On the Fine Grained Complexity of Finite Automata Non-Emptiness of Intersection

Michael Wehar
Swarthmore College

Mateus de Oliveira Oliveira
Bergen University

Developments in Language Theory 2021

Presented on August 20th, 2021
Outline

1 Introduction
   • Motivation
   • Problem Statement

2 Known Hardness Results

3 Our Contributions
   • Refined Simulation
   • Stronger Hardness Results

4 Conclusion
Regular languages are closed under intersection by the Rabin-Scott product construction.

Applying the product construction on DFA’s $D_1$ and $D_2$, the product automaton has up to $|D_1| \cdot |D_2|$ states.

In the worst case, we really need all of those states to capture the intersection language.

To answer some basic queries about the intersection of regular languages, the only known approach is to fully searching through a large Cartesian product graph.
Non-Emptiness of Intersection for DFA’s (DFA-NEI)

Given a finite list of DFA’s, is there a string that is accepted all of the DFA’s? In other words, do the regular languages associated with the DFA’s have a non-empty intersection?

- This is a classic PSPACE-complete problem [Kozen 1977].
- The standard solution considers the state diagram of the Rabin-Scott product automaton.
- For $k$ DFA’s with $n$ states each, we can solve $k$-DFA-NEI in $O(n^k)$ time.
Outline

1 Introduction
   • Motivation
   • Problem Statement

2 Known Hardness Results

3 Our Contributions
   • Refined Simulation
   • Stronger Hardness Results

4 Conclusion
There are many conditional lower bound results for the $k$-DFA-NEI problem going back to Kasai and Iwata 1985.

We will focus on two specific conditional lower bounds.

**Theorem (Karakostas, Lipton, and Viglas 2003)**

If $k$-DFA-NEI is solvable in $n^{o(k)}$ time, then $\text{NTIME}(n) \subseteq \text{DTIME}(2^{o(n)})$.

**Theorem (Fernau and Krebs 2017)**

If $k$-DFA-NEI is solvable in $n^{o(k)}$ time, then ETH is false i.e. 3-SAT is solvable in $\text{poly}(N) \cdot 2^{o(n)}$ time.
A 2-tape Binary Turing machine is a machine with a two-way read-only input tape and a two-way binary work tape.

More formally, it is a tuple $M = (Q, \{0, 1\}, c, q_0, F, \delta)$ where $Q$ is a set of states, $c$ is the maximum number of occurrences of special delimiter symbol $\#$, $q_0 \in Q$ is an initial state, $F$ is a set of final states, and

$$\delta : Q \times (\{0, 1\} \cup \{\#\})^2 \rightarrow \mathcal{P}(Q \times \{-1, 0, 1\}^2 \times (\{0, 1\} \cup \{\#\}))$$

is a partial transition function that assigns to each triple

$$(q, b_1, b_2) \in Q \times (\{0, 1\} \cup \{\#\})^2,$$

a set of tuples

$$\delta(q, b_1, b_2) \subseteq Q \times \{-1, 0, 1\}^2 \times (\{0, 1\} \cup \{\#\}).$$
Let a nondeterministic $m$-state 2-tape Binary Turing machine $M$ with at most $c$ occurrences of symbol $#$ and an input string $x$ of length $n$ be given.

Let $\sigma$ denote the number of cells used on $M$’s work tape.

**Theorem (Refined Simulation)**

For every $k$, we can efficiently compute $k$ DFA’s $\langle A_1, A_2, \ldots, A_k \rangle$ each with a binary input alphabet and $O(m^2 \cdot n \cdot \sigma^{1+c} \cdot 2^{\frac{\sigma}{k}})$ states such that $M$ accepts $x$ if and only if $\bigcap_{i=1}^{k} L(A_i) \neq \emptyset$. 

M. Wehar and M. Oliveira

Finite Automata Non-Emptiness of Intersection
Theorem (Refined Simulation)

For every $k$, we can efficiently compute $k$ DFA’s $\langle A_1, A_2, \ldots, A_k \rangle$ each with a binary input alphabet and $O(m^2 \cdot n \cdot \sigma^{1+c} \cdot 2^{\frac{\sigma}{k}})$ states such that $M$ accepts $x$ if and only if $\bigcap_{i=1}^{k} L(A_i) \neq \emptyset$.

- We construct the $k$ DFA’s to read in a binary string that encodes a Turing machine instruction sequence.
- As the DFA’s read the instruction sequence they each keep track of a different part of the work tape so that each automaton only needs to keep track of $\frac{\sigma}{k}$ work tape cells.
- The DFA’s collectively verify that the instruction sequence corresponds with a valid and accepting configuration sequence for $M$ on input $x$. 
From KLV 2003, we know that we cannot solve $k$-DFA-NEI more efficiently unless there exist faster algorithms for nondeterministic linear time.

By applying the simulation with $\sigma = n$, we strengthen this conditional lower bound to nondeterministic linear space.

Theorem (First Result)

*If $k$-DFA-NEI is solvable in $n^o(k)$ time, then $\text{NSPACE}(n) \subseteq \text{DTIME}(2^o(n))$.***
From Fernau and Krebs 2017, we know that we cannot solve $k$-DFA-NEI more efficiently unless there exist faster algorithms for 3-SAT.

Circuit-SAT for circuits with $n$ inputs, depth $d$, and size $s$ can be solved nondeterministically using $n + O(d) + O(\log(s))$ work tape cells.

By applying the simulation, we strengthen the conditional lower bound to linear depth and subexponential size circuits.

**Theorem (Second Result)**

*If $k$-DFA-NEI is solvable in $n^{o(k)}$ time, then satisfiability for $n$ input, depth $O(n)$, and size $2^{o(n)}$ Boolean circuits is solvable in $2^{o(n)}$ time.*
Circuit Lower Bounds

- From Abboud, Hansen, Williams, and Williams 2016, we know that more efficient algorithms for SAT imply circuit lower bounds.

- Therefore, it follows that faster algorithms for $k$-DFA-NEI would imply circuit lower bounds.

Strong Exponential Time Hypothesis (SETH)

- We also consider the consequences of smaller runtime improvements for $k$-DFA-NEI.

- In particular, if $k$-DFA-NEI is solvable in $O(n^{k-\varepsilon})$ time, then SETH (and NC-SETH) would be false.
Outline

1. Introduction
   - Motivation
   - Problem Statement

2. Known Hardness Results

3. Our Contributions
   - Refined Simulation
   - Stronger Hardness Results

4. Conclusion
Conclusion

Summary

- Answering basic queries about the intersection of regular languages (such as non-emptiness) is really hard.
- Finding even slightly faster algorithms for $k$-DFA-NEI would imply subexponential time algorithms for nondeterministic linear space problems and Circuit-SAT.

Future Work

- Is there any hope at proving unconditional time complexity lower bounds for the non-emptiness of intersection problem?
- Are there restricted classes of regular languages where the non-emptiness of intersection problem has more efficient algorithms?
Thank you!