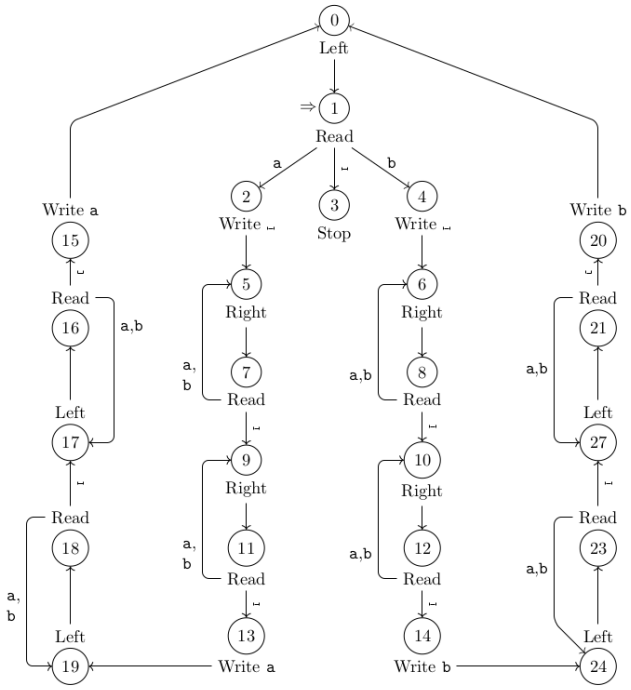
Turing Machines: Problems for Week 5
with Model Solutions

Exercise 1 Look at the Turing machine for the reversed copy, as shown below. Begin with a$\dot{\text{b}}$ in state 1 and trace execution until the machine halts (36 steps). This question is intended to illustrate how the program works.

