### グループ化 $<\text{mrow}>$

$<\text{mrow}>$  は、複数の要素をひとまとめにします。数学的な意味はありません。要素によっては、子要素を1つだけ要求するものがあります。そのような要素では、内容に複数の要素が含まれたときに、それらを自動的に  $<\text{mrow}>$  で囲んで、内容の要素を1つだけにします。例えば  $<\text{math}>$  がそうです。 $<\text{mrow}>$  は非常に多くの場面で利用されます。例えば、添え字や分数の分子や分母に複雑な式を書くときは、 $<\text{mrow}>$  でグループ化することでしょう。

### 分数 $<\text{mfrac}>$

$<\text{mfrac}>$  は子要素を2つ持ちます。分子と分母です。分子や分母の文字サイズは、分数を入れ子にするとベースの文字サイズより1段階小さくなります。この文字サイズの自動的な調整は、スクリプトレベルという方法で処理されます。添え字なども同様です。スクリプトレベルについては後述します。

$$1 + \frac{1}{1 + \frac{1}{x}}$$

```

<mn> 1 </mn>
<mo> + </mo>
<mfrac>
  <mn> 1 </mn>
  <mrow>
    <mn> 1 </mn>
    <mo> + </mo>
    <mfrac>
      <mn> 1 </mn>
      <mi> x </mi>
    </mfrac>
  </mrow>
</mfrac>

```

## 根号 <msqrt> <mroot>

<msqrt> は平方根で、子要素は 1 つです。<mroot> は冪根で、子要素は 2 つです。AH Formatter では、根号はフォントのグリフを利用せず、線描しています。太さや角度などの形状は設定ファイルで指定します。

$$\sqrt[3]{x + y}$$

```
<mroot>
  <mrow>
    <mi> x </mi>
    <mo> + </mo>
    <mi> y </mi>
  </mrow>
  <mn> 3 </mn>
</mroot>
```

## フェンス <mfenced>

<mfenced> は、(a, b, c) のようなパラメタの並びを表現します。括弧や区切りのコンマは変更することができます。<mfenced> は、右に示したように、等価な他の要素に置き換えることができます。

$$[0, 1)$$

```
<mfenced open="[">
  <mn> 0 </mn>
  <mn> 1 </mn>
</mfenced>      <mrow>
                  <mo> [ </mo>
                  <mn> 0 </mn>
                  <mo> , </mo>
                  <mn> 1 </mn>
                  <mo> ) </mo>
                </mrow>
```

## 不可視 <mphantom>

<mphantom> は、内容の大きさを維持したまま内容を不可視にします。

$$\frac{x + y + z}{x + z}$$

```
<mfrac>
  <mrow>
    <mi> x </mi>
    <mo> + </mo>
    <mi> y </mi>
    <mo> + </mo>
    <mi> z </mi>
  </mrow>
  <mrow>
    <mi> x </mi>
    <mphantom>
      <mo form="infix"> + </mo>
      <mi> y </mi>
    </mphantom>
    <mo> + </mo>
    <mi> z </mi>
  </mrow>
</mfrac>
```

この例では、分子と分母を揃えるために、一部を不可視にしています。form="infix" の指定がないと、この文脈では + 演算子が prefix に解釈されてしまい、アキの量が分子と違ってしまいます。

### 1.3.3 スクリプトとリミット

上付き下付きの添え字と、 $\lim_{x \rightarrow 0}$  のような上下に修飾のある構造を表現するための要素群です。

#### スクリプト <msub> <msup> <msubsup> <mmultiscripts>

<msub> は下付き、<msup> は上付きで子要素は 2 つです。<msubsup> は上下に添え字を付け、子要素は 3 つです。

$$B_3$$

```
<msub>
  <mi> B </mi>
  <mn> 3 </mn>
</msub>
```

$$B^{2^n}$$

```
<msup>
  <mi> B </mi>
  <msup>
    <mn> 2 </mn>
    <mi> n </mi>
  </msup>
</msup>
```

$$B_3^{2^n}$$

```
<msubsup>
  <mi> B </mi>
  <mn> 3 </mn>
  <msup>
    <mn> 2 </mn>
    <mi> n </mi>
  </msup>
</msubsup>
```

添え字の文字の大きさは、親文字よりも 1 段階小さい文字となります。この段階をスクリプトレベルと呼び、文字の大きさを表します。最初の状態のスクリプトレベルは 0 です。 $+1$  すると 1 段階文字が小さくなります。 $-1$  すると 1 段階大きくなります。どのくらい小さく（大きく）するかは調整できますが、初期値は  $0.71$  ( $\approx 1/\sqrt{2}$ ) です。小さくするときフォントサイズを  $\times 0.71$  し、大きくするとき  $\div 0.71$  します。上の例で、<msup> の  $n$  は、2 段階小さくなつて親文字の半分のフォントサイズとなっています。

