## Source task and conventions

Decide whether arbitrary subsets, arbitrary supersets, and anagram closures of regular languages are regular.

## Conventional solution and proof

All three assertions are false over an alphabet containing $a, b$. (a) The nonregular language $K = \{a^n b^n : n \geq 0\}$ is a subset of the regular language $a^*b^*$. (b) The same $K$ is a superset of the regular language $\{\varepsilon\}$. To prove nonregularity, suppose $K$ had pumping length $p$. Every decomposition $a^p b^p = xyz$ with $|xy| \leq p$ and $|y| > 0$ has $y = a^k$ with $k > 0$. Pumping $y$ zero times gives $a^{p-k}b^p \notin K$, a contradiction. (c) Let $L = (ab)^*$. Its anagrams are exactly the words with equally many $a$ and $b$ symbols. If this anagram language were regular, its intersection with $a^*b^*$ would be $K$ and would be regular. Intersection closure follows from a product DFA that tracks both states and accepts precisely when both components accept. This contradicts the proof above.

## Structural or analytic derivation

The anagram operation preserves the commutative count vector (the Parikh vector) but discards order. A finite automaton recognizing $(ab)^*$ does not thereby yield a finite automaton testing equality of two unbounded counts. The invariant identifies the obstruction and the regular intersection isolates it rigorously.

## Comparative cost and preferred method

The counterexamples settle universal closure assertions with finite proofs; exhaustive testing of words through any finite length cannot establish them. The conventional pumping and product-automaton proof is the appropriate efficient method. There is no valid complex-analytic acceleration of this semantic classification.