

Two Clever Regular Expression for "All Strings except 001"

Ajitesh Dasaratha

September 2025

If you remember this question from labs and lectures, you'll remember the hint: *Don't try to be clever!*. So let's do exactly that.

The most brute force way from lab and lectures (and probably still how I'd write this is an exam) is:

$$\epsilon + 0 + 1 + 00 + 01 + 10 + 11 + 000 + 010 + 011 + 100 + 101 + 110 + 111 + (0 + 1)(0 + 1)(0 + 1)(0 + 1)(0 + 1)^*$$

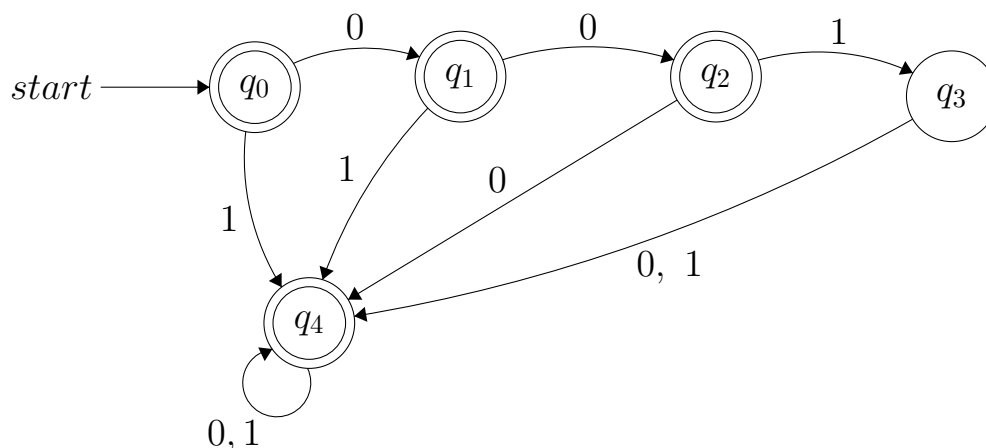
Let's clean this up.

First of all, you're allowed to use the notation $(0 + 1)^m$ as a shorthand for $(0 + 1)(0 + 1)\dots(0 + 1)\dots m \text{ times}$. That shortens the expression to $\epsilon + 0 + 1 + 00 + 01 + 10 + 11 + 000 + 010 + 011 + 100 + 101 + 110 + 111 + (0 + 1)^4(0 + 1)^*$. If $\Sigma = \{0, 1\}$, you can use Σ instead of $(0 + 1)$. So Σ^4 is a very succinct way to write $(0 + 1)(0 + 1)(0 + 1)(0 + 1)$. So now we have:

$$\epsilon + 0 + 1 + 00 + 01 + 10 + 11 + 000 + 010 + 011 + 100 + 101 + 110 + 111 + \Sigma^4\Sigma^*$$

This is just a notation trick though, let's actually try changing the logic of how the regular expression works.

Approach 1: Make DFA, Convert to Regular Expression



The great thing about this DFA is that since it doesn't have any loops, it's not too much of a headache to convert it into a regular expression. All we need to think about is: for each accepting state, what strings get us there?

- $q_0 : \epsilon$
- $q_1 : 0$
- $q_2 : 00$
- $q_4 : (1 + 01 + 000 + 001\Sigma)\Sigma^*$

(Quick explanation of the q_4 part: the strings 1, 01, 000, 001 Σ all get you to q_4 for the first time, then once you get there, you stay there forever, doesn't matter what characters you consume afterwards, hence the Σ^*)

And so the regular expression is just the union of all these strings, giving:

$$\epsilon + 0 + 00 + (1 + 01 + 000 + 001\Sigma)\Sigma^*$$

This isn't too bad, and if the banned string were 01101, we'd have:

$$\epsilon + 0 + 01 + 011 + 0110 + (1 + 00 + 010 + 0111 + 01100 + 01101\Sigma)\Sigma^*$$

But is there a way to use similar logic to the first brute force approach, and clean up from there?

Approach 2: Somehow clean up original expression

If we look at the original expression, the basic logic is: Brute force list out all strings with length 2 or less, brute force list of everything of length 3, then write a single expression for everything of length 4+.

The first part (length ≤ 2 can be rewritten as $(0 + 1 + \epsilon)^2$ (why?), which you're allowed to also write as $(\Sigma + \epsilon)^2$.

What about the strength with length 3 part? Let's see, how can string be 3 characters long and not be 001? It could contradict the character in either the first, second, or position: so that's $1\Sigma^2 + \Sigma 1\Sigma + \Sigma^2 0$. With that, the final expression is:

$$(\Sigma + \epsilon)^2 + 1\Sigma^2 + \Sigma 1\Sigma + \Sigma^2 0 + \Sigma^4 \Sigma^*$$

This also expands fairly well if the excluded string is longer. If we exclude 01101, then the expression is $(\Sigma + \epsilon)^4 + 1\Sigma^4 + \Sigma 0\Sigma^3 + \Sigma^2 0\Sigma^2 + \Sigma^3 1\Sigma + \Sigma^4 0 + \Sigma^6 \Sigma^*$.