Nonuniform Linear Depth Circuits Decide Everything

Luowen Qian*

Abstract

In this work, we introduce the nonuniform complexity class LD, which stands for linear depth circuit. We also prove that $\mathsf{LD} = \mathsf{ALL}$, which is the complexity class that contains all decision languages, and therefore resolving all questions on this new complexity class. No further research is needed [Mun20].

In principle, this shows that *anything* can be computed via circuits in linear time, despite with (possibly undecidable) pre-computation and very inefficient advice, however, we note that exponential sized advice suffices to achieve ALL.

Definition 1. *LD is the nonuniform complexity class such that for any language $L \in \mathsf{LD}$, there exists a family of unbounded-size fan-in-2 Boolean circuit[1] family $\{C_n\}_{n\in\mathbb{N}}$, such that:*

1. *The depth of circuit C_n is $O(n)$;*
2. *For any x, $x \in L$ if and only if $C_{|x|}(x) = 1$.*

Lemma 2. *For any $n \in \mathbb{N}$, we can generate all possible truth tables using a $(2n+1)$-depth circuit, namely, for any truth table $(\{0,1\}^n \mapsto \{0,1\}) \in \{0,1\}^{2^n}$, we can find a wire in the circuit that corresponds to this truth table.*

Proof. We prove this by induction.

For $n = 0$, 2 constant gates (which is of depth 1) suffice to generate all 2 possible truth tables.

Now, assume that this holds for $n-1$, let C_{n-1} be the $(2n-1)$-depth circuit that generates all truth tables of $(n-1)$ inputs. Given a truth table $f(x_1, ..., x_n)$, we can implement it by adding the gates: $(x_n = 1 \wedge f(x_1, ..., x_{n-1}, 1)) \vee (x_n = 0 \wedge f(x_1, ..., x_{n-1}, 0))$. Doing this for every truth table, we obtain a circuit of depth $(2n+1)$, as each $f(x_1, ..., x_{n-1}, c)$ is implemented by some wire at depth at most $(2n-1)$. □

Theorem 3. *$\mathsf{ALL} \subseteq \mathsf{LD}$.*

Proof. Given any language $L \in \mathsf{ALL}$, let $L_n = \{x : x \in L \wedge |x| = n\}$. By the lemma above, we know we can decide L_n using a circuit of depth $(2n+1)$ for every $n \in \mathbb{N}$. This gives us a circuit family for L. □

Corollary 4. *$\mathsf{LD} = \mathsf{ALL}$.*

Note that the circuit given by Lemma 2 is explicit, and there are exactly 2^{2^n} wires in depth $(2n+1)$, so to specify which truth table to use for the language, 2^n bits of advice suffices.

*Boston University. Email: `luowenq@bu.edu`

[1] Without loss of generality, we assume that each gate computes an arbitrary function $\{0,1\}^2 \mapsto \{0,1\}$.

Acknowledgements

The authors would like to thank Qipeng Liu for pointing out that to implement a circuit in LD, in particular, the one we construct in this work, either it still takes super-polynomial time to evaluate the circuit, or general relativity is false[2,3]. However, we would like to claim that this point is only of interest to physicists, and any complexity theorist should ignore this physical constraint and treat general relativity as one of the many incomplete (and thus incorrect) theories.

References

[Mun20] Randall Munroe. Further research is needed. `https://xkcd.com/2268/`, 2020. Accessed: 2020-03-31.

[2]One of the ways it is false could be that we are actually living in an exponential-dimensional *physical* space. In general, a polynomial-time agent can only access polynomially many bits in three-dimensional world, assuming general relativity and quantum mechanics.

[3]The moral of this story is that the circuit model and RAM model actually hide some non-local computation inside. Extreme care should be taken when one inspects super-polynomial hardness under non-Turing-machine models.