The key insight is that the machine needs some additional output symbols. Perhaps it can use A, B, and C to indicate that the cell previously contained the lower case version of the symbol, and also that the cell has already been processed. With these symbols, the machine can use a series of left-to-right scans that replace a single a-b-c non-adjacent sequence by A-B-C, effectively counting one of each input symbol. Then the machine can return to the left edge of the input string and look for another matched set a-b-c. Keep this up until a return sweep recognizes that all cells have been processed or until an unexpected input symbol is found. The following instructions can be developed:

$< s_0, \lambda, \lambda, \mathrm{S}, f >$	Recognize the empty string
$< s_0, a, A, \mathrm{R}, s_1 >$	The first a on this sweep
$< s_1, a, a, \mathrm{R}, s_1 >$	Skip over other as on this sweep
$< s_1, b, B, \mathrm{R}, s_2 >$	The first b on this sweep
$< s_2, b, b, \mathrm{R}, s_2 >$	Skip over other bs this sweep
$< s_2, c, C, \mathrm{L}, s_3 >$	The first c this sweep; start moving left
$< s_3, b, b, \mathrm{L}, s_3 >$	Ignore any bs when moving left
$< s_3, B, B, \mathrm{L}, s_3 >$	Ignore any Bs when moving left
$< s_3, a, a, \mathrm{L}, s_3 >$	Ignore any as when moving left
$< s_3, A, A, \mathrm{R}, ???>$	Past active region; start moving right

After using these instructions once, in sequence, the machine is ready to start sweeping right again. The only difference from the initial sweep right is that now there are As, Bs, and Cs on the tape. Additional instructions are needed to skip over them (in the proper order, of course).

How does the machine know it is done? It is done when a right sweep encounters only A, B, and C (but no lowercase a, b, or c). If the right sweep reads B in state s_0, move to state s_4. The instruction $< s_4, \lambda, \lambda, \mathrm{S}, f >$ will properly halt the machine at the first empty cell to the right of the string, after first verifying that all original symbols match in number and are in the proper order.

Here is the full list of instructions.

$< s_0, \lambda, \lambda, \mathrm{S}, f >$	Recognize an empty string
$< s_0, a, A, \mathrm{R}, s_1 >$	The first a on this sweep
$< s_0, B, B, \mathrm{R}, s_4 >$	This should be the final sweep
$< s_1, a, a, \mathrm{R}, s_1 >$	Skip over other as on this sweep
$< s_1, B, B, \mathrm{R}, s_1 >$	Skip over any Bs on this sweep
$< s_1, b, B, \mathrm{R}, s_2 >$	The first b on this sweep
$< s_2, b, b, \mathrm{R}, s_2 >$	Skip over other bs this sweep
$< s_2, C, C, \mathrm{R}, s_2 >$	Skip over any Cs on this sweep
$< s_2, c, C, \mathrm{L}, s_3 >$	The first c this sweep; start moving left
$< s_3, C, C, \mathrm{L}, s_3 >$	Ignore any Cs when moving left
$< s_3, b, b, \mathrm{L}, s_3 >$	Ignore any bs when moving left
$< s_3, B, B, \mathrm{L}, s_3 >$	Ignore any Bs when moving left
$< s_3, a, a, \mathrm{L}, s_3 >$	Ignore any as when moving left
$< s_3, A, A, \mathrm{R}, s_0 >$	Past active region; start moving right
$< s_4, B, B, \mathrm{R}, s_4 >$	Ignore any Bs in the final sweep
$< s_4, C, C, \mathrm{R}, s_4 >$	Ignore any Cs in the final sweep
$< s_4, \lambda, \lambda, \mathrm{S}, f >$	Recognize a valid string

■