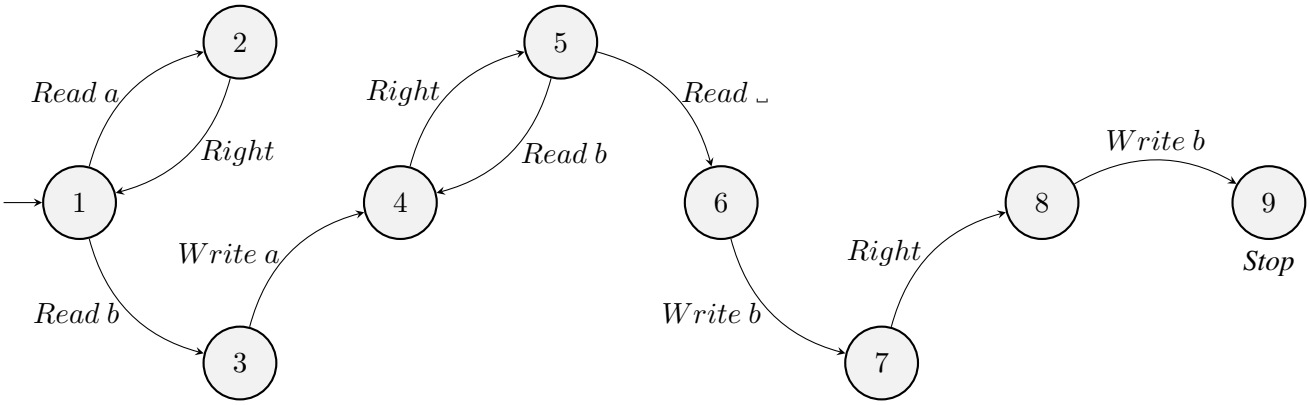


Solution 1

Step#	*Tape Contents*	*State*	*Next Step*	*Step#*	*Tape Contents*	*State*	*Next Step*
(1)	a$\dot{\text{b}}$	*1*	*Read* b	*(19)*	␣$\dot{b}$␣b	*7*	*Read* ␣
(2)	a$\dot{\text{b}}$	*4*	*Write* ␣	*(20)*	␣$\dot{b}$␣b	*9*	*Right*
(3)	a$\dot{\text{␣}}$	*6*	*Right*	*(21)*	␣b␣$\dot{b}$	*11*	*Read b*
(4)	a␣$\dot{\text{␣}}$	*8*	*Read* ␣	*(22)*	␣b␣$\dot{b}$	*9*	*Right*
(5)	a␣$\dot{\text{␣}}$	*10*	*Right*	*(23)*	␣b␣$\dot{b}$	*11*	*Read* ␣
(6)	a␣␣$\dot{\text{␣}}$	*12*	*Read* ␣	*(24)*	␣b␣$\dot{b}$	*13*	*Write a*
(7)	a␣␣$\dot{\text{␣}}$	*14*	*Write b*	*(25)*	␣b␣$b\dot{a}$	*19*	*Left*
(8)	a␣␣$\dot{b}$	*24*	*Left*	*(26)*	␣b␣$\dot{b}a$	*18*	*Read b*
(9)	a␣$\dot{\text{␣}}b$	*23*	*Read* ␣	*(27)*	␣b␣$\dot{b}a$	*19*	*Left*
(10)	a␣$\dot{\text{␣}}b$	*27*	*Left*	*(28)*	␣$\dot{b}$␣ba	*18*	*Read* ␣
(11)	a$\dot{\text{␣}}$␣b	*21*	*Read* ␣	*(29)*	␣$\dot{b}$␣ba	*17*	*Left*
(12)	a$\dot{\text{␣}}$␣b	*20*	*Write b*	*(30)*	␣$\dot{b}$␣ba	*16*	*Read b*
(13)	a$\dot{b}$␣b	*0*	*Left*	*(31)*	␣$\dot{b}$␣ba	*17*	*Left*
(14)	$\dot{a}b$␣b	*1*	*Read a*	*(32)*	$\dot{\text{␣}}b$␣ba	*16*	*Read* ␣
(15)	$\dot{a}b$␣b	*2*	*Write* ␣	*(33)*	$\dot{\text{␣}}b$␣ba	*15*	*Write a*
(16)	$\dot{\text{␣}}b$␣b	*5*	*Right*	*(34)*	$\dot{a}b$␣ba	*0*	*Left*
(17)	␣$\dot{b}$␣b	*7*	*Read b*	*(35)*	$\dot{\text{␣}}ab$␣ba	*1*	*Read* ␣
(18)	␣$\dot{b}$␣b	*5*	*Right*	*(36)*	$\dot{\text{␣}}ab$␣ba	*3*	*Stop*

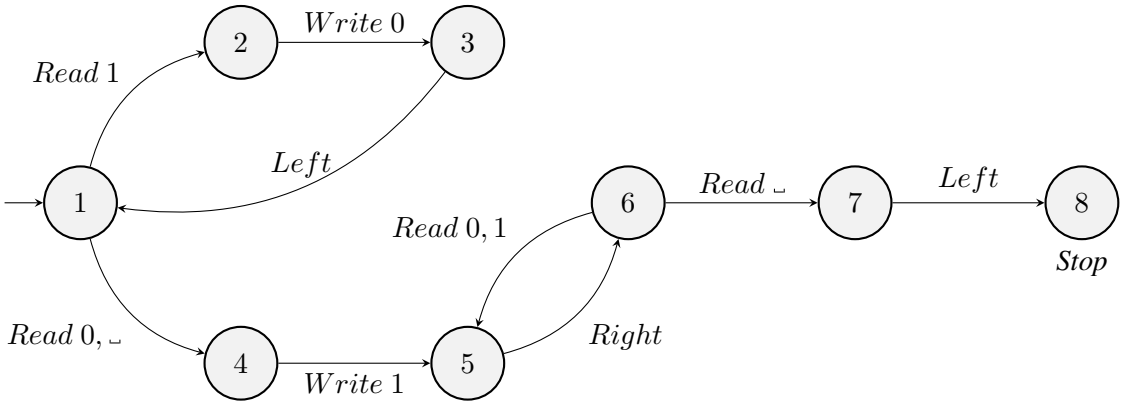
Exercise 2 Design a Turing machine that starts with on the leftmost cell of a block $\mathtt{a}^n\mathtt{b}^n$, with $n > 0$, on an otherwise blank tape, and ends on the rightmost cell of a block $\mathtt{a}^{n+1}\mathtt{b}^{n+1}$, on an otherwise blank tape. The leftmost character of the resulting block should be in the same position of the leftmost character of the initial block.

