This is about the extra-credit problem where we define $L_c'$ to be the language of strings over \{a, b\} with no block of exactly 3 b's, and in which every block of 2 b's has an odd number of chars before it. The top picture is a DFA for this language, which has seven accepting states—ignore the extra final state with $\epsilon$-arcs for now. The strategy of this DFA is that the columns tell how many consecutive b's have been read. It uses the vertical dimension to keep track of whether the number of chars read so far is even or odd—but to make the diagram nicer I've flipped odd and even in columns 1 and 3. Note that state $2_0$ is accepting since it means we've read 2 b's starting with an odd number of chars before them, but state $2_e$ is rejecting and following it by an a goes to the dead state.

An “intuitive” regular expression is harder than for the intended regular-credit problem. There are more basic cycles. Here is a stab at it

$$L_{ss} = (ba + a(ba)^*a + ab(ab)^*ba + ab(ab)^*bb(ba)^*ba + bbb(ba)^*aa + \ldots$$

The ... is because there are other cycles involving the arc from state $4^+_e$ back down to state $0_{odd}$. We really need to convert this DFA rigorously into a regular expression. Because this DFA has more than one final state different from start, we need to do the step in the text of adding a new final state $f$ with $\epsilon$-arcs from old final states. To ward off the resulting cluttering-up of the diagrams, we delay doing this until we are about to eliminate a state. The following steps also eliminate states in pairs to save iterations of the drawing—you may check this at any stage by eliminating first the upper state of the pair and then the lower state. Apologies for crowding of the arcs making some labels hard to place—and it’s quite likely I still have a typo somewhere. The final step is skipped because it is essentially a two-state base case since the arcs to $f$ just get tacked on.
The final language is $L_{0,0} \cdot (bbb^* + b + \epsilon) \cup L_{0,0} \cdot bbbb^* + bb + b + \epsilon$.