添え字が際限なく小さくなってしまうことを防ぐために、限界値が用意されています。その初期値は  $8pt$  です。スクリプトレベルをいくら大きくしても、それより小さくはありません。この値は絶対値です。親文字の大きさを変化させても、添え字が親文字と同じ倍率で変化するとは限らないことに注意してください。つまり、式全体としては相似ではないかも知れません。

<mmultiscripts> を使うと、親文字の左側にも添え字を付けることができます。

$$\frac{4}{3} F^2_1$$

```
<mmultiscripts>
  <mi> F </mi>
  <mn> 1 </mn>
  <mn> 2 </mn>
  <mprescripts/>
  <mn> 3 </mn>
  <mn> 4 </mn>
</mmultiscripts>
```

## リミット <munder> <mover> <munderover>

親文字の上下に修飾文字を付けます。<munder> は下に、<mover> は上に付けます。子要素は 2 つです。<munderover> は上下に付けます。子要素は 3 つです。

$$\lim_{x \rightarrow 0} \text{lim}$$

```
<munder>
  <mi> lim </mi>
  <mrow>
    <mi> x </mi>
    <mo> &RightArrow; </mo>
    <mn> 0 </mn>
  </mrow>
</munder>
```

$$\overbrace{x + y}^{\text{overbrace}}$$

```
<mover>
  <mrow>
    <mi> x </mi>
    <mo> + </mo>
    <mi> y </mi>
  </mrow>
  <mo> &OverBrace; </mo>
</mover>
```

$$\sum_{i=0}^n$$

```
<munderover>
  <mo> &sum; </mo>
  <mrow>
    <mi> i </mi>
    <mo> = </mo>
    <mn> 0 </mn>
  </mrow>
  <mi> n </mi>
</munderover>
```

上下の修飾文字の大きさは、<mover accent="false"> または <munder accentunder="false"> のときスクリプトレベル +1 されます。true のときはスクリプトレベルの変化はありません。初期値は false なので、何もしなければ 1 段階小さい文字になります。

<mover> は、文字にアクセントを付けるのにも利用されます。そのため、修飾文字がアクセントかどうかを区別する accent 属性があります。また、修飾文字が演算子のときは親文字の幅いっぱいに演算子を左右に引き伸ばすことができます。それを指示するのは <mo> の stretchy 属性で行ないです。上の <mover> の例では、stretchy は指定されていませんが、演算子辞書で stretchy が指定されているために、&OverBrace; が引き伸ばされています。

accent と stretchy の組み合わせでアクセント文字がどう変化するのかを、親文字が斜体の場合とそうでない場合の例で示します。上は U+00AF を、下は U+005E を使った例です。

$\bar{p}$	$\bar{p}$	$\bar{p}$	$\bar{p}$	$\bar{p}$	$\bar{p}$	$\bar{p}$	$\bar{p}$
$\hat{p}$	$\hat{p}$	$\hat{p}$	$\hat{p}$	$\hat{p}$	$\hat{p}$	$\hat{p}$	$\hat{p}$
accent stretchy	true true	true false	false true	false false	true true	true false	false true

AH Formatter では、次のような調整をしています。

- accent="true"かつ stretchy="true" のとき、斜体の上のアクセントは、アクセントがちょうど斜体の文字の上になるように位置と大きさを調整しています。
- &Hat; (U+005E) をアクセント文字に使うことがあります。U+005E は、たいていのフォントではアクセントには不向きな大きさをしています。そのため、stretchy="true" のときは、高さを低く調整しています。

#### 1.3.4 表

行列は、表で表現します。

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

```

<mo> ( </mo>
<mtable>
<mtr>
<mtd> <mn> 1 </mn> </mtd>
<mtd> <mn> 0 </mn> </mtd>
<mtd> <mn> 0 </mn> </mtd>
</mtr>
<mtr>
<mtd> <mn> 0 </mn> </mtd>
<mtd> <mn> 2 </mn> </mtd>
<mtd> <mn> 0 </mn> </mtd>
</mtr>
<mtr>
<mtd> <mn> 0 </mn> </mtd>
<mtd> <mn> 0 </mn> </mtd>
<mtd> <mn> 3 </mn> </mtd>
</mtr>
</mtable>
<mo> ) </mo>
```

