Intersection Non-Emptiness and Hardness within Polynomial Time

Michael Wehar
Temple University

Mateus de Oliveira Oliveira
Bergen University

Developments in Language Theory 2018

September 14, 2018

Overview

1. Background
2. Our Contribution
3. Unary Finite Automata
4. Main Result

Intersection Non-Emptiness Problem

Intersection Non-Emptiness for DFA's

Given a finite list of DFA's, is there a string that is accepted all of the DFA's?

- This is a classic PSPACE-complete problem [Kozen 1977].
- Standard solution considers the state diagram of the Cartesian product automaton.
- For k DFA's with n states each, we can solve this problem in $O(n^k)$ time.

It's Hard to Find Faster Algorithms

- **More recent work:** If we can solve intersection non-emptiness in $n^{o(k)}$ time, then
  - there exist faster algorithms for several hard problems[1]
  - the complexity classes NL and P are not equal[2]
- If for some $k \in \mathbb{N}$ and $\varepsilon > 0$ we can solve intersection non-emptiness for k DFA's in $O(n^{k-\varepsilon})$ time, then
  - there exist slightly faster algorithms for SAT and QBF[3]
- Although these results are more recent, conditional lower bounds for intersection non-emptiness with a fixed number of DFA's go back to Kasai and Iwata 1985.

[1] Several papers including KLV 2003 and Fernau and Krebs 2016
[2] Wehar 2014
[3] Wehar 2016

Triangle Finding and 3SUM

- **Triangle Finding:** Given a Graph G with n vertices, do there exist vertices a, b, and c such that a and b are adjacent, b and c are adjacent, and c and a are adjacent?
  - Solvable in $O(n^{2.373})$ time by a reduction to matrix multiplication
- **3SUM:** Given a set of integers S of size n, do there exist elements a, b, and c such that $a + b + c = 0$?
  - Solvable in $O(n^2)$ time and sublogarithmic factor improvements are known
- It is considered hard to find faster algorithms for Triangle Finding and 3SUM.

Overview

1 Background

2 Our Contribution

3 Unary Finite Automata

4 Main Result

Direct Reductions

- We investigate the intersection non-emptiness problem when $k = 2$ and $k = 3$.
- We provide direct reductions:
  - From Triangle Finding for a graph with n vertices and m edges to intersection non-emptiness for two DFA's where the first DFA has $O(m \log(n))$ states and the second DFA has $O(n \log(n))$ states.
  - From 3SUM for a set of n integers in $[-n^k, n^k]$ to non-emptiness of intersection for three DFA's where each DFA has $O(kn \log(n))$ states.

Faster Algorithms

- We show that there exist faster algorithms when the DFA's are restricted to a unary input alphabet.
- Intersection non-emptiness for two unary DFA's is solvable in linear time.
- We connect the complexity of intersection non-emptiess for three unary DFA's to the complexity of Triangle Finding:
  - **Main Result:** For every $\alpha > 0$, intersection non-emptiness for three unary DFA's is solvable in $O(n^{\frac{\alpha}{2}})$ time if and only if Triangle Finding is solvable in $O(n^{\alpha})$ time.
  - Since Triangle Finding can be solved in $O(n^{2.373})$ time, intersection non-emptiness for three unary DFA's can be solved in $O(n^{1.1865})$ time.

Overview

1 Background

2 Our Contribution

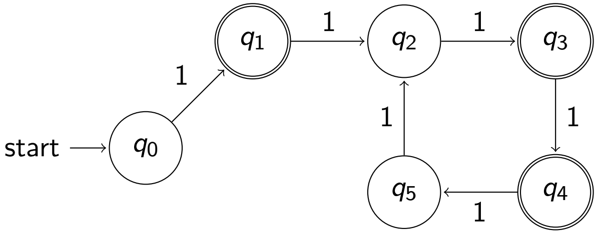
3 Unary Finite Automata

4 Main Result

Unary Finite Automata

- A unary DFA is simply a DFA over a unary input alphabet.
- Such automata consist of a segment and a cycle.
- Each final state from a unary DFA can be viewed as a constraint over natural numbers.
- It is known that intersection non-emptiness for unary DFA's is NP-complete [Stockmeyer and Meyer 1973].

