

Testing Matching Performance

Meshal Binnasban

August 11, 2025

Testing Matching Performance

These tests are based on the regex enumeration code and the matching string regeneration code.

The algorithms tested are: 1. Derivatives with simplifications. 2. Play article’s original marked approach. 3. Shifts, a new variation developed with input-carrying marks.

The test computes the average of a fixed number of runs of each matcher for every regex–string pair.

1 First Tests

We start with 1,000,000 enumerated regexes and 9 matching strings. Each matcher was executed 1000 times per regex–string pair, and the average times (in seconds) were recorded. The results are:

$$\text{Derivatives} = 1.8074 \times 10^{-7}$$

$$\text{Play} = 2.9413 \times 10^{-7}$$

$$\text{Shifts} = 3.0245 \times 10^{-7}$$

Summary

The Derivatives matcher achieved the lowest average time, followed by Play, with Shifts being the slowest. These regexes and strings are sequentially enumerated, giving a potential advantage to the Derivatives approach.

2 Parallel Test with Batching

In this test, we enumerated 1,000,000 regexes in parallel, using a batch size of 100,000 regexes. For each regex, up to 9 matching strings were generated. Each algorithm was run 1000 times per regex–string pair, and the average times (in seconds) were recorded.

$$\text{Derivatives} = 3.997828078658495 \times 10^{-7}$$

$$\text{Play} = 6.267262704770589 \times 10^{-7}$$

$$\text{Shifts} = 1.151973014828378 \times 10^{-6}$$

Summary: In this parallel setting, the Derivatives matcher remained the fastest, followed by Play, while Shifts continued to be the slowest. The performance gap between Derivatives and Play widened compared to earlier sequential runs, and Shifts was significantly slower than both.