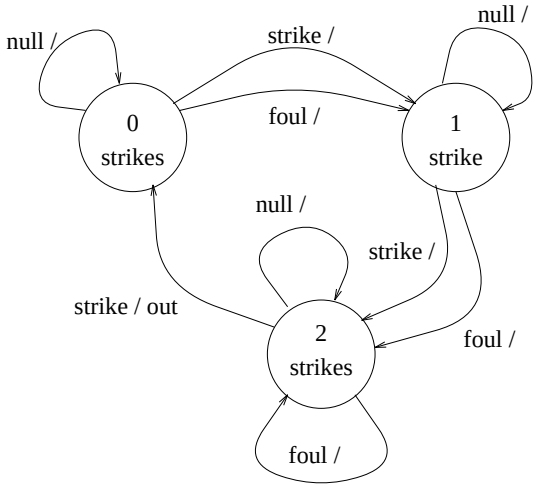
Strike State Machine

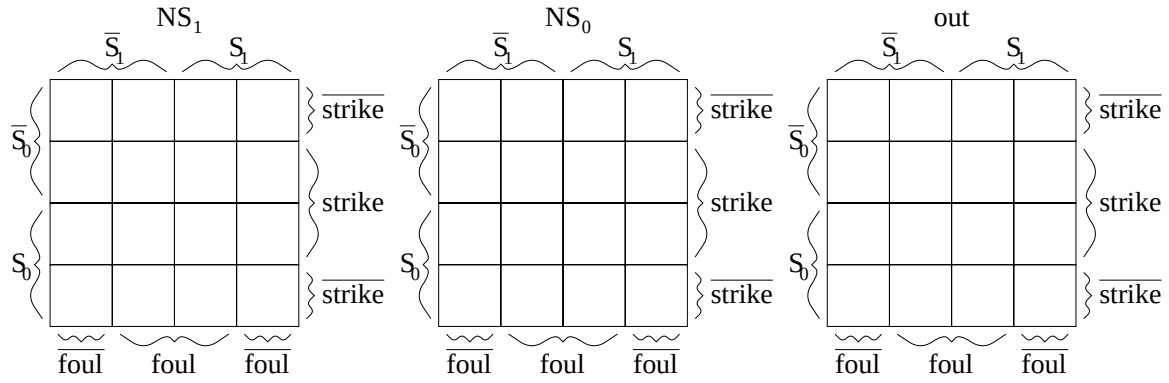
You're designing a baseball scoreboard. A sub-problem involves keeping track of strikes and fouls. A *strike* occurs when the batter swings at the ball and misses, or when the batter does not swing at a "good" pitch. If a batter earns three strikes during his/her turn at bat, he/she is "out". A *foul* occurs when the batter hits the ball, but the ball does not land in "fair" territory. Fouls are counted as strikes until the batter earns two strikes. Then fouls are ignored. The state diagram below captures these rules. The null input indicates neither a strike nor a foul occurred during that clock cycle. The strike and foul input cannot occur simultaneously, so the output for this case is don't care.



Part A Using the state diagram above, complete the state table for this diagram. The inputs are strike, foul, and state bits S_1 and S_0. The outputs are next state bits NS_1 and NS_0, and out.

S_1	S_0	$strike$	$foul$	NS_1	NS_0	out
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	1	0	0			
0	1	0	1			
0	1	1	0			
1	0	0	0			
1	0	0	1			
1	0	1	0			

Part B Determine the simplified logical expression for NS_1, NS_0, and *out* using your state table in part A. Complete the Karnaugh maps below, identify the prime implicants, and express the result.



$NS_1 =$ _____

$NS_0 =$ _____

$out =$ _____

Part C Implement this state machine using your expression from part B, plus two register cells. Use an icon for a one bit register cell. You do not have to show the implementation of a register here. Your logic should be simplified.