また、連立方程式などの複数の式の上下の揃えも表で行ないます。

$$8.44x + 55y = 0$$

$$3.1x - 0.7y = -1.1$$

```
<math>
  <mtable groupalign="{'decimalpoint' left 'left' 'decimalpoint' left 'left' 'decimalpoint'}">
    <mtr>
      <mtd>
        <maligngroup/>
        <mn> 8.44 </mn>
        <mo> &InvisibleTimes; </mo>
        <maligngroup/>
        <mi> x </mi>
        <maligngroup/>
        <mo> + </mo>
        <maligngroup/>
        <mn> 55 </mn>
        <mo> &InvisibleTimes; </mo>
        <maligngroup/>
        <mi> y </mi>
        <maligngroup/>
        <mo> = </mo>
        <maligngroup/>
        <mn> 0 </mn>
      </mtd>
    </mtr>
    <mtr>
      <mtd>
        <maligngroup/>
        <mn> 3.1 </mn>
        <mo> &InvisibleTimes; </mo>
        <maligngroup/>
        <mi> x </mi>
        <maligngroup/>
        <mo> - </mo>
        <maligngroup/>
        <mn> 0.7 </mn>
        <mo> &InvisibleTimes; </mo>
        <maligngroup/>
        <mi> y </mi>
        <maligngroup/>
        <mo> = </mo>
        <maligngroup/>
        <mo> - </mo>
        <mn> 1.1 </mn>
      </mtd>
    </mtr>
  </mtable>
```

### 1.3.5 初等数学

MathML 3.0 で初等数学で用いられる表現ができるように要素が追加されました。ここでは、ごく基本的な例だけを示します。

$$\begin{array}{r} 424 \\ + 33 \\ \hline \end{array}$$

```
<mstack>
  <mn> 424 </mn>
  <msrow>
    <mo> + </mo>
    <mn> 33 </mn>
  </msrow>
  <msline/>
</mstack>
```

$$\begin{array}{r}
 435.3 \\
 3) \overline{1306} \\
 \underline{12} \\
 \overline{10} \\
 \underline{9} \\
 \overline{16} \\
 \underline{15} \\
 \overline{1.0} \\
 \underline{9} \\
 \overline{1}
\end{array}$$

```
<mlongdiv>
  <mn> 3 </mn>
  <mn> 435.3 </mn>
  <mn> 1306 </mn>
  <msgroup position="2" shift="-1">
    <msgroup>
      <mn> 12 </mn>
      <msline length="2"/>
    </msgroup>
    <msgroup>
      <mn> 10 </mn>
      <mn> 9 </mn>
      <msline length="2"/>
    </msgroup>
    <msgroup>
      <mn> 16 </mn>
      <mn> 15 </mn>
      <msline length="2"/>
      <mn> 1.0 </mn>
    </msgroup>
    <msgroup position="-1">
      <mn> 9 </mn>
      <msline length="3"/>
      <mn> 1 </mn>
    </msgroup>
  </msgroup>
</mlongdiv>
```

## 1.4 応用例

実際の応用例をいくつか示します。これらは、<mspace>などによる微調整なしでの AH Formatter の組版結果です。

### Quadratic Formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

### Standard Deviation

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

### Cauchy's Integral Formula

$$f(a) = \frac{1}{2\pi i} \oint_{\gamma} \frac{f(z)}{z-a} dz$$

### Einstein's Field Equations

$$R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R + g_{\mu\nu}\Lambda = \frac{8\pi G}{c^4}T_{\mu\nu}$$

### Legendre's Differential Equation

$$\frac{d}{dx} \left[ (1-x^2) \frac{d}{dx} P_n(x) \right] + n(n+1)P_n(x) = 0$$

### Sophomore's Dream

$$\int_0^1 x^x dx = \sum_{n=1}^{\infty} (-1)^{n+1} n^{-n} = - \sum_{n=1}^{\infty} (-n)^{-n}$$

### Maxwell's Equations

$$\left\{ \begin{array}{l} \nabla \times \overleftarrow{B} - \frac{1}{c} \frac{\partial \overleftarrow{E}}{\partial t} = \frac{4\pi}{c} \overleftarrow{j} \\ \nabla \cdot \overleftarrow{E} = 4\pi\rho \\ \nabla \times \overleftarrow{E} + \frac{1}{c} \frac{\partial \overleftarrow{B}}{\partial t} = \overleftarrow{0} \\ \nabla \cdot \overleftarrow{B} = 0 \end{array} \right.$$

### Ramanujan Identity

$$\frac{1}{(\sqrt{\phi\sqrt{5}} - \phi)e^{\frac{25}{\pi}}} = 1 + \frac{e^{-2\pi}}{1 + \frac{e^{-4\pi}}{1 + \frac{e^{-6\pi}}{1 + \frac{e^{-8\pi}}{1 + \dots}}}}$$

## 1.5 まとめ

ここまで、ごく一部の機能の紹介をしましたが、これだけでも多くの数式を表現できます。オンラインモードや、演算子の他の属性、行分割制御など、紹介し切れていない重要な機能は多くあります。是非勉強してみてください。

## 2. 多言語組版

AH Formatter は、日本語や英語はもちろん、他の多くの言語を組版することができます。どちらかというと、「言語」を組版するよりも、「文字」を組版します。なんとか文字という文字の分類は「スクリプト」と呼ばれ、ISO 15924 で規定されています。「言語」は ISO 639 で規定されています。スクリプトは物理的な性質で、言語は論理的な性質と言えます。例えば、英語もフランス語もドイツ語もラテン文字を使って表現されます。日本語をラテン文字を使って表現するとき、その文字をローマ字と呼びます。

