Circuit Complexity

Andy Yang

Previously

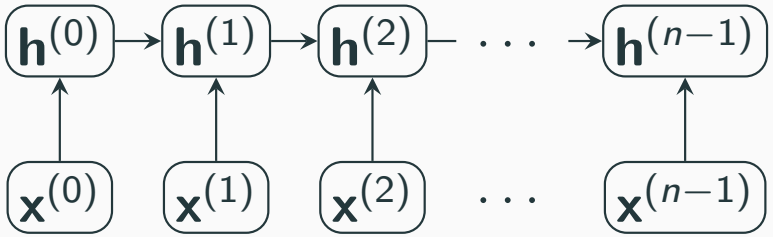


Figure 1: A simple RNN.

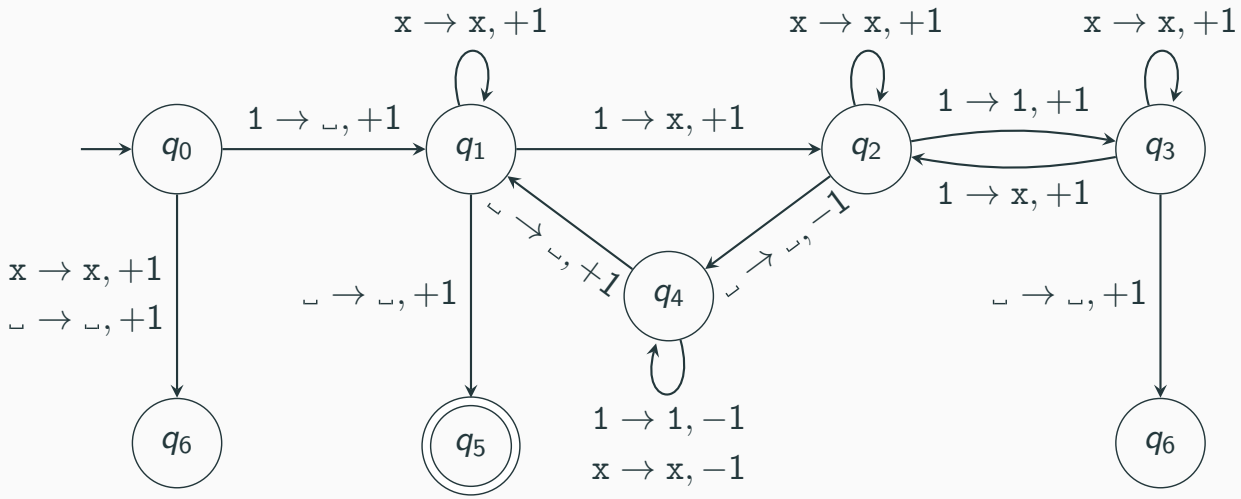


Figure 2: A Turing Machine for $\{1^{2^m} \mid m \geq 0\}$

Why are RNNs less popular today?

RNNs are not easily parallelizable during training

Parallelism

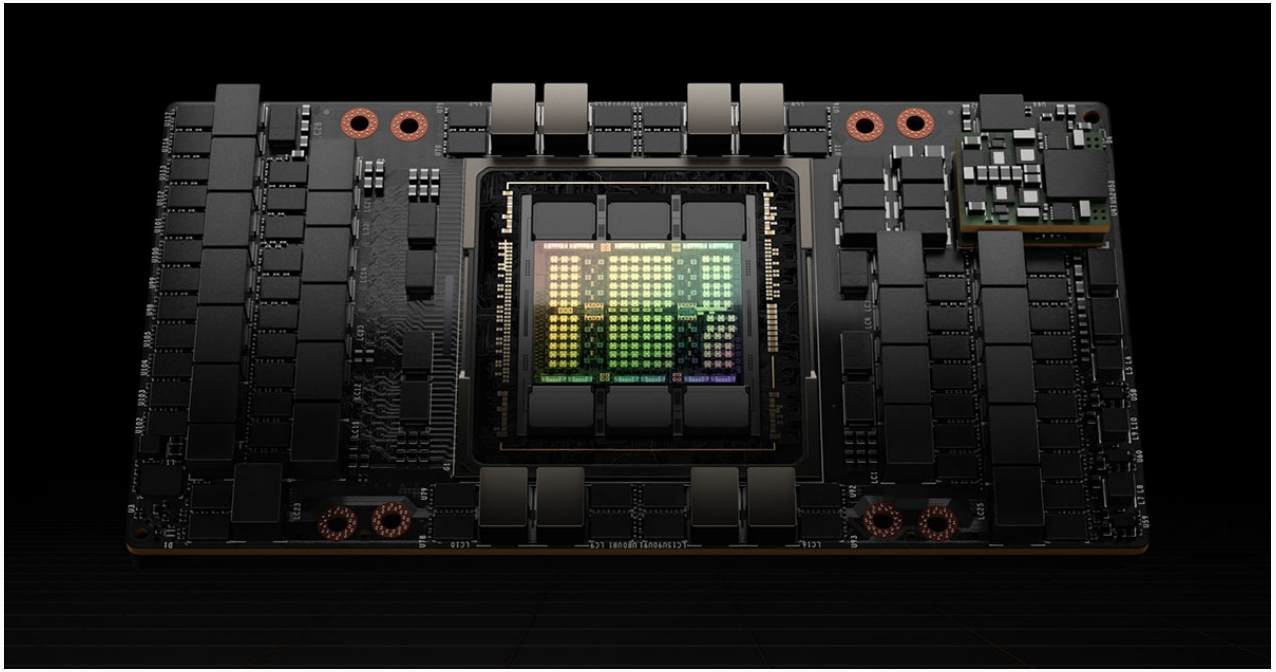
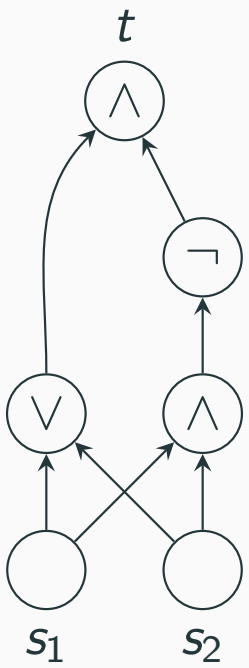


