Correcting the decompressor algorithm

In reviewing the June 1984 Computer article, “A Technique for High-Performance Data Compression,” we came across a problem with the decompressor algorithm when it deals with a special case. The fix for this is to replace the line:

FINchar → output;

with

FINchar → stack;

The corrected decompressor algorithm, is

1. Decompression: First input code → CODE → OLDcode;
2. with CODE = code (K), K → output;
3. 3. K → FINchar;
4. 4. Next Code: Next input code → CODE → INcode
5. 5. If no new code: EXIT
6. 6. If CODE not defined (special case):
7. 7. FINchar → stack;
8. 8. OLDcode → CODE;
9. 9. code (OLDcode, FINchar) → INcode;
10. 10. Next Symbol: IF CODE = code (ωK): K → stack;
11. 11. code (ω) → CODE;
12. 12. Go to Next Symbol;
13. 13. If CODE = code (K): K → output;
15. 15. Do while stack not empty:
16. 16. Stack top → output; POP stack;
17. 17. OLDcode, K → string table;
18. 18. INcode → OLDcode;
19. 19. Go to Next Code;

The compression of the string “ababcababab” results in output codes “124358” in Figure 5 of the article. The decompressor, upon encountering code 8, has just put code 7, which stands for the characters “bc,” into the string table, and FINchar is “b” because of the decoding of code 5. Since code 5 stands for two characters (i.e., “ba”), it is decoded by pushing “a” onto the stack, then “b” is output (which is followed by assigning “b” to FINchar), and finally “a” is popped off of the stack.

Code 8 is handled initially by the section of the algorithm where CODE is not defined (special case). The original algorithm calls for FINchar to be immediately output in the following four steps:

6. IF CODE not defined (special case):
7. FINchar → output;
8. OLDcode → CODE;
9. code (OLDcode, FINchar) → INcode;

The OLDcode → CODE transfer will cause code 5 (“ba”) to become the next code to be decoded. Note that step 7 causes “b” (FINchar) to go to output. In this way, the decoding of code 8 will be “bba,” and the decompressed string will be “ababcababab.” The last three characters will not be in order.

This problem can be corrected by substituting in step 7 the statement, FINchar → stack. In this way, FINchar, which is “b,” will be popped from the stack only after the OLDcode → CODE transfer causes code 5 to be decoded. To illustrate, code 5 (“ba”), after becoming CODE will be decoded by pushing “a” on top of the stack (forcing the down “b” in the stack) and then it will output the “b” that was part of “ba.” Popping the stack until it is empty will output “a” and “b” in succession. The resultant output for code 8 will then be “bba” and the original string, “ababcababab,” will be recovered with the last three letters in their correct order.