言語の違いはハイフネーションなどの組版規則の違いとなって現れます。スクリプトの違いは文字進行方向などに影響します。多くは、言語が決まればスクリプトも予想できますが、一部の言語はそれだけではスクリプトが決定できないことがあります。中国語（中文）がそうです。中国と台湾では、使われる漢字のスクリプトが異なります。

AH Formatter V6.2 では、以下のスクリプトを扱うことができます。

- ラテン文字      • タイ文字      • グルムキ文字      • カンナダ文字
- ギリシャ文字    • クメール文字    • グジャラート文字    • マラヤラム文字
- キリル文字     • ラオ文字     • オリヤ文字     • ハングル
- アラビア文字   • デヴァナガリ文字   • タミル文字   • ひらがな・カタカナ
- ヘブライ文字    • ベンガリ文字    • テルグ文字    • 漢字

スクリプトには大雑把に言って単純なものと複雑なものがあります。単純なスクリプトは、Unicode をそのまま並べればほぼ組むことができます。複雑なスクリプトは、文字の並びによってさまざまな字形の変化を伴い、アクセントなどの修飾位置も複雑です。文字の順序を入れ替えるものもあります。その方法はスクリプトごとに独特で、フォント中に記述されている情報と、それをどう処理するかのロジックによって実現されています。上の一覧で、

- 単純なスクリプト
- 複雑なスクリプト

となっています。Unicode には、上記以外にも多くの見たこともないようなスクリプトが登録されています。それが複雑なスクリプトであれば、AH Formatter では、それを正しく扱うことはできません。例えば、シンハラ文字やチベット文字などです。単純なスクリプトであれば、AH Formatter でもそれなりに表示できるかも知れません。例えば、クルジア文字やアルメニア文字です。

複雑なスクリプトの文字を表示するためには、OpenType や TrueType のフォントが必要です。複雑なスクリプト処理に必要な OpenType フォントの仕様は、ある程度公開されています。Windows Vista 以降に同梱のインド系の OpenType には、新しいバージョンである V2 仕様のフォントが含まれています。V2 の仕様はフォント処理のために必要な情報があまり公開されていません。多くのフォントには V1 の機能も含まれているので、AH Formatter はそちらの機能を使ってそのフォントを扱うことができますが、一部のフォントには V2 の機能しか含まれていないものがあります。AH Formatter ではそのようなフォントを扱うことができません。しかし、幸いなことに優れたフリーのフォントも多数存在するので、それを使って組版することができます。

以降の例示の文章は、世界人権宣言 <http://www.unicode.org/udhr/> の一節です。

## 2.1 ラテン文字

ラテン文字は英語をはじめ多数の言語に使われています。世界中で最も使われている文字です。 Wikipediaによると、右の図の地域で使われているそうです。

AH Formatterでは、ラテン文字で表現される言語のうち、40弱の言語のハイフネーションに対応しています。それ以外の言語でも組版することはできますが、ハイフネーションはできません。例えば、ベトナム語のハイフネーションはできません。



英語

All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.

フランス語

Tous les êtres humains naissent libres et égaux en dignité et en droits. Ils sont doués de raison et de conscience et doivent agir les uns envers les autres dans un esprit de fraternité.

ベトナム語

Tất cả mọi người sinh ra đều được tự do và bình đẳng về nhân phẩm và quyền. Mọi con người đều được tạo hoá ban cho lý trí và lương tâm và cần phải đối xử với nhau trong tình bằng hữu.

## 2.2 ギリシャ文字

ギリシャ文字はギリシャ語で使われます。AH Formatterは、ギリシャ語のハイフネーションも可能です。

ギリシャ語

'Ολοι οι ἀνθρωποι γεννιούνται ελεύθεροι καὶ ίσοι στην αξιοπρέπεια καὶ τα δικαιώματα. Είναι προικισμένοι με λογική καὶ συνείδηση, καὶ οφείλουν να συμπεριφέρονται μεταξύ τους με πνεύμα αδελφοσύνης.

## 2.3 キリル文字

キリル文字はロシア語などロシア周辺で用いられている文字です。 Wikipediaによると、右の図の地域で使われているそうです。 言語によってはハイフネーションも可能です。



ロシア語

Все люди рождаются свободными и равными в своем достоинстве и правах. Они наделены разумом и совестью и должны поступать в отношении друг друга в духе братства.

## ブルガリア語

Всички хора се раждат свободни и равни по достойнство и права. Те са надарени с разум и съвест и следва да се отнасят помежду си в дух на братство.

