1. abbab[] -> >abbab

If Then

| In state | Reading | Write | Go to | Move head |
| --- | --- | --- | --- | --- |
| Start state | a | Mark ~~ | Scan R until [] | R |
| Start state | b | Mark ~~ | Scan R until [] | R |
| Scan R until [] | a | Mark .. | Scan R until [] | R |
| Scan R until [] | b | Mark .. | Scan R until [] | R |
| Scan R until [] | [] | Mark ^ | Found [] | L |
| Found [] | a with .. | Mark ^ | Move a R | R |
| Found [] | b with .. | Mark ^ | Move b R | R |
| Move a R | a or b with ^ | a | Scan L until .. | L |
| Move b R | a or b with ^ | b | Scan L until .. | L |
| Scan L until .. | a with ^ | a with ^ | Scan L until .. | L |
| Scan L until .. | b with ^ | b with ^ | Scan L until .. | L |
| Scan L until .. | a with .. | Mark ^ | Move a R | R |
| Scan L until .. | a with .. | Mark ^ | Move b R | R |
| Scan L until .. | a or b with ~~ | > | Done | R |

1. 1. {w ∈ {a,b}\* | w has odd length and its middle char is a}
2. Move to first a or b
   1. If a or b cant be found, skip
3. If first char is a, then replace with 0
4. Scan head to next 0 or 1 or []
5. If char left of head is a, replace with 0
   1. Repeat from step 1 until recursion is done.
6. If char left of head is not a, then it’s done
7. If char left of head is b, then replace with 1
8. Scan head to next 0 or 1 or []
9. If char left of head is b, replace with 1
   1. Repeat from step 5 until recursion is done.
10. If char left of head is not b, then it’s done
11. If [] is not found, then reject
12. If there is something left of [], then replace a with 0
13. If thing left of [] is == to thing on right, then string is accepted
    1. Else reject string
    2. { | k = nm}
14. If string is not abc, then reject string
15. Mov head to left start
16. Scan right until b, marking each a
17. Mark pair of b and c
18. If no more c, then reject string
    1. else>
19. Unmark b until a
    1. At a, repeat from 3
20. If all a and c are marked, then accept string
    1. Else reject string
    2. { | n is a power of 2}
21. Start at left char, mark every other a
22. If only 1 a, then replace with 0
    1. If head has another a, then string accepted
    2. Else rejected
23. Repeat from 1
    1. If odd 0’s, then reject string
    2. Else accept