A Satisfiability Algorithm for Synchronous Boolean Circuits

SUMMARY The circuit satisfiability problem has been intensively studied since Ryan Williams showed a connection between the problem and lower bounds for circuit complexity. In this letter, we present a \#SAT algorithm for synchronous Boolean circuits of \( n \) inputs and \( s \) gates in time \( 2^{n(1-\frac{1}{\log n})} \) if \( s = o(n \log n) \).

key words: circuit satisfiability problem, synchronous circuit, exact exponential algorithm, circuit complexity

1. Introduction

The circuit satisfiability problem is defined as follows.

Input: A Boolean circuit \( C \)

Output: Does \( C \) output 1 on some input?

The circuit satisfiability problem has been intensively studied since Ryan Williams showed a connection between the problem and lower bounds for circuit complexity [1]. In this letter, we consider the circuit satisfiability problem for synchronous circuits. Synchronous circuits have appeared in the well-known book by Wegener (Chapter 12.1 of [2]).

The result of this letter is the following one theorem. The size of a circuit is the number of gates in the circuit. \#SAT is the counting version of a satisfiability problem (SAT), and obviously harder than SAT.

Theorem 1. The \#SAT problem for synchronous circuits of \( n \) inputs and size \( s \) can be solved in time \( 2^{n(1-\frac{1}{\log n})} \) if \( s = o(n \log n) \).

Thus, if the size of a synchronous circuit is \( o(n \log n) \), then the running time is \( 2^{n(1-\frac{1}{\log n})} \). If the size is \( O(n) \), then the running time is \( 2^{n(1-\Theta(1))} \).

Any (non-synchronous) circuit of size \( O(n) \) and depth \( O(\log n) \) can be easily converted to a synchronous circuit of size \( O(n \log n) \). Therefore, if our algorithm is slightly improved on the size, then we obtain a \#SAT algorithm for bounded depth circuits of size \( O(n) \) and depth \( O(\log n) \).

2. Preliminaries

Circuits are formally defined as directed acyclic graphs. The nodes of in-degree 0 are called inputs, and each one of them is labeled by a variable. The other nodes are called gates, and each one of them is labeled by a Boolean function. In this letter, the gates are AND gates of fan-in two, OR gates of fan-in two, and NOT gates. There is a single specific node called output. The size of a circuit is the number of gates in the circuit.

A synchronous circuit is a circuit with the additional property that all paths from the inputs to some gate have the same length.

3. The Key Lemma

In this section, we prove the key lemma, which shows a property of synchronous circuits. Once the lemma is proved, our algorithm is almost straightforward as described in the next section.

Lemma 2. For any synchronous circuit of size \( s = o(n \log n) \), there exist \( n - \frac{n}{2^{\log n}} \) inputs such that the number of circuits after all assignments to the inputs is at most \( 2^s \), and such inputs can be found in polynomial time of \( s \).

Proof. A \( k \)-th layer is gates whose path from the inputs to the gate has length \( k \). There exists the \( k' \)-th layer such that \( k' \leq \frac{4s}{n} \) and the number of gates in the layer is at most \( \frac{n}{4} \) since the circuit is a synchronous circuit of size \( s \). We focus on the \( k' \)-th layer.

We separate the inputs of the circuit to \( X_1 \) and \( X_2 \). Each gate of the \( k' \)-th layer is connected from at most \( 2^{\frac{s}{n}} \) inputs. Thus, the sum of the number of gates connected from each input in the \( k' \)-th layer is at most \( \frac{n}{4} \cdot 2^{s} \). We can select \( \frac{n}{4} \) inputs as \( X_1 \) such that the number of gates connected from inputs in \( X_1 \) is at most \( \left( \frac{n}{4} \cdot 2^{s} \right) \cdot \left( \frac{n}{2^{\log n}} \right) = \frac{n}{2^{\log n}} \). Let \( X_2 \) be the inputs which are not selected as \( X_1 \).

\( X_2 \) is the inputs which satisfy Lemma 2. Let \( G_1 \) be the gates which are connected from some input in \( X_1 \) and contained in the \( k' \)-th layer. The number of gates in \( G_1 \) is at most \( \frac{n}{2^{\log n}} \). Let \( G_2 \) be the other gates in the \( k' \)-th layer. After an assignment to \( X_2 \), the outputs of the gates in \( G_2 \) are fixed, and the number of gates are at most \( \frac{n}{4} \). The number of inputs which are contained in \( X_2 \) and connect to some gate in \( G_1 \) is at most \( \frac{n}{2^{\log n}} \cdot 2^{s} = \frac{n}{2^{\log n}} \). Thus, the number of circuits after all assignments to \( X_2 \) is at most \( 2^{s} \cdot \frac{n}{2^{\log n}} \leq 2^{s} \).

Obviously, we can find such inputs in polynomial time of \( s \).
4. A Satisfiability Algorithm

In this section, we describe our algorithm, and prove Theorem 1.

Proof of Theorem 1. Our algorithm consists of three steps.

1. We find \( n - \frac{n}{2^{\Omega(n)}} \) inputs which satisfy Lemma 2.
2. We assign all patterns to the \( n - \frac{n}{2^{\Omega(n)}} \) inputs, and memorize the each number of remained circuits.
3. We execute the brute-force search for each circuit of \( n \frac{n}{2^{\Omega(n)}} \) inputs.

The algorithm obviously runs in time \( 2^n(1 - \frac{1}{2^{\Omega(n)}}) \). \( \square \)

References

[1] R. Williams, “Algorithms for circuits and circuits for algorithms,” Proc. CCC, pp.248–261, 2014.
[2] I. Wegener, The Complexity of Boolean Functions, Wiley-Teubner, 1987.