## 2.4 アラビア文字

アラビア文字は右から左に書く文字で、中東から北アフリカのイスラム圏の言語で使われます。ラテン文字など左から右に書く文字との混植では、複雑な BIDI 処理が行なわれます。

### アラビア語

اعتمد بموجب قرار الجمعية العامة 217 ألف (3-د) المؤرخ في 10 كانون الأول / ديسمبر 1948.  
يولد جميع الناس أحرازاً متساوين في الكرامة والحقوق. وقد وهبوا عقلاً وضميراً ولهم أن يعامل بعضهم بعضًا بروح الإخاء.

### ウルドゥ語

تمام انسان آزادی اور حقوق و عزت کے اعتبار سے برابر پیدا ہوئے ہیں۔ انہیں ضمیر اور عقل و دیعت ہوئی ہے۔ اس لئے انہیں ایک دوسرے کے ساتھ بھائی چارے کا سلوک کرنا چاہیے۔

## 2.5 ヘブライ文字

ヘブライ文字はアラビア文字と同様に右から左に書く文字で、主にヘブライ語で使われます。

### ヘブライ語

כל בני אדם נולדו בני חורין ושוויים בראכם ובמצפונם. כולם חוננו בתבונה ובזכויותיהם. לפיכך חובה עליהם לנ هو איש ברעהו ברוח של אחווה.

## 2.6 タイ文字

タイ文字はタイ語で使われる文字で、親文字の前後上下に声調記号などの修飾がある複雑なスクリプトです。タイ語ではいくつもの単語を分かち書きしないで繋げて書くことが多く、行分割位置を探すには辞書が必要です。

### タイ語

ມນຸຍໍ່ທັງໝາຍເກີດມາມີອີສະຮະແລະເສມອກາຄກັນໃນເກີຍຣຕິສັກດີ [ເກີຍຣຕິສັກດີ] ແລະ ຕີທີ ຕ່າງມີເຫຼຸຜລແລະມືອນຈຽບ  
ແລະຄວຽມປຸງຕິດຕໍ່ກັນດ້ວຍເຈຕນາຮນ໌ແໜ່ງກຣາດຮກກາວ

## 2.7 クメール文字

クメール文字はカンボジアのクメール語で使われる文字です。Windows Vista 以降に同梱の DaunPenh というフォントは、なぜか字形が非常に小さくデザインされています。そのため、欧文などと混植をした場合、バランスが非常に悪いです。ここでは KhmerOS を使っています。

クメール語

មនុស្សទាំងអស់ កៅតមកមានសេវភាព និងសមភាព ក្នុងផ្លូវកសេចក្តីផ្លូវនិងសិទ្ធិ។ មនុស្ស មាន វិចារណាថ្មាននិងសតិសម្បជញ្ញា៖ដាប់ពីកំណើត ហើយគួរពីតុលចំពោះត្រាវិញ្ញាទេមក ក្នុង ស្ថាផីភាពរភាពដាបងប្រឈម។

## 2.8 ラオ文字

ラオ文字はラオスのラオ語で使われる文字です。

ラオ語

ມະນຸດເກີດມາມີເຮັດແສວລົງພາບ ແລະ ວະເໜີໜ້າກັນໃນທາງກາງດຕີສັກ ແລະ ທາງສົດດ້ວຍມະນຸດມີສະຕິສໍາບັດຊັ້ນຢະ (ຮູ້ດີຮູ້ຊື່ວ) ແລະ ມີມະໂນທ່າລົງຕ້ອງປະເີດຕົນຕໍ່ກັນໃນທາງຜົ່ນ້ອງ.

## 2.9 デヴァナガリ文字

デヴァナガリ文字はインドのヒンディ語やサンスクリット語などで使われる文字です。

ヒンディ語

सभी मनुष्यों को गौरव और अधिकारों के मामले में जन्मजात स्वतन्त्रता और समानता प्राप्त है। उन्हें बुद्धि और अन्तरात्मा की देन प्राप्त है और परस्पर उन्हें भाईचारे के भाव से बर्ताव करना चाहिए।

インド政府が定めている公用語はヒンディー語で、英語が準公用語となっていますが、インドの各地方には指定言語が定められており、それらはその地方での公用語となっています。

インドで発行されている紙幣はヒンディー語と英語で書かれていますが、裏面には、主要な公用語（指定言語）である 15 の言語で金額が表記されています。

ଦଶ ଟେକା	アッサム語	ベンガリ文字
ଦଶ ଟାକା	ベンガル語	ベンガリ文字
ଦଶ ଟାକା	グジャラート語	グジャラート文字
କମ୍ପୁଲାଟାଯାଗିର୍ଜି	カンナダ語	カンナダ文字
ହାତୀ ପର ୧୨	カシミル語	アラビア文字
ଦଶ ରୂପଯ୍ୟ	コンカニ語	デヴァナガリ文字
ପଣ୍ଡିତ ଗୁପ୍ତ	マラヤラム語	マラヤラム文字
ଦହା ରୂପଯ୍ୟ	マラ泰イ語	デヴァナガリ文字
ଦଶ ରୂପଯ୍ୟ	ネパール語	デヴァナガリ文字
ଦଳୀ ଟଙ୍କା	オリヤ語	オリヤ文字
ଦମ ରୂପଦେ	パンジャブ語	グルムキ文字
ଦଶରୂପକାଣି	サンスクリット語	デヴァナガリ文字
ପତନ୍ତୁ ଶୁପାଯ୍	タミル語	タミル文字
ପଦମ୍ବାହାଯିବ	テルグ語	テルグ文字
ରୂପ ରୂପି	ウルドゥー語	アラビア文字

