Published: 05/12/67

Identification

Index and Errata for BN.4.02
Robert R. Fenichel

Purpose

It is nearly impossible to find stuff in BN.4.02 without using this appendix.

Method

The code "(e)" denotes an example; "(d)" denotes a description.

Index

Errata

Page	Line	For	Read
9	6	in page 10	on page 8
	9	S	$
12	5	of B	B
13	-8	fixed	string
15	7	<sentence label>	<sentence label>|
	7.5	~	<sentence label><sentence middle>
16	-6	(<	<
17	-2	definitions	definition
	-3	definitions	definition
22	10	See subsection	——
	11	J, item #3.	p1 Acts like "=$($p1$)", even to the point of treating its detour like a continuation. See subsection J, item #3.
22	15	thet	that
	-1	$)	$)/x1
24	6.5	——	<bexpr>::=<bterm>|<bterm>. OR.<bexpr>
	7	<bexpr>::=	<bterm>::=
25	-7	]	.
27	4.5	——	<ls name>::=<id>
28	20	successful	successful, the blanks/ noblks mode is restored and

The following is a replacement for page 33.

```
*       Here we go.  You should know that "porcify" is a definition which
*   takes a split word (say, "squid," split into "squ" and "id") and
*   produces Pig Latin.  So the point of "bigword" is mainly just to split
*   words into pieces for "porcify."
*       We start by noting the input scan pointer, J.  We will use J
*   to compute the quantity of output which we have produced; in the
*   case of words starting with consonants, for example, the output
*   produced is as long as the input [ (ending J) - (starting J) ]
*   plus two for the added "ay."
*       Now we use "marks" and we start collecting letters.
*   Suppose the first letter is a vowel.  In this case, the detour of
*   the "vowel" component is irrelevant, and we proceed to "big-word-4."
*   The "letter*" component collects the rest of the word and
*   the "compute" ups the column-count to show the word and the
*   coming "yay".  Now we go to "porcify" with an imaginary word,
*   split into a leading "y" and a trailing string which is the
*   word we really found.
*       Suppose the first letter of the word is not a vowel ("big-word-1"),
*   but rather it is "y".  Then we skip the detour of the "y" component, and
*   we continue with big-word-3.  If a consonant follows the "y", then we
*   drop immediately to the next line, which is exactly where we were a
*   minute ago when the word began with a vowel.  If a non-consonant follows
*   the "y," however, we slyly reinitialize the letter-collector with a new
*   "marks" component.  Only then do we drift down to the next line, where
*   the code which thinks it is converting "am" into "amyay" is actually
*   converting "yam."
*       Finally, suppose the word begins with some letter not one of
*   [a,e,i,o,u,y].  We collect its initial string, collect the remainder,
*   and porcify the word split into this pair.
*       The punctuation-handler is rather an anticlimax.
*
punctuation..  blanks  compute(startword=j)
    marks  puncts*  install/(
         // paragraph/(=$( $))  compute(column-count=5)   =$(///$))
    compute(column-count=column-count+j -startword)=p1
*
*       Finally, the definitions
*
.definitions.
porcify=(2)$($q2$q1ay$)
p1=$($p1$)
p2=$($p2$)
p3=$($p3$)
y=$(y$)
ay=$(ay$)
end
```