

Turing Machine

This example shows the construction of a Turing machine to perform the AND function. The two input bits are set up in adjacent squares, followed by a question mark, which will be replaced by the result. Two asterisks are placed at the ends of the data area to act as delimiters. The ten quintuples below specify the operation of the machine, though for any worked example (in this case 1 AND 1), only five of the ten will be used

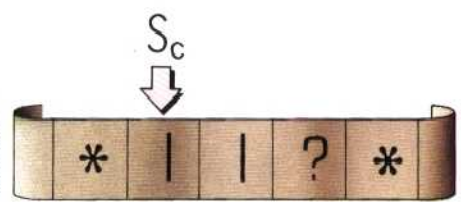
S _A	*	*	S _A	R
S _A	0	0	S _B	R
S _A	1	1	S _C	R
S _B	0	0	S _D	R
S _B	1	1	S _D	R
S _C	0	0	S _D	R
S _C	1	1	S _E	R
S _D	?	0	S _F	R
S _E	?	1	S _F	R
S _F	*	*	H	R



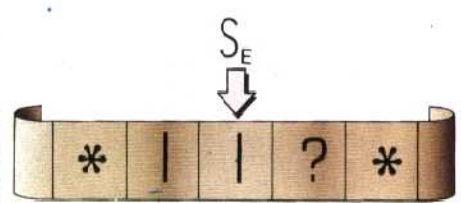
The machine starts off in state S_A with the head positioned over the leftmost asterisk. The only effect of this quintuple is to move the tape head to the right



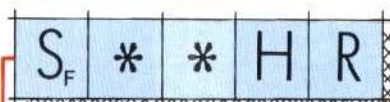
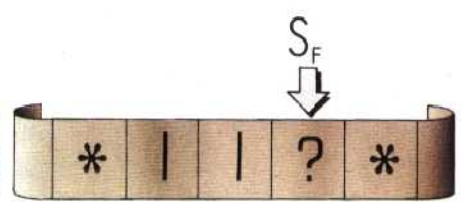
If the next square contains a 1, then this quintuple will be selected, and the machine goes into state S_C and is instructed to move right. If a 0 had been read, the outcome would be S_B



With the machine in state S_C, a 1 in the second square results in S_E. For all other eventualities, the machine would go into S_D



Reading a question mark, it is the state of the machine, S_E or S_D, that determines whether a 1 or a 0 is written in its place as the result. Either way, the machine is put into state S_F



The machine now enters the halt state (H) over the second asterisk. You can test out the operation on paper for 1 AND 0, 0 AND 1, and 0 AND 0