## 2.10 ベンガリ文字

ベンガリ文字はバングラデシュでのベンガル語やインド北東部アッサム州でのアッサム語に使われる文字です。

ベンガル語

সমস্ত মানুষ শাক্ষীনভাবে সমান মর্যাদা এবং অধিকার নিয়ে জন্মগ্রহণ করে। তাঁদের বিবেক এবং বুদ্ধি আছে ; সুতরাঃ সকলেরেই একে অপরের প্রতি ব্রাহ্মসমূভ মনোভাব নিয়ে আচরণ করা উচিত।

## 2.11 グルムキ文字

グルムキ文字はインド北部パンジャブ州でのパンジャブ語に使われる文字です。

パンジャブ語

ਸਾਰਾ ਮਨੁੱਖੀ ਪਰਿਵਾਰ ਆਪਣੀ ਮਹਿਮਾ, ਸ਼ਾਨ ਅਤੇ ਹੱਕਾਂ ਦੇ ਪੱਖੋਂ ਜਨਮ ਤੋਂ ਹੀ ਆਜ਼ਾਦ ਹੈ ਅਤੇ ਸੁਤੇ ਸਿੱਧ ਸਾਰੇ ਲੇਕ ਬਰਾਬਰ ਹਨ। ਉਨ੍ਹਾਂ ਸਭਨਾ ਨੂੰ ਤਰਕ ਅਤੇ ਜ਼ਸੀਰ ਦੀ ਸੌਗਾਤ ਮਿਲੀ ਹੋਈ ਹੈ ਅਤੇ ਉਨ੍ਹਾਂ ਨੂੰ ਭਰਾਤਰੀਭਾਵ ਦੀ ਭਾਵਨਾ ਰਖਦਿਆਂ ਆਪਸ ਵਿਚ ਵਿਚਰਣਾ ਚਾਹੀਦਾ ਹੈ।

## 2.12 グジャラート文字

グジャラート文字はインド西部グジャラート州でのグジャラート語に使われる文字です。

グジャラート語

પ્રતિજ્ઞા અને અધિકારોની દાખિએ સર્વ માનવો જન્મથી સ્વતંત્ર અને સમાન હોય છે. તેમનામાં વિચારશક્તિ અને અંતઃકરણ હોય છે અને તેમણે પરસ્પર બંધુત્વની ભાવનાથી વર્તવું જોઇએ.

## 2.13 オリヤ文字

オリヤ文字はインド東部オリッサ州でのオリヤ語に使われる文字です。Windows Vista 以降に同梱の Kalinga は、V2 のフォントで V1 の機能が含まれていません。ここでは Lohit Oriya を使ってています。(オリヤ語の世界人権宣言は見つかりませんでしたので、次は Wikipedia から引用)

オリヤ語

ଓଡ଼ିଆ ଏକ ଲକ୍ଷ୍ମୋ-ଜଗରୋପୀୟ ଭାଷାଗୋଷ୍ଠୀ ଅନ୍ତର୍ଗତ ଲକ୍ଷ୍ମୋ-ଆର୍ଯ୍ୟ ଭାଷା । ଓଡ଼ିଶା ସମେତ ଏହା ପଣ୍ଡିତବଳୀ, ଛତିଶଗଡ଼, ଝାଡ଼ଖଣ୍ଡ, ଆନ୍ଧ୍ରପ୍ରଦେଶ ଓ ଗୁଜରାଟ (ମୂଳତ୍ୱ ସୁରକ୍ଷା [୧]) ରେ କହାଯାଇଥାଏ ।

## 2.14 タミル文字

タミル文字はインド南部タミル・ナードゥ州でのタミル語に使われる文字です。

タミル語

மனிதப் பிறிவியினர் சகலரும் சுதந்திரமாகவே பிறக்கின்றனர் ; அவர்கள் மதிப்பிலும், உரிமைகளிலும் சமமானவர்கள், அவர்கள் நியாயத்தையும்

மனச்சாட்சியையும் இயற்பண்பாகப் பெற்றவர்கள். அவர்கள் ஒருவருடனாருவர் சகோதர உணர்வுப் பாங்கில் நடந்துகொள்ளல் வேண்டும்