Solution 2



Exercise 3 Design a Turing machine that starts on the rightmost character (units digit) of a positive integer written in binary, and adds one. The new number should have its units digit in the same place as the original. You should end on the units digit. The tape alphabet is $\{0, 1, ␣\}$.

Solution 3

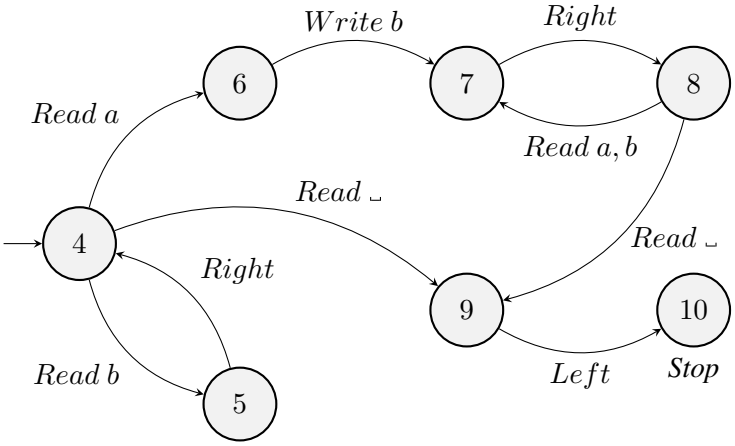


Exercise 4 (a) Design a Turing machine that starts on the leftmost character of a nonempty block on an otherwise blank tape. It converts the leftmost `a` to `b` and changes nothing else. If the block is entirely `b`'s, it should change nothing. It should end on the rightmost character. The tape alphabet is $\{\texttt{a}, \texttt{b}, \textvisiblespace\}$.

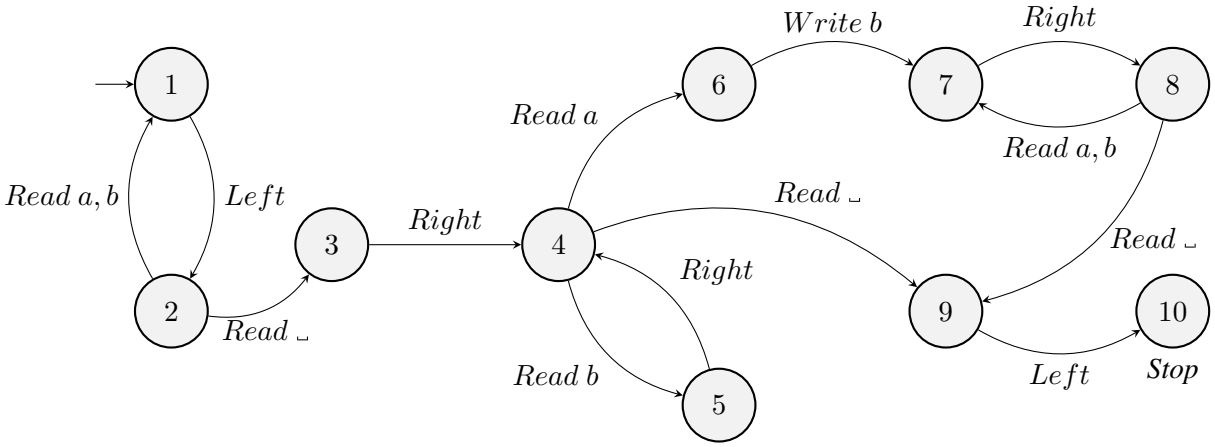
(b) (Bonus) How would you answer the previous question if you don't know where the head is at the start? It might be anywhere on the input block, including the leftmost or rightmost character.

