THEORIE-TAG 2023 A. Lin, G. Zetzsche (Eds.): Theorietag 2023, Kaiserslautern, 4. – 6.10.2023
RPTU & MPI-SWS Kaiserslautern, Technical Report, pp. 51–53.

The Pumping Lemma for Regular Languages is Hard

Hermann Gruber$^{(A)}$ Markus Holzer$^{(B)}$ Christian Rauch$^{(B)}$

$^{(A)}$Planerio GmbH, Gewürzmühlstr. 11, 80538 München
h.gruber@planerio.de

$^{(B)}$Institut für Informatik, Universität Giessen
Arndtstr. 2, 35392 Giessen
{holzer,christian.rauch}@informatik.uni-giessen.de

The automata theory and formal languages curriculum introduces pumping lemmata for regular and context-free languages to demonstrate non-regularity or non-context-freeness in specific cases. Variations of these lemmata are taught based on instructor preferences and chosen materials. For example, refer to the pumping lemma in [6, page 70, Theorem 11.1], which outlines a key criterion for language regularity.

Lemma 1 *Let L be a regular language over Σ. Then, there is a constant p (depending on L) such that the following holds: If $w \in L$ and $|w| \geq p$, then there are words $x \in \Sigma^*$, $y \in \Sigma^+$, and $z \in \Sigma^*$ such that $w = xyz$ and $xy^tz \in L$ for $t \geq 0$—it is then said that y can be* pumped *in w.*

A lesser-known pumping lemma, attributed to Jaffe [5], characterizes the regular languages, by describing a necessary and sufficient condition for languages to be regular. For other pumping lemmata see, e.g., the annotated bibliography on pumping [7]:

Lemma 2 *A language L is regular if and only if there is a constant p (depending on L) such that the following holds: If $w \in \Sigma^*$ and $|w| = p$, then there are words $x \in \Sigma^*$, $y \in \Sigma^+$, and $z \in \Sigma^*$ such that $w = xyz$ and*[1]

$$wv = xyzv \in L \iff xy^tzv \in L$$

for all $t \geq 0$ and each $v \in \Sigma^*$.

For a regular language L the value of p in Lemma 1 can always be chosen to be the number of states of a finite automaton, regardless whether it is deterministic (DFA) or nondeterministic (NFA), accepting L. Sometimes an even smaller number suffices. For instance, the language

$$L = a^* + a^*bb^* + a^*bb^*aa^* + a^*bb^*aa^*bb^*,$$

is accepted by a (minimal) deterministic finite automaton with five states, the sink state included, but for $p = 1$ the statement of Lemma 1 is satisfied since regardless whether the considered word

This is a summary of a paper presented at the 27th International Conference on Implementation and Application of Automata (CIAA) held in Famagusta, Cyprus, September 19–22, 2023.

[1] Observe that the words $w = xyz$ and xy^tz, for all $t \geq 0$, belong to the same Myhill-Nerode equivalence class of the language L. Thus, one can say that the pumping of the word y in w *respects equivalence classes*.

starts with a or b, this letter can be readily pumped. For Lemma 2 the situation is even more involved and we refer to [2] and [3] for a detailed discussion on that subject. This gives rise to the definition of the LANGUAGE-PUMPING-PROBLEM or for short PUMPING-PROBLEM:

INPUT: a finite automaton A and a natural number p, i.e., an encoding $\langle A, 1^p \rangle$.

OUTPUT: Yes, if and only if the statement from Lemma 1 holds for the language $L(A)$ w.r.t. the value p.

A similar definition applies when considering the condition of Lemma 2 instead.

These problems turn out to be surprisingly difficult, even in the case of deterministic finite automata as inputs. The following table summarizes our findings for finite automata in general. The **coNP**-hardness result for NFAs gives us a nice non-approximability by-product under the

	PUMPING-PROBLEM w.r.t. ...	
	Lemma 1	Lemma 2
DFA	**coNP**-complete	
NFA	**coNP**-hard contained in $\Pi_2^{\mathbf{P}}$	**PSPACE**-complete

Table 1: Complexity of the PUMPING-PROBLEM for variants of finite state devices in general.

assumption of the so-called *Exponential-Time Hypothesis (ETH)* [1, 4]: there is no deterministic algorithm that solves 3SAT in time $2^{o(n+m)}$, where n and m are the number of variables and clauses, respectively. More precisely we find the following non-approximability statement:

Theorem 1 *Let A be an NFA with s states, and let δ be a constant such that $0 < \delta \leq 1/2$. Then no deterministic $2^{o(s^\delta)}$-time algorithm can approximate the minimal pumping constant w.r.t. Lemma 1 (Lemma 2, respectively) within a factor of $o(s^{1-\delta})$, unless ETH fails.*

When considering restricted automata such as, e.g., unary automata, the situation changes dramatically. For NFAs the **coNP**-hardness result and thus its intractability remains for both considered pumping lemmata, while for DFAs the problem becomes efficiently solvable. More precisely, for both pumping lemmata the PUMPING-PROBLEM can be shown to be complete for deterministic logspace **L** under weak reductions.

References

[1] M. CYGAN, F. FOMIN, Ł. KOWALIK, D. LOKSHTANOV, D. MARX, M. PILIPCZUK, M. PILIPCZUK, S. SAURABH, *Parameterized Algorithms*, chapter Lower Bounds Based on the Exponential-Time Hypothesis. Springer, 2015, 467–521.

[2] H. GRUBER, M. HOLZER, C. RAUCH, The Pumping Lemma for Regular Languages is Hard. In: B. NAGY (ed.), *Proceedings of the 27th International Conference on Implementation and Application of Automata*. Number 14151 in LNCS, Springer, Famagusta, Cyprus, 2023, 128–140.

[3] M. HOLZER, C. RAUCH, On Jaffe's Pumping Lemma, Revisited. In: H. BORDIHN, N. TRAN, G. VASZIL (eds.), *Proceedings of the 25th International Conference on Descriptional Complexity of Formal Systems*. Number 13918 in LNCS, Springer, Potsdam, Germany, 2023, 65–78.

[4] R. IMPAGLIAZZO, R. PATURI, F. ZANE, Which Problems Have Strongly Exponential Complexity? *J. Comput. System Sci.* **63** (2001) 4, 512–530.

[5] J. JAFFE, A necessary and sufficient pumping lemma for regular languages. *SIGACT News* **10** (1978) 2, 48–49.

[6] D. C. KOZEN, *Automata and Computability*. Undergraduate Texts in Computer Science, Springer, 1997.

[7] A. NIJHOLT, YABBER—Yet Another Bibliography: Pumping Lemma's. An Annotated Bibliography of Pumping. *Bull. EATCS* **17** (1982), 34–53.