Unary DFA: Example



- The segment has length two and the cycle has length four.
- The DFA accepts:
  - the string of length 1
  - strings with length congruent to 3 mod 4
  - strings with length greater than 0 and divisible by 4

Overview

1. Background
2. Our Contribution
3. Unary Finite Automata
4. Main Result

Main Result: Part 1

Theorem

If intersection non-emptiness for three unary DFA's $\in DTIME(n^{\frac{\alpha}{2}})$, then Triangle Finding $\in DTIME(n^{\alpha})$.

- Given a graph G with n vertices, we construct DFA's D_1, D_2, and D_3 with $O(n^2)$ states each such that G contains a triangle if and only if $L(D_1) \cap L(D_2) \cap L(D_3) \neq \emptyset$.
- We can compute three relatively prime integers a_1, a_2, and $a_3 \in [n, 5n]$ such that:
  - D_1 will have cycle length $a_1 \cdot a_2$
  - D_2 will have cycle length $a_2 \cdot a_3$
  - D_3 will have cycle length $a_3 \cdot a_1$

Main Result: Part 1

Theorem

If intersection non-emptiness for three unary DFA's $\in DTIME(n^{\frac{\alpha}{2}})$, then Triangle Finding $\in DTIME(n^{\alpha})$.

- The DFA's read in an integer that is uniquely associated with a triple of integers (i, j, k).
- This association is a bijective mapping from triples of integers in $[0, a_1) \times [0, a_2) \times [0, a_3)$ to integers in $[0, a_1 \cdot a_2 \cdot a_3)$:
  - (i, j, k) maps to $i \cdot a_2 \cdot a_3 + j \cdot a_3 \cdot a_1 + k \cdot a_1 \cdot a_2 \ mod \ a_1 \cdot a_2 \cdot a_3$.

Main Result: Part 1

Theorem

If intersection non-emptiness for three unary DFA's $\in DTIME(n^{\frac{\alpha}{2}})$, then Triangle Finding $\in DTIME(n^{\alpha})$.

- The integers i, j, and k each represent a choice of a vertex from G when i, j, and $k \in [0, n)$, respectively.
- If an input encodes a triple of integers (i, j, k), then
  - D_1 accepts when (i, j) represents an edge in G
  - D_2 accepts when (j, k) represents an edge in G
  - D_3 accepts when (k, i) represents an edge in G
- Together the DFA's collectively determine whether or not an input encodes a triangle.

Main Result: Part 2

Theorem

If Triangle Finding $\in DTIME(n^{\alpha})$, then intersection non-emptiness for three unary DFA's $\in DTIME(n^{\frac{\alpha}{2}})$.

- Let unary DFA's D_1, D_2, and D_3 each with at most n states be given.
- We first consider final states on the segments of the DFA's.
- We can quickly check to see if any of these finals states correspond with a string in the intersection.

Main Result: Part 2

Theorem

If Triangle Finding $\in DTIME(n^{\alpha})$, then intersection non-emptiness for three unary DFA's $\in DTIME(n^{\frac{\alpha}{2}})$.

- Next, we consider final states on the cycles of the DFA's.
- Each final state has an associated natural number constraint that checks the remainder modulo the cycle length.
- Let c_1, c_2, and c_3 denote the cycle lengths of D_1, D_2, and D_3, respectively.
- Let d denote the greatest common divisor of c_1, c_2, and c_3.
- Further, let $c_1' = \frac{c_1}{d}$, $c_2' = \frac{c_2}{d}$, and $c_3' = \frac{c_3}{d}$.

Main Result: Part 2

Theorem

If Triangle Finding $\in DTIME(n^{\alpha})$, then intersection non-emptiness for three unary DFA's $\in DTIME(n^{\frac{\alpha}{2}})$.

