# Z3-Noodler-Mocha at SMT-COMP 2025

Shaoke Cui, Chuan Luo*, Zhenwei Yang, Chunming Hu
*School of Software,Beihang University*
Beijing, China
{cuisk, chuanluo, zwyang, hucm}@buaa.edu.cn

## I. Introduction

*Z3-Noodler-Mocha* is a derived tool built on *Z3-Noodler* [1]. *Z3-Noodler* is the winner of the QF_Strings division in the Single Query Track. However, we found that some optimization can significantly improve the performance of *Z3-Noodler*.

## II. Z3-Noodler-Mocha

In this section, we introduce the core work of *Z3-Noodler-Mocha*.

### A. Lyndon Optimization

One optimization in *Z3-Noodler-Mocha* is based on the observation of automaton concatenation. Consistent with *Z3-Noodler* [1], we define the language $L(A)$ of the automaton $A$ to be the set of all words for which $A$ has an accepting run. Suppose there are $n$ automata $A_1, A_2, ..., A_n$, we can get the following theorem, which is trivial to prove:

**Theorem 1.** $\forall 1 \leq i \leq n, L(A_i) \in w^* \implies \forall \sigma \in S_n, \mathrm{concat}(A_1, \ldots, A_n) = \mathrm{concat}(A_{\sigma(1)}, \ldots, A_{\sigma(n)})$.

$w$ is an arbitrary word and $S_n$ denotes all possible permutations from $1$ to $n$. For this equation constraint, existing solvers can easily prove that it is SAT, since the equality always holds. But it is difficult for them to prove that the corresponding inequality is UNSAT, which requires exhausting the entire search space. *Z3-Noodler-Mocha* proposed an optimization method dubbed *Lyndon Optimization*, which is based on the KMP algorithm [2] and tries to find out whether $L(A) \in w^*$ holds and removes such constraints in the preprocessing stage.

Furthermore, according to the Lyndon–Schützenberger Theorem [3], we can conclude the following lemma:

**Lemma 1.** $\forall \sigma \in S_n, \mathrm{concat}(A_1, A_2, \ldots, A_n) = \mathrm{concat}(A_{\sigma(1)}, A_{\sigma(2)}, ..., A_{\sigma(n)}) \implies \forall 1 \leq i \leq n, L(A_i) \in w^*$.

Without loss of generality, the optimization method used by *Z3-Noodler-Mocha* can effectively eliminate a class of constraints.

### B. Regular Constraint Extraction

*Regular Constraint Extraction* (RCE) is an optimization algorithm implemented in *Z3-Noodler-Mocha*. For the following word equations, RCE will extract the regular constraints that are implicitly embedded in word equations.

$$sV_1V_2 \ldots V_n = V_1V_2 \ldots V_n \tag{1}$$

$$V_1V_2 \ldots V_n s = V_1V_2 \ldots V_n \tag{2}$$

$s$ is a string literal and $V$ is a string variable.

In the equation 1, we can infer that $V_1 \in s^*.x$, where $x$ is an arbitrary prefix of $s$. In the equation 2, we can infer that $V_n \in y.s^*$, where $y$ is an arbitrary suffix of $s$. Both $x$ and $y$ may be empty (*i.e.,* each may have length zero).

### C. Pruning Optimization

The main goal of *Pruning Optimization* (PO) is to reduce the search space of *Nielsen Transformation* and *Stabilization-based procedure* in *Z3-Noodler*. In the *Nielsen Transformation*, we improved the judgement of unsatisfiable predicates to reduce the impact of search space explosion on the decision procedure. Besides, we optimize *Stabilization-based procedure* by merging the constraints of the transducers with the same input and output.

## III. Acknowledgment

The authors would like to thank the developers of the *Z3-Noodler* solver and all the authors who contributed to the modifications that have been integrated. Their solver is the foundation of our improvement.

## References

[1] D. Chocholatý, V. Havlena, L. Holík, J. Hranicka, O. Lengál, and J. Síc, "Z3-noodler 1.3: Shepherding decision procedures for strings with model generation," in *Tools and Algorithms for the Construction and Analysis of Systems - 31st International Conference, TACAS 2025, Held as Part of the International Joint Conferences on Theory and Practice of Software, ETAPS 2025, Hamilton, ON, Canada, May 3-8, 2025, Proceedings, Part II*, ser. Lecture Notes in Computer Science, A. Gurfinkel and M. Heule, Eds., vol. 15697. Springer, 2025, pp. 23–44.

[2] M. Régnier, "Knuth-morris-pratt algorithm: An analysis," in *Mathematical Foundations of Computer Science 1989, MFCS'89, Porabka-Kozubnik, Poland, August 28 - September 1, 1989, Proceedings*, ser. Lecture Notes in Computer Science, A. Kreczmar and G. Mirkowska, Eds., vol. 379. Springer, 1989, pp. 431–444.

[3] P. Dömösi and G. Horváth, "Alternative proof of the lyndon-schützenberger theorem," *Theor. Comput. Sci.*, vol. 366, no. 3, pp. 194–198, 2006. [Online]. Available: https://doi.org/10.1016/j.tcs.2006.08.023

* Chuan Luo is the corresponding author.