Coding system for AUT-QE

de Bruijn, N.G.

Published: 01/01/1970

Citation for published version (APA):
Coding system for AUT-QE.

by N.G. de Bruijn.

The expressions and categories to be stored are all of the form EXPRESSION, as defined in the following syntax. The notion is a slight extension of those explained in [2] and [3].

The basic symbols are:

- type
- genre
- [ ]
- \{ \}
- ( )

and, furthermore, the elements of the sets <variable>, <constant> and <dummy variable>. These three sets are disjoint; <variable> and <constant> contain positive integers only; <dummy variable> contains integers ≤ -1000 only.

The notions <EXPRESSION> and <EXPRESSION string> are defined by:

- <EXPRESSION string> ::= <EXPRESSION> | <EXPRESSION string> , <EXPRESSION>
- <EXPRESSION> ::= type | genre | <constant> | <variable> | <dummy variable> |
  \{ <constant> (<EXPRESSION string>) \} \{ <EXPRESSION> \} <EXPRESSION> |
  [<dummy variable> , <EXPRESSION> ] <EXPRESSION>

There are three arrays in which the information about EXPRESSIONs and EXPRESSION strings is stored: list1[1:P], list2[1:P], list3[1:P].

Every integer k (1 ≤ k ≤ P) refers to an EXPRESSION string. In our present discussion we shall denote this string by \( \Omega_k \) (metalingual symbol). If \( \Omega_k \) has the form \( \Lambda_h \), \( \Lambda \) (where \( \Lambda \) is an EXPRESSION) then we have list 1[k] = h; if \( \Omega_k \) has the form \( \Lambda \), where \( \Lambda \) is an EXPRESSION, we have list 1[k] = 0. The information about \( \Lambda \) is stored in list 2[k] and list 3[k].

- If \( \Lambda = \text{type} \) then list 2[k] = 0, list 3[k] = -1000.
- If \( \Lambda = \text{genre} \) then list 2[k] = 0, list 3[k] = -2000.
- If \( \Lambda = c \), where \( c \in \text{<constant>} \), then list 2[k] = c, list 3[k] = 0.
- If \( \Lambda = x \), where \( x \in \text{<variable>} \), or \( x \in \text{<dummy variable>} \), then list2[k] = x, list3[k] = -5000 or -4000.

The entry -4000 should not be used if \( \Omega_k \) is not an indicator string (\( \Omega_k \) is certainly no indicator string if \( x \) is a dummy variable).

- If \( \Lambda \) has the form \( c(\text{<EXPRESSION string>} \), and if that EXPRESSION string is \( \Omega_h \), then list2[k] = c, list3[k] = h.
- If \( \Lambda \) has the form \( \{ \Lambda_1, \Lambda_2 \} \), and if \( \Omega_k \) is the EXPRESSION string \( \Lambda_1, \Lambda_2 \) (this string consists of just two expressions), then list2[k] = -12, list3[k] = h.
If \( \Lambda \) has the form \([t, \Lambda_1] \Lambda_2\), and if \( \Omega_h \) is the expression string \( \Lambda_1, \Lambda_2 \), then
\[
\text{list2}[k] = t, \quad \text{list3}[k] = h.
\]

Note that the above system is obtained from the one in [1] for expressions of the form \(<\text{constant}>\ (<\text{expression string}>)\) if we add the following conventions:

<table>
<thead>
<tr>
<th>type</th>
<th>is considered as</th>
</tr>
</thead>
<tbody>
<tr>
<td>genre</td>
<td>&quot; &quot; &quot;</td>
</tr>
<tr>
<td>c</td>
<td>&quot; &quot; &quot;</td>
</tr>
<tr>
<td>x</td>
<td>&quot; &quot; &quot;</td>
</tr>
<tr>
<td>t</td>
<td>&quot; &quot; &quot;</td>
</tr>
</tbody>
</table>

- \( [\Lambda_1] \Lambda_2 \) " " " \( -12(\Lambda_1, \Lambda_2) \)
- \( [t, \Lambda_1] \Lambda_2 \) " " " \( t(\Lambda_1, \Lambda_2) \)

We did not put the empty string into our syntax. Nevertheless we consider the empty string occasionally, and we give it list number 0, i.e. \( \Omega_0 \) represents the empty string.

We remind the reader of the definition of indicator string. An indicator string is either the empty string or a string of variables (satisfying the condition that the indicator string of the last variable is obtained by taking that last entry away). In the non-empty case it can, of course, be considered as an EXPRESSION string and will be stored as such.

The contents of a book are stored in three arrays: indstr[1:m], middle[1:m], cat[1:m].

If \( 1 \leq n \leq m \), and if the indicator string of the \( n \)-th line of the book is \( \Omega_k \), then \( \text{indstr}[n] = k \).

If the middle part of the \( n \)-th line is an EXPRESSION \( \Lambda \), and if \( \Omega_k \) is the string consisting of the single entry \( \Lambda \), then \( \text{middle}[n] = k \). (Note that \( \text{list1}[k] = 0 \) in this case.)

If the middle part of the \( n \)-th line is \( \text{PN} \), then \( \text{middle}[n] = -1 \).

If the middle part of the \( n \)-th line is \( \text{EB} \), and if \( \Omega_k \) is the extended indicator string of that line (i.e. the indicator string followed by \( n \)) then \( \text{middle}[n] = -100 -k \).

If the middle part of the \( n \)-th line is not \( \text{EB} \), and if \( \text{cat}[n] = k \), then \( \Omega_k \) is the EXPRESSION string consisting of just one entry, viz. the category part of the \( n \)-th line. (Whence \( \text{list1}[k] = 0 \) in this case.) If, however, the middle part is \( \text{EB} \), then \( \Omega_k \) is the category string of the extended indicator string of
that line. (If $x_1, \ldots, x_j, n$ is the extended indicator string, then this category string is $\Gamma_1, \ldots, \Gamma_j, \Gamma_{j+1}$, forming the categories of $x_1, \ldots, x_j, n$, respectively.)

Note that this difference between EB or non-EB applies to list1[cat[n]] only.

References.

