



Proposal for Ancient Egyptian encoding in Unicode



We propose, on the line of L2/16-018, to use two main control characters to encode hieroglyphic texts.


When written in lines, Egyptian hieroglyphs are roughly organized in quadrants, as to fill an ideal square or rectangle. For instance, the sequence , which writes the word « p.t », « the sky », is often written as .

Along the lines of the Manuel de codage, which is currently the de-facto standard, we propose to use two control characters :


Those characters would be EGYPTIAN HIEROGLYPH HORIZONTAL JOINER (here rendered as « * ») and EGYPTIAN HIEROGLYPH VERTICAL JOINER (here rendered as « : »).


The « * » control groups horizontally ; the « : » control groups vertically, and has a greater precedence than « * ». It allows to create quadrants which are vertical boxes containing small horizontal groups.


So  will be rendered as .

A sentence like  « the sun is in the sky » would be encoded as







However, this simple quadrant layout is too limited. Sometimes, the content of one small horizontal group itself contains a vertical group. For instance : 



It happens a lot on monumental texts carved on stone ; and is less frequent (but still attested) in hieratic texts ( from Sinuhe Sinuhe AOS, vs. 18 text).

More complex groups, with two level of embedding, are also attested :  (Abydos temple of Ramesses II. p. 531-532.).

Our feeling is that the system should be able to note any depth of embedding, as it's quite likely that even more convoluted groups are possible.

Besides, the text often contain complex kerning combination of signs, like  or , where available space left by one sign is filled by others. Those complex combinations can involve a combine a single sign like  with a complex group like .

Our proposal :

For inner group, consider that the repetition of the operator gives it a greater precedence. For instance,  would be encoded as . (repeating the « :: »).

A group like  would be encoded as .

Advantage over the use of parenthesis :

The system, with unlimited depth, would be strictly equivalent to allowing parenthesis. However, It's easier to limit its depth of the system, and thus to use it with a technology which doesn't allow (or makes it difficult) to use context-free grammars (although allowing parenthesis with a fixed depth would also work).

Advantage over adding new operators : other proposals have suggested to add another level of operators to handle the phenomenon. The present system has the advantage that it can be extended if needed without changing the character list.

Combination with insertion operators

M.-J. Nederhof has proposed a number of precise insertion operators to handle cases like $\overset{\circ}{\mathbb{R}}$ or $\overset{\circ}{\mathbb{R}}$. If we want to combine them with “*” and “:”, we need to define their precedences and associativity. Both giving them higher and lower priority than * and : have their advantages and disadvantages. Groups like $\overset{\circ}{\mathbb{R}}$ or $\overset{\circ}{\mathbb{R}}$ might favour a higher priority; groups like $\overset{\circ}{\mathbb{R}}$ favour a lower one.

In any case, for those operators, we would also repeat the operator to increase its priority.

Practical implementation

We understand that the implementation will probably use mainly GSUB table.

In this respect, the previous considerations about grouping and associativity are mainly there to define what should be drawn. The system would probably try eagerly GSUB from the longest one to the shortest one (or does the Universal Shaping Engine allow more complex rendering ?)