Figure 3: A GPU

Example

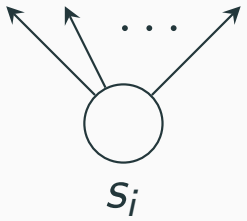
Here's a circuit with input length 2. It computes the XOR function. We draw the inputs at the bottom and the output at the top.



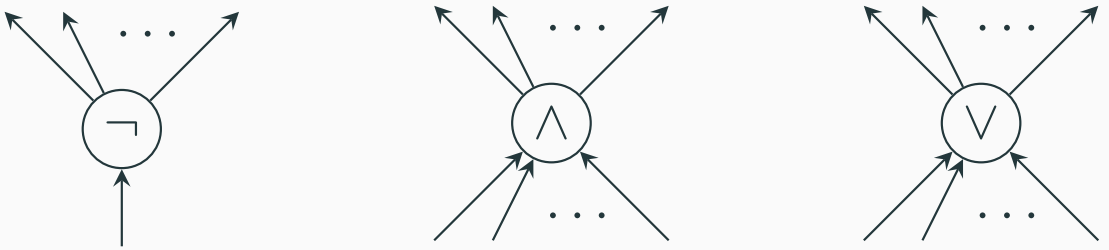
Circuits (Formally)

A *(Boolean) circuit* C with input length n is a directed acyclic graph with

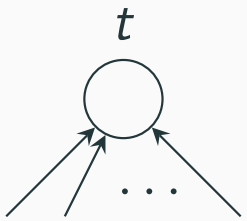
1. n nodes $s_0, \ldots, s_{n-1}$ with zero fan-in, designated as *input* nodes:



2. zero or more fan-in (in-degree) *gate* nodes, each labeled with a function:



3. A node t, which can be either an input or gate node, is designated the *output* of the circuit.



Accepting Strings

Given an input string $\mathbf{w} \in \{0,1\}^n$, each input node s_i is assigned the value w_i, and each gate node labeled f computes its value by applying f to the values of its in-neighbors. Thus, we can think of the circuit as computing a function $C : \{0,1\}^* \to \{0,1\}$, mapping each input string $\mathbf{w}$ to the value of t.

Notions of Complexity

- **Depth**: Longest path from input to output
- **Size**: Total number of nodes
- **Fan-in**: Number of wires entering each node

Bounded vs Unbounded Input Size

- In many applications, inputs are of bounded size
- We care about natural language processing, potentially unbounded lengths

Circuit Families

Definition (Boolean circuit families)
A *circuit family* is a sequence $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ such that for each n, C_n is a circuit with input length n. We treat $\mathcal{C}$ as a function on $\{0,1\}^*$ as follows.

For every $\mathbf{w} \in \{0,1\}^*$ with length n, $\mathcal{C}(\mathbf{w}) = C_n(\mathbf{w})$. Then the language defined by $\mathcal{C}$ is $L(\mathcal{C}) = \{\mathbf{w} \in \{0,1\}^* \mid \mathcal{C}(\mathbf{w}) = 1\}$. The *depth* and *size* of $\mathcal{C}$ are the functions $n \mapsto \text{depth}(C_n)$ and $n \mapsto |C_n|$.