Solution 4

(a) Turing machine to convert the leftmost `a` *to* `b` *and change nothing else.*



(b) Same Turing machine as above, without knowing the initial head position.



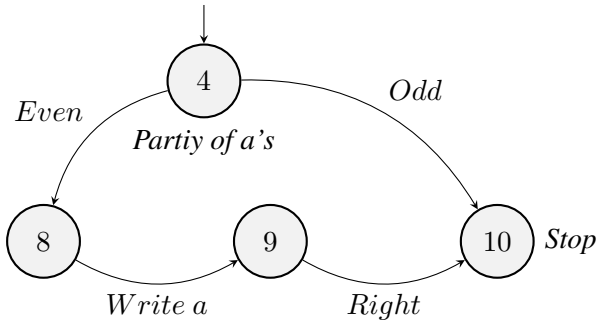
Exercise 5 In this question the tape alphabet is $\{\mathtt{a}, \mathtt{b}, \sqcup\}$. Starting on the leftmost character of a nonempty $\{\mathtt{a}, \mathtt{b}\}$-block on an otherwise blank tape, your task is as follows. If the block contains an even number of a's, append one a to the right of the block. Otherwise leave the tape contents unchanged. In each case you should end on the space to the right of the (resulting) block.

(a) Design a machine that solves this task, using the parity checking machine as a macro.

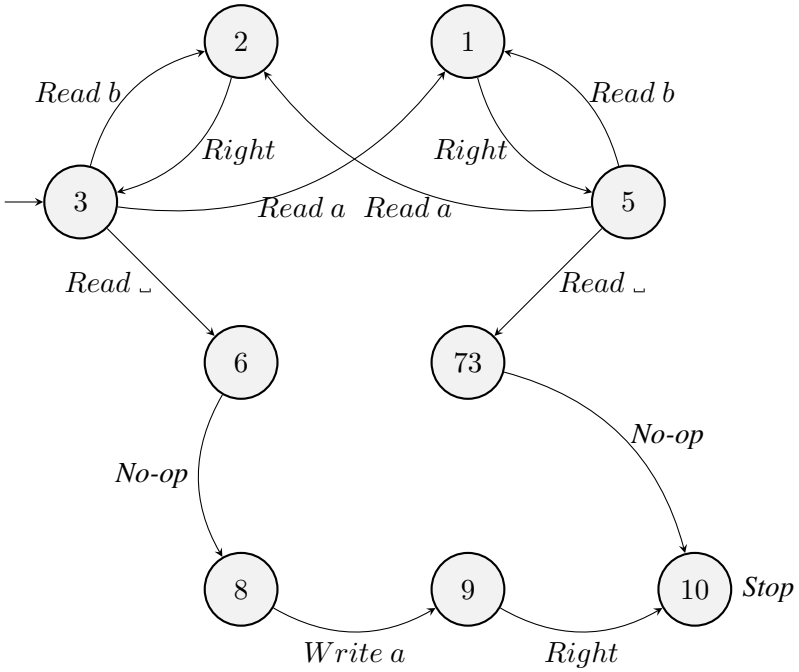
(b) Now give the full program by expanding the macro.

Solution 5

(a) Turing machine using the parity checking macro.



(b) Turing machine using the parity checking macro fully expanded.



Exercise 6 We would like to design a Turing machine that peforms a "shifting-over" operation. Ideally, we would like to create an extra cell at the current head position (wherever the head is), in which we could store some character. However, we cannot edit the tape in this way. Rather, we need to move the contents of each of the cells to the right of the current head position one cell right, and then find our way back to the current head position. For example, if the tape initially contains $1\dot{1}01$, with the tape head on the second 1 (from left), then you should obtain $1\dot{\sqcup}101$ at the end, with the head on the newly created extra cell. Design a Turing machine that will perform this operation, assuming that the Turing machine starts on a non-empty block on an otherwise blank tape.
Hint: Leave a special symbol, like the *blank* ($\sqcup$), to mark the position to which the head must return.

Solution 6