Source task and conventions

Construct regular expressions for nonempty decimal strings representing exactly 42, not 42, and a value greater than 42.

Conventional solution and proof

Let $D = [0\text{-}9]$ and $N = [1\text{-}9]$ be alphabet shorthands. The three expressions are

$$(a)\quad 0^*42,$$
$$(b)\quad 0^*\big(0 \mid N \mid [1\text{-}3]D \mid 4[013\text{-}9] \mid [5\text{-}9]D \mid NDDD^*\big),$$
$$(c)\quad 0^*\big(4[3\text{-}9] \mid [5\text{-}9]D \mid NDDD^*\big).$$

In $NDDD^*$, the two explicit D symbols after N impose at least three significant digits. To prove the classification, remove all leading zeros from a nonzero string. One remaining digit has value 1–9. Two remaining digits represent 10–99 and the displayed alternatives partition these values, excluding 42 in (b) and keeping 43–99 in (c). At least three significant digits represent a value at least 100. An all-zero nonempty string is supplied by 0^*0 in (b). Every case is disjoint by significant length or first digit, and no expression accepts the empty string.

Structural or analytic derivation

A finite-state comparison automaton can store whether the significant prefix has length zero, one, two or at least three and, at length two, whether its value lies below, at or above 42. This gives a finite-state proof of regularity. The expressions above are preferable for the source's requested output because they directly enumerate the decisive cases.