Important Circuit Families

Definition (AC^k, TC^k, and NC^k)

We define the following classes of languages:

- AC^k is the class of languages that can be recognized by families of circuits with unbounded fan-in, $O(\text{poly}(n))$ size, and $O((\log n)^k)$ depth.
- TC^k is like AC^k, but also allows MAJORITY gates, which have unbounded fan-in and output 1 iff at least half of their inputs are 1.
- NC^k is the class of languages that can be recognized by families of circuits with fan-in at most 2, $O(\text{poly}(n))$ size, and $O((\log n)^k)$ depth.

Binary Addition in AC^0

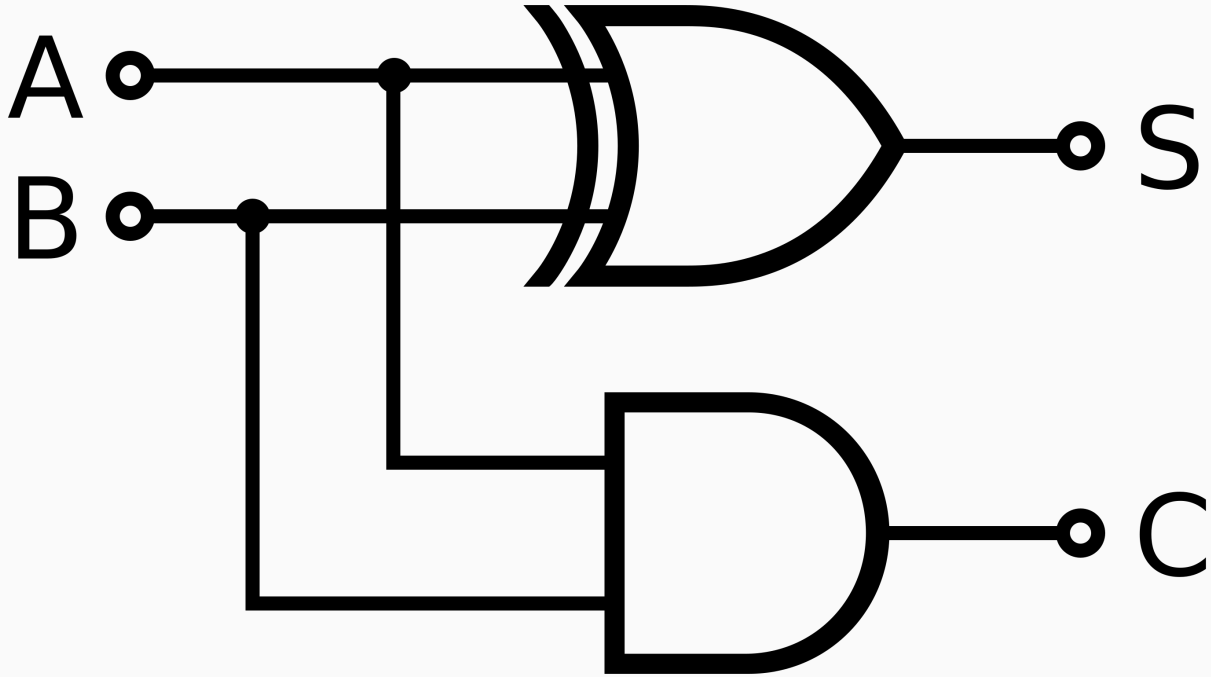


Figure 4: From Wikipedia

MAJORITY **in** TC^0

On the board

BFVP **in** NC^1

$$
\begin{aligned}
S &\to F_1 \\
F_1 &\to (F_1 \wedge F_1) \\
&\quad \mid (F_0 \vee F_1) \mid (F_1 \vee F_0) \mid (F_1 \vee F_1) \\
&\quad \mid (\neg F_0) \\
&\quad \mid 1 \\
F_0 &\to (F_0 \wedge F_0) \mid (F_0 \wedge F_1) \mid (F_1 \wedge F_0) \\
&\quad \mid (F_1 \vee F_1) \\
&\quad \mid (\neg F_1) \\
&\quad \mid 0
\end{aligned}
$$

Linguistically, the ability to evaluate Boolean formulas is directly relevant to computations underlying compositional semantics.

Uniformity

A circuit family contains a different circuit for each length n, with no constraint on the relationship between the circuits. This has some surprising consequences.

Example

Let L be any *unary* language, that is, $L \subseteq \{1\}^*$. For each $n \in \mathbb{N}$, if $1^n \in L$, let C_n be a circuit that always has value 1 (an AND gate with fan-in zero), and if $1^n \notin L$, let C_n be a circuit that has value 0 (an OR gate with fan-in zero). Then, L is recognized by a circuit family with $O(n)$ size and $O(1)$ depth, and is therefore in AC^0, even if it is undecidable.

DLOGTIME Uniform

To prevent such consequences, we impose a *uniformity* restriction, which says that, given n, the circuit C_n must be constructible under some limitation on computational resources, in the following sense.