## 2.15 テルグ文字

テルグ文字はインド南東部アーンドラ・プラデーシュ州でのテルグ語に使われる文字です。(テルグ語の世界人権宣言は見つかりませんでしたので、次は Wikipedia から引用)

テルグ語

ఆంధ్రప్రదేశ్ రాష్ట్ర అధికార భాష తెలుగు. భారత దేశంలో తెలుగు మాతృభాషగా మాట్లాడే 8.7 కోట్ల (2001) జనాభాతో ప్రాంతాలలో మొదటి స్థానంలో ఉంది. ప్రపంచంలోని ప్రజలు అత్యధికముగా మాట్లాడే భాషలలో పదమూడవ స్థానములోను, భారత దేశములో హిందీ, బెంగాలీ తర్వాత మూడవ స్థానములోను నిలుస్తుంది. పాతవైన ప్రపంచ భాష గణాంకాల (ఎఫ్స్) లాగ్ ప్రకారం ప్రపంచవ్యాప్తంగా 7.4 కోట్లు మందికి మాతృభాషగా ఉంది.

## 2.16 カンナダ文字

カンナダ文字はインド南西部カルナータカ州でのカンナダ語に使われる文字です。

カンナダ語

ಎಲ್ಲಾ ಮಾನವರೂ ಸ್ವತಂತ್ರರಾಗಿಯೇ ಜನಿಸಿದ್ದಾರೆ. ಹಾಗೂ ಘನತೆ ಮತ್ತು ಹಕ್ಕುಗಳಲ್ಲಿ ಸಮಾನರಾಗಿದ್ದಾರೆ. ವಿವೇಕ ಮತ್ತು ಅಂತಃಕರಣ ಗಳನ್ನು ಪದೆದವರಾದ್ದರಿಂದ ಅವರು ಪರಸ್ಪರ ಸಹೋದರ ಭಾವದಿಂದ ವರ್ತಿಸಬೇಕು.

## 2.17 マラヤラム文字

マラヤラム文字はインド南部ケーララ州でのマラヤラム語に使われる文字です。

マラヤラム語

മനുഷ്യരെല്ലാവരും തുല്യാവകാശങ്ങളോടും അന്തസ്ഥിതിയാണ്  
സ്വാത്രത്യത്വത്തോടുകൂടി ജനിച്ചിട്ടുള്ളവരാണ്. അന്തേക്കാനും ഭാത്യഭാവത്തോടെ  
പെരുമാറുവാനാണ് മനുഷ്യനു വിവേകബുദ്ധിയും മനസ്സാക്ഷിയും  
സിദ്ധമായിരിക്കുന്നത്.

## 2.18 ハングル

ハングルは朝鮮語・韓国語に使われる文字です。

朝鮮語・韓国語

모든 인간은 태어날 때부터 자유로우며 그 존엄과 권리에 있어 동등하다. 인간은 천부적으로 이성과 양심을 부여받았으며 서로 형제애의 정신으로 행동하여야 한다.

## 2.19 ひらがな・カタカナ・漢字

ひらがな・カタカナは日本語だけで使われる文字ですが、漢字は日本語だけではありません。漢字のスクリプトは3種類あります。日本で使われる漢字 Hani、中国で使われる簡体字 Hans、台湾・香港・マカオで使われる繁體字 Hant です。スクリプトは3種類ですが、言語は日本語と中国語（中文）しかありません。中国語は、言語指定 `xml:lang="zh"` だけからスクリプトを推測できないのです。言語指定に国コードを併記して、`xml:lang="zh-CN"` のようにしたり、スクリプトを `script="Hans"` と明示する必要があります。

日本語

すべての人間は、生まれながらにして自由であり、かつ、尊厳と権利について平等である。人間は、理性と良心とを受けられており、互いに同胞の精神をもって行動しなければならない。

简体字中国語

人人生而自由，在尊严和权利上一律平等。他们赋有理性和良心，并应以兄弟关系的精神相对待。

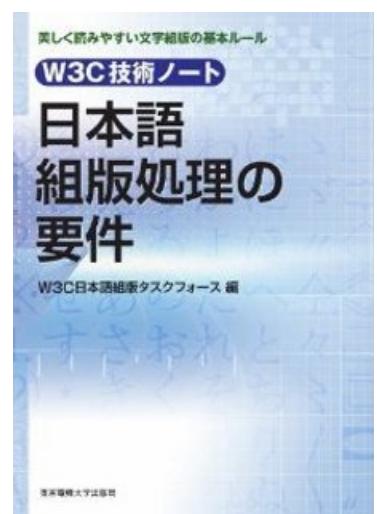
繁體字中国語

人人生而自由，在尊嚴和權利上一律平等。他們賦有理性和良心，並應以兄弟關係的精神相對待。

AH Formatter は、もちろん縦書きやルビにも対応しています。

ば り、 る。 あ す  
な ら い ひ ん べ べ  
ら い た が 人 て の  
な い い い 間 人 人  
い い に は は は  
い 。 に は 、 、 は  
い 。 同 互 か に  
い い 胞 が は う  
い の 理 人 う  
い の 性 尊 う  
い の 精 尊 う  
い の 神 と 生  
い し と 権 ま  
し し と 利 な  
し し と と が  
し し と 良 な  
し し と 心 が  
し し と を に  
し し と 行 つ  
し し と 動 い  
し し し な て  
し し け お あ で

