The Minimum Circuit Size Problem

John Gouwar   Caleb Robelle

DIMACS REU Program

June 1, 2020

Overview

Boolean Functions

A boolean function on n variables is a function $f : \{0,1\}^n \to \{0,1\}$ that can be encoded as a truth table of size 2^n.

Boolean Functions

A boolean function on n variables is a function $f : \{0,1\}^n \to \{0,1\}$ that can be encoded as a truth table of size 2^n.

Example of encoding of function on 4 variables:

n	f(n)
0000	1
0001	0
0010	1
0011	0
$\vdots$	
1110	1
1111	0

In this case, $\langle f \rangle = 010 \ldots 010$ and $|\langle f \rangle| = 2^4$

Boolean Circuits

Any boolean function can be modeled by a sequence of boolean (AND, OR, and NOT) gates.

Boolean Circuits

Any boolean function can be modeled by a sequence of boolean (AND, OR, and NOT) gates.

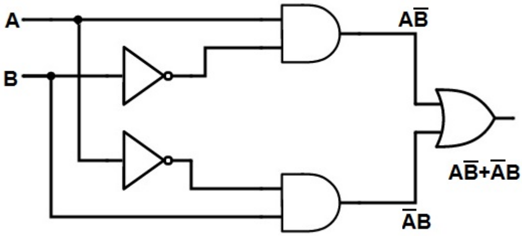

Figure: Example for XOR

We consider the size of a circuit c, denoted $SIZE(c)$, to be the number of gates in the circuit.

The Minimum Circuit Size Problem

Problem

Given a boolean function f and some natural number s, MCSP asks the question: Can f be computed by a circuit of size at most s? More formally,

$$\mathrm{MCSP} = \{\langle f, s\rangle \mid f \textit{ is computed by a circuit } c \textit{ such that } SIZE(c) \leq s\}$$

The Minimum Circuit Size Problem

Problem

Given a boolean function f and some natural number s, MCSP asks the question: Can f be computed by a circuit of size at most s? More formally,

$$\mathrm{MCSP} = \{\langle f, s\rangle \mid f \textit{ is computed by a circuit } c \textit{ such that } SIZE(c) \leq s\}$$

Example

For all $s \geq 5$, $\langle XOR, s\rangle \in \mathrm{MCSP}$

Many-one reductions

Definition

Given problems A and B, and a complexity class $\mathcal{C}$, we say that $A \leq_m^{\mathcal{C}} B$ if there exists a $\mathcal{C}$-computable function f such that $f(x) \in B$ if and only if $x \in A$.

Many-one reductions

Definition

Given problems A and B, and a complexity class $\mathcal{C}$, we say that $A \leq_m^{\mathcal{C}} B$ if there exists a $\mathcal{C}$-computable function f such that $f(x) \in B$ if and only if $x \in A$.

Example

Consider the following languages:

- $L_1 = \{x \in \{0,1\}^* | \, x \text{ is composed of alternating 1's and 0's}\}$
- $L_2 = \{x \in \{0,1\}^* | \, x \text{ is composed of alternating pairs of 1's and 0's}\}$

Hardness and Completeness

Definitions

A problem A is **hard** for a complexity class $\mathcal{C}$ if for every $B \in \mathcal{C}$, there exists a reduction from B to A. A problem A is **complete** for a class $\mathcal{C}$ if A is both hard for $\mathcal{C}$ and $A \in \mathcal{C}$.

Hardness and Completeness

Definitions

A problem A is **hard** for a complexity class $\mathcal{C}$ if for every $B \in \mathcal{C}$, there exists a reduction from B to A. A problem A is **complete** for a class $\mathcal{C}$ if A is both hard for $\mathcal{C}$ and $A \in \mathcal{C}$.

Example

[Your favorite choice of NP-complete problem] is complete for NP under many-one polynomial time reductions.

Statistical Zero Knowledge (SZK)

Definition

SZK is the class of problems which have interactive statistical zero knowledge proofs as solutions. These are proofs in which a prover and a verifier interact in such a way that the verifier is certain that the prover knows the secret, but does not give the secret away directly.

Statistical Zero Knowledge (SZK)

Definition

SZK is the class of problems which have interactive statistical zero knowledge proofs as solutions. These are proofs in which a prover and a verifier interact in such a way that the verifier is certain that the prover knows the secret, but does not give the secret away directly.

Example

Proving knowledge of the difference between Coke and Pepsi:

- Prover claims to know the difference between Coke and Pepsi

Statistical Zero Knowledge (SZK)

Definition

SZK is the class of problems which have interactive statistical zero knowledge proofs as solutions. These are proofs in which a prover and a verifier interact in such a way that the verifier is certain that the prover knows the secret, but does not give the secret away directly.

Example

Proving knowledge of the difference between Coke and Pepsi:

- Prover claims to know the difference between Coke and Pepsi
- Verifier flips a coin fifty times and gives the prover Pepsi if heads and Coke if tails to check if the prover has the knowledge

Statistical Zero Knowledge (SZK)

Definition

SZK is the class of problems which have interactive statistical zero knowledge proofs as solutions. These are proofs in which a prover and a verifier interact in such a way that the verifier is certain that the prover knows the secret, but does not give the secret away directly.

Example

Proving knowledge of the difference between Coke and Pepsi:

- Prover claims to know the difference between Coke and Pepsi
- Verifier flips a coin fifty times and gives the prover Pepsi if heads and Coke if tails to check if the prover has the knowledge
- If the prover knows the difference, they should pick the right beverage every time.

Statistical Zero Knowledge (SZK)

Definition

SZK is the class of problems which have interactive statistical zero knowledge proofs as solutions. These are proofs in which a prover and a verifier interact in such a way that the verifier is certain that the prover knows the secret, but does not give the secret away directly.

Example

Proving knowledge of the difference between Coke and Pepsi:

- Prover claims to know the difference between Coke and Pepsi
- Verifier flips a coin fifty times and gives the prover Pepsi if heads and Coke if tails to check if the prover has the knowledge
- If the prover knows the difference, they should pick the right beverage every time.
- If the prover has no knowledge, they only have a $\frac{1}{2^{50}}$ of getting it right every time.

Non-Interactive Statistical Zero Knowledge (NISZK)

Definition

The complexity class NISZK is a subset of SZK where the proof system can be defined non-interactively. In these systems, communication only comes from the prover, but both the prover and the verifier have access to a random string.

Goal

Our goal is to show that a problem related to MCSP, known as MKTP, is hard for the class NISZK_L under $\leq_m^{\mathrm{NC}^0}$ reductions.

Acknowledgements

We would like to thank the DIMACS REU program for giving us the opportunity with funding through the through NSF grant CCF-1836666.

We would also like to thank our adviser, Dr. Eric Allender, for his support and mentorship.

Logic Gate Source:
https://www.electronicshub.org/exclusive-or-gatexor-gate/