Definition (DLOGTIME **uniformity, 1)**
Let $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ be a circuit family, and assume that the nodes of C_n are numbered from 0 to $|C_n| - 1$. We say that $\mathcal{C}$ is DLOGTIME-*uniform* if there is a (deterministic) Turing machine that runs in logarithmic time and accepts those tuples $\langle f, i, j, 1^n \rangle$ such that in C_n, node i has label f and there is an edge from node i to node j.

Some Language Classes

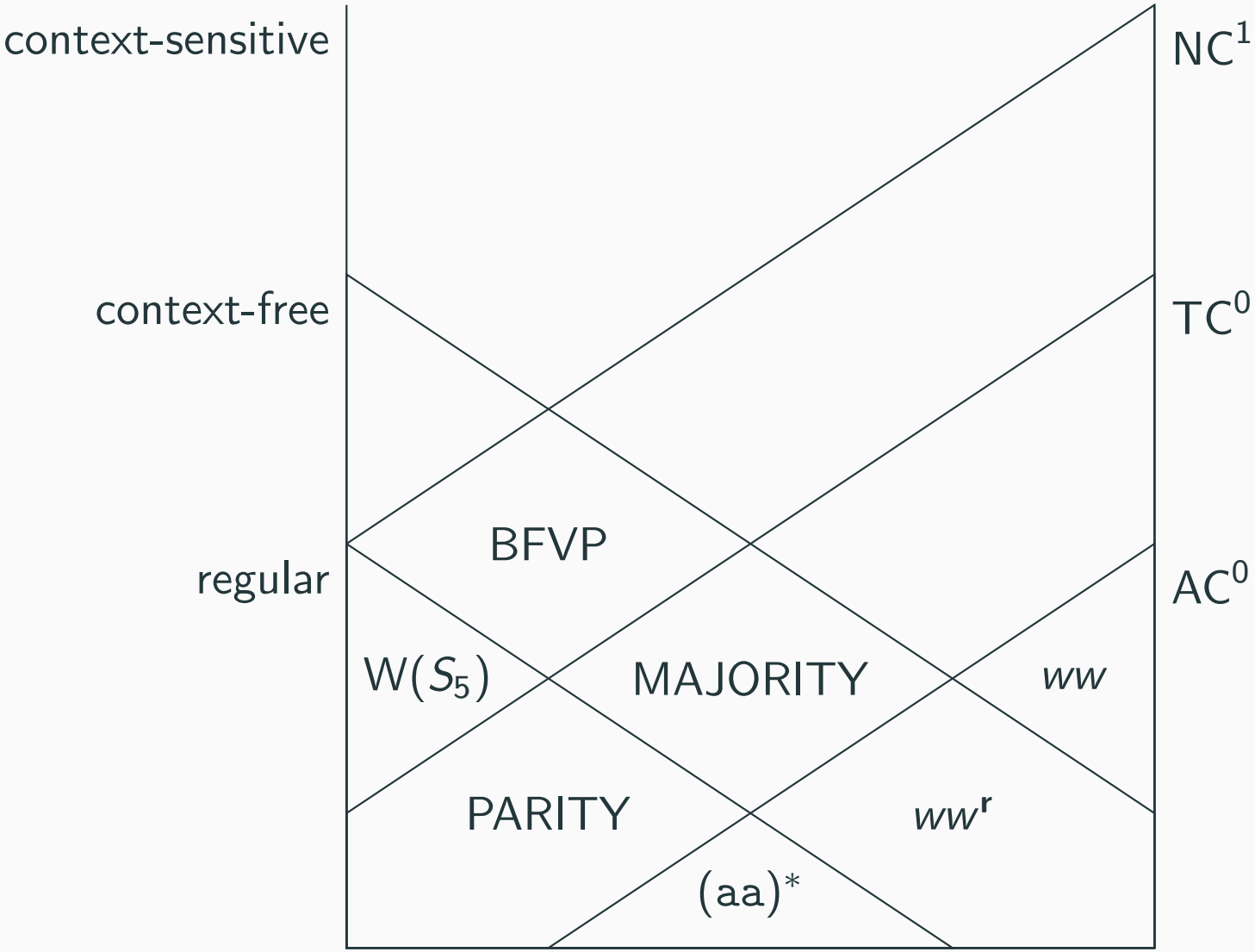


Figure 5: Some complexity classes defined by circuit families and logics, compared with the perhaps more familiar Chomsky hierarchy.

Inclusions

$$\mathsf{AC}^0 \subsetneq \mathsf{TC}^0 \subseteq \mathsf{NC}^1$$

References

[1] David A. Mix Barrington, Neil Immerman, and Howard Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41(3):274–306, 1990. doi: https://doi.org/10.1016/0022-0000(90)90022-D.