

# Additional content to CIAA 2010 submission 8: A Polynomial Time Match Test for Large Classes of Extended Regular Expressions

Daniel Reidenbach and Markus L. Schmid \*

Department of Computer Science, Loughborough University,  
Loughborough, Leicestershire, LE11 3TU, United Kingdom  
{D.Reidenbach,M.Schmid}@lboro.ac.uk

**Abstract.** This document presents additional content related to [1]. We discuss the results of some practical tests carried out to further substantiate our theoretical insights and to scrutinise the practical significance of Janus automata as presented in [1].

## Test Results

In order to further substantiate our theoretical insights and to scrutinise the practical significance of Janus automata, we discuss some test results. These tests compare the performance of the original REGEX engine of Java with a Java implementation of the Janus automaton<sup>1</sup>. This implementation is very basic and does not make use of any optimisations. Hence, it simply constructs the automaton for a given pattern  $\alpha$  and, in order to simulate the nondeterminism of our model, successively executes the corresponding computation for all possible counter bounds if their sum does not exceed the length of the input word  $w$ .

Our tests, performed on a standard PC, solve the membership problem for random instances  $(\alpha, w)$  with  $w \in \Sigma^+$ ,  $|\Sigma| = 2$ , and, on average,  $|w| \approx 3|\alpha|$ . For each  $x \in \text{var}(\alpha)$ ,  $20 \leq |\alpha|_x \leq 25$ . The six test scenarios considered depend on the size of  $\text{var}(\alpha)$ , which is 10 or 15, and on  $\text{vd}(\alpha)$ , which is 2, 3 or 4; in each of these scenarios, 500 instances are tested, comprising 250 examples with  $w \in L_\Sigma(\alpha)$  and 250 examples with  $w \notin L_\Sigma(\alpha)$ . If the test for a single instance is not completed within 20 minutes, then it is aborted. The following table shows the results of these tests. Here, “JREG” denotes the total runtime of the Java REGEX engine and “JANUS” the total runtime of the Janus automaton; after these runtimes, which are given in seconds, the total number of aborted tests is shown in parentheses.

---

\* Corresponding author.

<sup>1</sup> The Java source code as well as the original test sets are available at <http://www-staff.lboro.ac.uk/~coms10/>.

Instances	$ \text{var}(\alpha) $	$ w / \alpha $	$\text{vd}(\alpha)$	JANUS	JREG	JREG / JANUS
500	10	3.08	2	114 (0)	2653 (0)	23.2
500	10	3.08	3	5272 (0)	234903 (142)	44.56
500	10	3.08	4	36951 (0)	429527 (313)	11.62
500	15	3.16	2	440 (0)	10818 (0)	24.61
500	15	3.16	3	23506 (5)	371430 (232)	15.8
500	15	3.05	4	264239 (119)	526005 (403)	1.99

We can observe that the runtime of both algorithms substantially grows with the variable distance, and this holds for both choices of  $|\text{var}(\alpha)|$ . Hence, the results seem to confirm that the variable distance is a crucial parameter contributing to the complexity of the match test. Furthermore, we may state that, at least for the particular test scenarios considered, the Janus approach seems to be much more efficient than the established Java REGEX engine (in this context we wish to point out that, in many test scenarios, the evaluated runtime of the REGEX engine benefits from the large number of aborted tests). Therefore, and since we anticipate that even simple refinements of our implementation would yield a dramatically improved efficiency, the results strongly suggest that the Janus approach is practically worthwhile and deserves further studies.

## References

- [1] D. Reidenbach and M.L. Schmid. A polynomial time match test for large classes of extended regular expressions. In *Proc. 15th International Conference on Implementation and Application of Automata, CIAA 2010*, Lecture Notes in Computer Science, 2010. To appear.