AHFormatter は、日本語の組版について W3C 技術ノートの「日本語組版処理の要件」(<http://www.w3.org/TR/jlreq/ja/>) の多くを取り入れています。日本語版が東京電機大学出版局から出版されています。ISBN 978-4-501-55020-2



## 2.20 異体字

漢字の字形については、JIS での変遷など過去に複雑な経緯がありますが、割愛します。

現在の Unicode では、漢字の異体字を異体字セレクタ (IVS : Ideographic Variation Selector) という選択用の文字を使って表現することになっています。異体字選択機能を使うには、フォントが異体字を持っていなければなりません。異体字を持っていないフォントでは、異体字セレクタは無視されます。

葛&#xE0100;城市	葛城市
葛&#xE0101;飾区	葛飾区

U+E0100 は 0 番の字形で標準的なものを選択します。U+E0101 は 1 番の字形、U+E0102 は 2 番の字形、という具合です。何番にどういう字形が割り当てられているかは管理され決められているので、どのフォントを使っても、そのフォントがその番号の異体字を持っていれば同じ字形が表示されます。字形のデータは Ideographic Variation Database (<http://www.unicode.org/ivd/>) から入手できます。

U+9089 邊 邊 邊 邊 邊 邊 邊 邊 邊 邊 邊 邊 邊

U+908A 邊 邊 邊 邊 邊 邊 邊 邊 邊 邊

## 2.21 住基ネット統一文字コード

2002 年 8 月 5 日からスタートした住基ネット（住民基本台帳ネットワークシステム）は、「統一文字コード」という文字コードを利用します。その仕様は、「財団法人 地方自治情報センター（LASDEC）」から、極めて限定期に提供されているだけで、実質非公開と言えます。

この統一文字コードは、Unicode のように見えますが、Unicode ではありません。

Unicode 1.1 が 1993 年、2.0 が 1996 年、3.0 が 1999 年です。JIS X 0213 は 2000 年です。この JIS には、字形と共に UCS も併記されているのですが、そこには括弧付きで示されたものがあります。それは当時 Unicode として未割り当てであった文字ですが、JIS が割り当てを想定したと思われる UCS が括弧付きで書かれています。その後の Unicode の改訂で、JIS の思惑と異なる UCS への割り当てが行なわれました。それに伴い、2004 年の JIS X 0213 追補 1 では、括弧付き UCS であった文字のうち、363 字の UCS の変更が行なわれました。

1-86-72 7668 ebc6	渦	85 (水) 7 39-33	17474 5998
1-86-73 7669 ebe7 (FA46)	海	85 (水) 7 39-33	17503 5922
1-86-74 766a ebe8	涂	85 (水) 7 39-38	17518 5987
1-86-75 766b ebe9	涇	85 (水) 7 39-39	17526 5966

統一文字コードでは、JIS X 0213 の括弧付きの UCS がそのまま採用されています。そのため、統一文字コードは、JIS とも不整合になっているのです。JIS X 0213 ない統一文字コード専用の変体仮名や異体字等は、Unicode のハングルなどの領域に割り当てを行なったため、ハングルと統一文字コードの併用は不可能になりました。

もはや得体の知れないコード体系になっているわけです。何か専用のエンコーディング名が定義されていればよかつたんですが、それはありません。あくまで Unicode の顔をしているので、Unicode ベースの AH Formatter では、何もしないと意図しない文字が表示されるか、文字が欠けるかしてしまいます。ハングルの領域に割り当てられた文字は漢字やかなとは認識されないので、組版上不都合が起こるでしょう。次の例のようにちょっとだけずれているものも多数あるので、一見しただけではわからない場合もあるでしょう。

UCS	統一文字コード	Unicode
U+FA45	概	海
U+FA46	海	渚
U+FA47	渚	漢
U+FA48	漢	煮
U+FA49	煮	𩫑
U+FA4A	𩫑	琢
U+FA4B	琢	碑
U+FA4C	碑	社
U+FA4D	社	祉
U+FA4E	祉	祈
U+FA4F	祈	祐
U+FA50	祐	祖
U+FA51	祖	祝

AH Formatter は、統一文字コード用のオプションを販売しており、これに対応しています。ただし、もちろん文書中の文字コードは統一文字コードとして解釈され、ハングルとの併用などはできません。また、統一文字コードの表示には、統一文字コードに対応したフォントが必要となります。