Majority is in NC^1 (lecture notes)

Course: Circuit Complexity, Autumn 2024, University of Chicago
Instructor: William Hoza (`williamhoza@uchicago.edu`)

---

1 Shallow circuit models

Definition 1 (Circuit depth)**.** The *depth* of a circuit is the length of the longest directed path in the underlying graph.

Definition 2 (The NC hierarchy)**.** Let i be a nonnegative integer. A function $f\colon \{0,1\}^* \to \{0,1\}^*$ is in NC^i if, for every n, there is a circuit of depth $O(\log^i n)$ and size $\mathrm{poly}(n)$ (with bounded fan-in) that computes f restricted to inputs of length n. We also define $\mathsf{NC} = \bigcup_i \mathsf{NC}^i$.

Definition 3 (AC circuits)**.** An AC *circuit* is a circuit of the following type:

- The gates are arranged in alternating layers of AND gates and OR gates.
- The gates have unbounded fan-in.
- At the bottom, there are constants, variables, and negated variables. Negations do not count toward the size or depth of the circuit.

Definition 4 (The AC hierarchy)**.** Let i be a nonnegative integer. A function $f\colon \{0,1\}^* \to \{0,1\}^*$ is in AC^i if, for every n, there is an AC circuit of depth $O(\log^i n)$ and size $\mathrm{poly}(n)$ that computes f restricted to inputs of length n.

It is common to abuse notation by referring to AC circuits as "AC^0 circuits," especially when the depth is $o(\log n)$. The notation AC^i, NC^i, etc. is often reserved for languages, i.e., functions outputting a single bit. We have

$$\mathsf{NC}^0 \subseteq \mathsf{AC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{AC}^1 \subseteq \mathsf{NC}^2 \subseteq \cdots \subseteq \mathsf{NC} \subseteq \mathsf{P/poly}.$$

The notation AC^i, NC^i, etc. is also sometimes used to refer to *uniform* versions of these complexity classes. For example, you might see statements such as $\mathsf{NC} \subseteq \mathsf{P}$.

2 Shallow circuits for addition and majority

Theorem 1. $\mathsf{ADD}_{2\times n} \in \mathsf{AC}^0$.

Note: Strictly speaking, it doesn't make sense to say that $\mathsf{ADD}_{2\times n}$ is in AC^0, because $\mathsf{ADD}_{2\times n}$ has a finite domain. What we mean is that the *infinite family* of functions $\mathsf{ADD}_{2\times 1}, \mathsf{ADD}_{2\times 2}, \mathsf{ADD}_{2\times 3}, \ldots$, viewed as a single function on $\{0,1\}^*$, is in AC^0. This is a common and convenient abuse of notation.

Proof sketch. Say we are trying to compute $z = x + y$ where $x, y \in \{0, 1, \ldots, 2^n - 1\}$. Recall the notion of *carry bits* from the standard grade-school addition algorithm. Let c_i be the carry bit at position i, where "position 0" refers to the least significant bit. Then

$$c_i = \bigvee_{j \le i} \left(x_j \wedge y_j \wedge \bigwedge_{j<k<i} (x_k \vee y_k) \right)$$

(an AC^0 circuit). Furthermore, $z_i = x_i + y_i + c_i \bmod 2$ (an NC^0 circuit). Thus, $\mathsf{ADD}_{2\times n} \in \mathsf{NC}^0 \circ \mathsf{AC}^0 = \mathsf{AC}^0$. $\square$

Is it possible to improve Theorem 1 to get an NC^0 circuit? Strictly speaking, the answer is no:

Proposition 1. $\mathsf{ADD}_{2\times n} \notin \mathsf{NC}^0$.

Proof sketch. In an NC^0 circuit, each output bit depends on only $O(1)$ input bits. In contrast, the most significant bit of $x + y$ depends on all the bits of x and y. (Think about the case that $x = 2^n - 1$ and y is a power of two, or vice versa.) □

However, it is possible in NC^0 to do something called "three-to-two addition," which is almost as good as actual addition.

Lemma 1 (Three-to-Two Addition)**.** *For every $n \in \mathbb{N}$, there is a function $C\colon (\{0,1\}^n)^3 \to (\{0,1\}^{n+1})^2$ such that $C \in \mathsf{NC}^0$, and for every $x, y, z \in \{0, 1, \ldots, 2^n - 1\}$, the circuit C computes integers $C(x, y, z) = (u, v)$ satisfying $u + v = x + y + z$.*

Proof sketch. Let $v_{i+1}u_i = \mathsf{ADD}_{3\times 1}(x_i, y_i, z_i)$. □

Corollary 1. $\mathsf{ADD}_{n\times n} \in \mathsf{NC}^1$.

Proof sketch. A layer of three-to-two addition circuits reduces the number of summands from n down to $2n/3$, while increasing the bit-length of the summands by one. After $O(\log n)$ layers of three-to-two addition circuits, we have just two summands, each with bit-length $n + O(\log n)$. Then we can apply Theorem 1. Thus, $\mathsf{ADD}_{n\times n} \in \mathsf{AC}^0 \circ \mathsf{NC}^1 = \mathsf{NC}^1$. □

Corollary 2. $\mathsf{MAJ}_n \in \mathsf{NC}^1$.

Corollary 3 (Adleman's theorem for NC^1)**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$. Suppose f can be computed by a "randomized NC^1 circuit," i.e., there is a circuit $C\colon \{0,1\}^n \times \{0,1\}^r \to \{0,1\}$ with bounded fan-in and depth $O(\log n)$ such that for every $x \in \{0,1\}^n$, we have*

$$\Pr_{y\in\{0,1\}^r}[C(x, y) = f(x)] \geq 2/3.$$

Then $f \in \mathsf{NC}^1$.

Proof sketch. Mimic the standard proof of Adleman's theorem, and use the fact that $\mathsf{MAJ}_n \in \mathsf{NC}^1$. □

Note that the circuit constructed in Corollary 3 is nonuniform, just like the standard version of Adleman's theorem. Because of Corollary 3, if you ever encounter a complexity class with a name like "RNC" or "BPNC," it probably refers to functions computable by *uniform* randomized NC circuits.