- For each integer $i \in [0, d)$, we construct a tripartite graph G_i.
- We construct the G_i's so that the DFA's have a non-empty intersection if and only if there exists $i \in [0, d)$ such that G_i has a triangle.
- Let $i \in [0, d)$ be given. The graph G_i has three groups of vertices such that:
    - the first group has g_1 vertices where $g_1 = gcd(c'_1, c'_2)$
    - the second group has g_2 vertices where $g_2 = gcd(c'_2, c'_3)$
    - the third group has g_3 vertices where $g_3 = gcd(c'_3, c'_1)$

Main Result: Part 2

Theorem

If Triangle Finding $\in DTIME(n^{\alpha})$, then intersection non-emptiness for three unary DFA's $\in DTIME(n^{\frac{\alpha}{2}})$.

- For each $j \in \{1, 2, 3\}$, the vertices in the jth group are labeled with integers from 0 to $g_j - 1$.
- Each edge is associated with a final state from the DFA's.
- There is an edge between the vertex labeled x from the first group and the vertex labeled y from the second group if there is a final state from D_2 whose natural number constraint's remainder is congruent to i mod d, x mod g_1, and y mod g_2.
- Edges are defined similarly between other groups of vertices.

Main Result: Part 2

Theorem

If Triangle Finding $\in DTIME(n^{\alpha})$, then intersection non-emptiness for three unary DFA's $\in DTIME(n^{\frac{\alpha}{2}})$.

- At this point, it can be seen that a triangle of G_i is exactly a choice of finals states f_1, f_2, and f_3 from D_1, D_2, and D_3, respectively, such that:
  - there exists a string that leads to all three finals states
  - each final state's associated remainder is congruent to i mod d

Main Result: Part 2

Theorem

If Triangle Finding $\in DTIME(n^{\alpha})$, then intersection non-emptiness for three unary DFA's $\in DTIME(n^{\frac{\alpha}{2}})$.

- The groups of vertices may be lopsided in the sense that one group may contain more vertices than another.
- We can reduce triangle finding for G_i to triangle finding for a collection of tripartite subgraphs with an equal number of vertices in each group.
- Finally, we just need to solve Triangle Finding on each of these subgraphs.

Main Result: Part 2

Theorem

If Triangle Finding $\in DTIME(n^{\alpha})$, then intersection non-emptiness for three unary DFA's $\in DTIME(n^{\frac{\alpha}{2}})$.

- Based on the definitions of g_1, g_2, and g_3, we have the following inequalities:
  - $g_1 \cdot g_2 \leq c_2'$
  - $g_2 \cdot g_3 \leq c_3'$
  - $g_3 \cdot g_1 \leq c_1'$
- Without loss of generality, the second group is the smallest meaning that $g_2 = min(g_1, g_2, g_3)$.

Main Result: Part 2

Theorem

If Triangle Finding $\in DTIME(n^{\alpha})$, then intersection non-emptiness for three unary DFA's $\in DTIME(n^{\frac{\alpha}{2}})$.

- Therefore, when we rebalance, we can split into roughly $\frac{g_1 \cdot g_3}{g_2^2}$ subgraphs each with $O(g_2)$ states.
- Finally, we get that solving Triangle Finding for all of the balanced subgraphs across all of the G_i's takes time $O(d \cdot g_1 \cdot g_3 \cdot g_2^{\alpha-2}) \leq O(d \cdot {c'_1}^{\frac{\alpha}{2}}) \leq O(d \cdot (\frac{n}{d})^{\frac{\alpha}{2}})$.

Summary of Reduction

- The reduction can be broken into two phases.
- **Phase 1:** We reduce one instance of intersection non-emptiness for three unary DFA's to d instances of triangle finding where each graph has $g_1 + g_2 + g_3$ vertices.
- **Phase 2:** We further reduce each of these instances of triangle finding to $\frac{g_1 \cdot g_3}{g_2^2}$ instances of triangle finding where each graph has $O(g_2)$ vertices.