1 TM that accepts L=(aaaa^\*b^\*)

|  |  |
| --- | --- |
| Sub-Rule | Encode String |
| (q0,a) =(q1,a,R) | D1011011011011 |
| (q0,b)=(q5,b,R) | D1011101111110111011 |
| (q0,[])=(q5,[],R) | D101011111101011 |
| (q1,a)=(q2,a,R) | D110110111011011 |
| (q1,b)=(q5,b,R) | D11011101111110111011 |
| (q1,[])=(q5,[],R) | D1101011111101011 |
| (q2,b)=(q5,b,R) | D111011101111110111011 |
| (q2,[])=(q2,[],R) | D11101011101011 |
| (q2,a)=(q3,a,R) | D11101101111011011 |
| (q3,a)=(q3,a,R) | D111101101111011011 |
| (q3,b)=(q4,b,R) | D111101110111110111011 |
| (q3,[])=(q4,[],R) | D11110101111101011 |
| (q4,b)=(q4,b,R) | D1111101110111110111011 |
| (q4,a)=(q5,a,R) | D111110110111111011011 |

Final state is : Q0 q1 q2 q5.

W=aaaab=110110110110111 (expected accept )

reject

110110110110111D1011011011011D1011101111110111011D101011111101011D110110111011011D11011101111110111011D1101011111101011D111011101111110111011D11101011101011D11101101111011011D111101101111011011D111101110111110111011D11110101111101011D1111101110111110111011D111110110111111011011F101101110111111

W=aab = 110110111 (expected reject)

Accept

110110111D1011011011011D1011101111110111011D101011111101011D110110111011011D11011101111110111011D1101011111101011D111011101111110111011D11101011101011D11101101111011011D111101101111011011D111101110111110111011D11110101111101011D1111101110111110111011D111110110111111011011F101101110111111

W=bab=1110110111 (expected reject)

1110110111D1011011011011D1011101111110111011D101011111101011D110110111011011D11011101111110111011D1101011111101011D111011101111110111011D11101011101011D11101101111011011D111101101111011011D111101110111110111011D11110101111101011D1111101110111110111011D111110110111111011011F101101110111111

Accept

2.Desing a Tm for L=ab\*a

|  |  |
| --- | --- |
| Sub-Rule | Encode String |
| (q0,a) =(q1,x,R) | D1011011011011 |
| (q0,b)=(q4,b,R) | D101110111110111011 |
| (q0,[])=(q4,[],R) | D10101111101011 |
| (q1,b)=(q1,y,R) | D1101110110111011 |
| (q1,[])=(q4,[],R) | D110101111101011 |
| (q1,a)=(q2,x,R) | D110110111011011 |
| (q2,b)=(q4,b,R) | D11101110111110111011 |
| (q2,a)=(q4,a,R) | D111011011111011011 |
| (q2,[])=(q3,[],R) | D111010111101011 |

Accept states: q3. = 1111

W= aba= 110111011 (expected accept)

110111011D1011011011011D1101110110111011D110101111101011D101110111110111011D110110111011011D111011011111011011D11101110111110111011D111010111101011D10101111101011F1111

W=aab=110110111 (expected reject)

110110111D1011011011011D110110111011011D11101110111110111011D101110111110111011D10101111101011D1101110110111011D110101111101011D11101110111110111011D111010111101011F1111

3.Binary Counter. This program reads the current binary number printed on the tape and increments it by 1 before stopping.

|  |  |
| --- | --- |
| Sub-Rule | Encode String |
| (q0,[]) =(q1,[],R) | D10101101011 |
| (q0,0)=(q0,0,L) | D10110101101 |
| (q0,1)=(q0,1,L) | D1011101011101 |
| (q1,[])=(q2,1,L) | D11010111011101 |
| (q1,0)=(q2,1,R) | D1101101110111011 |
| (q1,1)=(q1,0,R) | D110111011011011 |
| (q2,[])=(q3,[],R) q3=final state | D111010111101011 |
| (q2,0)=(q2,0,L) | D111011011101101 |
| (q2,1)=(q2,1,L) | D11101110111011101 |

Comment ( []=1, 0=11, 1=111 )

W=10 = 111011 (expected to become W=11 = 1110111) increment by1

111011D10101101011D10110101101D1011101011101D11010111011101D1101101110111011D110111011011011D111010111101011D111011011101101D11101110111011101F1111

Accept