We study the problem of finding good elimination orderings for the state elimination algorithm, which is one of the most popular algorithms for the conversion of finite automata into equivalent regular expressions. Based on graph separator techniques we are able to describe elimination strategies that remove states in large induced subgraphs that are "simple" like, e.g., independent sets or subgraphs of bounded treewidth, of the underlying automaton, that lead to regular expressions of moderate size. In particular, we show that there is an elimination ordering such that every language over a binary alphabet accepted by an $n$-state deterministic finite automaton has alphabetic width at most $O(1.742^n)$, which is, to our knowledge, the algorithm with currently the best known performance guarantee. Finally, we apply our technique to the question on the effect of language operations on regular expression size. In case of the intersection operation we prove an upper bound which matches, up to a small factor, a lower bound recently obtained in (Gelade and Neven, STACS 2008; Gruber and Holzer, ICALP 2008), and thus settles an open problem stated in (Ellul et al., JALC 2005).