Boolean Circuit

Xin Li

lixints@cs.jhu.edu
Topics

• *The circuit model*

• *Circuit size*

• *Circuit satisfiability*

• *Size Hierarchy*

• *Karp-Lipton Theorem*
Boolean Circuit

• A different computational model.

• Generalizing Boolean formulas and a simplified model of chips.

• A Boolean circuit is a diagram to compute an output from a binary input string.

• Computation: applying a sequence of basic Boolean operations: AND, OR, NOT.
**Boolean Circuit**

- For any $n$, an $n$-input 1-output Boolean circuit is a DAG with $n$ sources and one sink.

- All non-source vertices are called gates and labeled with one of $\lor$, $\land$, $\neg$.

- The size of the circuit $C$, $|C|$ is the # of vertices in it.
Boolean Circuit

- The # of incoming edges to a gate is called fan-in.

- For $\lor$, $\land$, fan-in=2; for $\neg$, fan-in=1

- The # of outgoing edges of a gate is called fan-out.

- Boolean formulas: fan-out =1; General circuits: fan-out can be larger than 1.
Boolean Circuit

• Boolean circuit is a **non-uniform** computational model.

• TM model: the same TM is used for all possible input sizes.

• Boolean circuit: can have a different circuit for every input size.

• Thus, even if $P \neq NP$, it is possible to have a small size (e.g., poly size) circuit tailored to solve SAT.
Boolean Circuit

• Definition: Let $T: \mathbb{N} \rightarrow \mathbb{N}$ be a function.

• A $T(n)$ size circuit family is a sequence $\{C_n, n \in \mathbb{N}\}$ of Boolean circuits, where $C_n$ has $n$ inputs and 1 output, and $|C_n| \leq T(n)$ for every $n$.

• Say a language $L \in \text{SIZE}(T(n))$ if $\exists$ a $T(n)$ size circuit family $\{C_n, n \in \mathbb{N}\}$ such that
  \[ \forall \, x \in \{0, 1\}^n, \, x \in L \iff C_n(x) = 1 \]
Boolean Circuit

• A CNF is a special circuit, so any Boolean function on $n$ bits can be computed by a Boolean circuit of size at most $n2^n$.

• Include even uncomputable functions (by TMs).

• More interesting: small size circuits.

• Definition: $P_{/poly}$ is the class of languages decidable by poly-size circuits.

$$P_{/poly} = \bigcup_{c \geq 0} \text{Size}(n^c)$$
Poly-Size Boolean Circuit

• Definition: $P_{\text{poly}}$ is the class of languages decidable by poly-size circuits.

$$P_{\text{poly}} = \bigcup_{c \geq 0} \text{Size}(n^c)$$

• Theorem: $P \subseteq P_{\text{poly}}$.

• Proof: similar to the proof of Cook-Levin Theorem.
Poly-Size Boolean Circuit

• Theorem: $P \subseteq P_{poly}$.

• Proof: if $L$ is decided by a TM running in time $T(n)$, then consider the $T(n)+1$ configurations $C_0, \ldots, C_{T(n)}$.

• Each one can be computed from the previous one by using an $O(T(n))$ size CNF, and the output can be computed from $C_{T(n)}$.

• This gives a circuit of size $O(T(n)^2)$. 
Poly-Size Boolean Circuit

• Theorem: $P \subseteq P_{\text{poly}}$.

• Note that the depth of the circuit is also $O(T(n))$.

• Also, the circuit is not only of poly-size, but itself can be computed by a TM in poly time or even log space.

• Such circuits are called P-uniform circuits (restricting $P_{\text{poly}}$ to be P-uniform collapses it to $P$, why?).
Poly-Size Boolean Circuit

• **Theorem:** $P \subseteq P_{/\text{poly}}$.

• The relation is proper, i.e., $P \subset P_{/\text{poly}}$. For example, $P_{/\text{poly}}$ consists of undecidable languages.

• **Claim:** Let $L \subseteq \{0, 1\}^*$ be any unary language, i.e., $L \subseteq \{1^n, n \in \mathbb{N}\}$. Then $L \in P_{/\text{poly}}$.

• **Proof:** for every $n \in \mathbb{N}$, construct a different circuit $C_n$. If $1^n \in L$, then $C_n$ is the AND of all input bits; otherwise $C_n$ just outputs 0.
Poly-Size Boolean Circuit

• Any unary language is in $P/poly$.

• Can take any undecidable language, and encode it as a unary language.

• Example: $UHALT=\{1^n, \text{the binary expression of } n \text{ encodes a pair } <M, x> \text{ such that } TM \text{ M halts on } x.\}$

• $HALT \leq_m UHALT$ by the computable mapping $<M, x> \rightarrow 1^n, n$ is the binary expression of $<M, x>$. 
Circuit Satisfiability

• Boolean circuits can be used to give another proof of Cook-Levin.

• Definition: The language $\text{CKT-SAT} = \{\text{all circuits (string representation) that are satisfiable}\}$

• Theorem: $\text{CKT-SAT}$ is NP-complete.

• Clearly $\text{CKT-SAT} \in \text{NP}$ (computing the output given the description of a circuit can be done in poly time).
Circuit Satisfiability

• Show that $\forall L \in NP$, $L \leq_p \text{CKT-SAT}$.

• $L \in NP => \exists$ a polynomial function $P: N \rightarrow N$ and a polynomial time TM $M$ s.t. $\forall x \in \{0, 1\}^*$, $x \in L$ iff $\exists u \in \{0, 1\}^{P(l_x)}$ s.t. $M(x, u)=1$.

• $\exists u \in \{0, 1\}^{P(l_x)}$ s.t. $M(x, u)=1$ $\iff$ $C_x(u)=1$ where $C_x$ is the circuit computing $M(x, \ )$.

• Thus $x \rightarrow C_x$ is a poly-time reduction (computing $C_x$ in poly-time is shown before).
Circuit Satisfiability

• CKT-SAT is NP-complete.

• Lemma: CKT-SAT $\leq_p$ SAT.

• For every gate $V_i$ of $C_n$, introduce a variable $Z_i$ and add a corresponding CNF.

• Example: if $V_i = V_j \land V_k$, add $Z_i = Z_j \land Z_k$ and express it as a CNF. Finally, add the output variable (i.e., let the output=1).
Circuit Size for a Function

- Any Boolean function on \( n \) bits can be computed by a Boolean circuit (CNF) of size at most \( n2^n \).

- Can we do better?

- Theorem: \( \exists \) a Boolean function on \( n \) bits (in fact, most such functions) that requires circuit size \( \Omega(2^n/n) \).