CS 579: Computational Complexity Fall 2025

Lecture 02 - August 28

Prof. Fernando Granha Jeronimo *Super Scribes: William Gay & Sasha Levinshteyn*

1 Space Bounded Computation

1.1 The Basic Complexity Classes

See the previous lecture notes for the definitions of some of the basic complexity classes.

1.2 Savitch's Theorem

Last lecture, we defined **DSPACE**$(s(n))$ and **NSPACE**$(g(n))$. A critical question is the relationship between deterministic space and nondeterministic space. Does nondeterminism grant any power in terms of space complexity?

Theorem 1 (Savitch's Theorem)**.** *We find that*

$$\mathbf{NSPACE}(g(n)) \subseteq \mathbf{DSPACE}(g(n)^2),$$

meaning that nondeterminism gives at most a quadratic improvement.

Definition 2. In order to show this theorem, we want to think about Turing machines. A snapshot of a Turing machine is called a configuration. We can capture all configurations and how the TM can move from one to another in a configuration graph. Note that this configuration graph can have at most $2^{O(s(n))}$ vertices, where $s(n)$ is the space usage of the TM. The configuration graph ends up being strongly explicit, meaning that it is easy to describe/enumerate the edges.

Definition 3 (STCONN)**.** STCONN is the problem of determining whether there is a path between two vertices s and t in a *directed* graph.

Algorithm. *Given a graph $G = (V, E)$ and two vertices s and t, we ask the following question as a subproblem: is there a path between s and t of length at most k? Let $P(s, t, m)$ be true if and only if there is a path from s to t with length at most 2^m. We have the following recurrence:*

$$P(s,t,m) = \bigvee_{z \in V} \left[P(s,z,m-1) \wedge P(z,t,m-1)\right].$$

This comes from splitting the path in two. This recurrence has $O(\log|V|)$ levels of recursion. Each level requires $O(\log|V|)$ bits of storage for the relevant variables. As such, the resulting algorithm for STCONN takes $O((\log|V|)^2)$ space.

Proof. Consider the configuration graph of our nondeterministic TM N with a space bound of $g(n)$. We let s be the starting configuration and t be a vertex connected to all the accepting configurations. Then, STCONN accepts if and only if the configuration graph of N has an accepting path.

Since the number of configurations is $2^{O(g(n))}$, STCONN will run on the configuration graph in $O(g(n)^2)$ space. We can then construct a deterministic TM M that checks the configuration graph and solves the same problem as N with $O(g(n)^2)$ space. □

2 Analysis of Boolean Functions

Boolean functions are fundamental objects in theoretical computer science. They are the functions that can be computed by Boolean circuits, decision trees, and related computational models. Formally, an *n-ary Boolean function* is a map

$$f : \{-1, 1\}^n \to \{-1, 1\}.$$

Throughout, we use the domain $\{-1, 1\}^n$ instead of $\{0, 1\}^n$, as this choice simplifies the algebraic and analytic treatment. There is a natural identification between $\{0, 1\}$ and $\{-1, 1\}$. Concretely, the map $x \mapsto (-1)^x$ is the unique non-trivial character of the group $\mathbb{Z}/2\mathbb{Z}$. This viewpoint connects Boolean functions with harmonic analysis on the group $(\mathbb{Z}/2\mathbb{Z})^n$.

2.1 The Function Space and Fourier Spectrum

We may also regard the elements of $\{-1, 1\}$ as real numbers. Thus we can treat Boolean functions as real-valued functions:

$$f : \{-1, 1\}^n \to \mathbb{R}.$$

This allows us to employ the tools of linear algebra, by viewing Boolean functions as elements of the real vector space $\mathbb{R}^{\{0,1\}^n}$. A natural basis for this space is given by the *characters* of $(\mathbb{Z}/2\mathbb{Z})^n$. For each subset $S \subseteq [n] := \{1, 2, \ldots, n\}$, define the character

$$\chi_S(x) := \prod_{i \in S} x_i, \quad x \in \{-1, 1\}^n.$$

For example:

$$\chi_\varnothing(x) = 1, \quad \chi_{\{i\}}(x) = x_i, \quad \chi_{\{i,j\}}(x) = x_i x_j, \text{ etc.}$$

Since the characters $\{\chi_S : S \subseteq [n]\}$ form a basis, every Boolean function f admits a unique expansion:

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\, \chi_S(x),$$

where the coefficients $\widehat{f}(S) \in \mathbb{R}$ are called the *Fourier coefficients* of f.

Example (XOR on 2 bits)**.** Define $f(x_1, x_2) = x_1 x_2$. This is already a character: $f = \chi_{\{1,2\}}$. Hence its Fourier expansion has a single coefficient $\widehat{f}(\{1, 2\}) = 1$.

Example (AND on 2 bits)**.** In our convention ($-1 = \text{TRUE}$, $+1 = \text{FALSE}$),

$$\text{AND}(x_1, x_2) = \max\{x_1, x_2\}.$$

Its Fourier expansion is

$$\text{AND}(x_1, x_2) = \tfrac{1}{2} + \tfrac{1}{2}x_1 + \tfrac{1}{2}x_2 - \tfrac{1}{2}x_1 x_2.$$

Example (OR on 2 bits)**.** Similarly,

$$\text{OR}(x_1, x_2) = \min\{x_1, x_2\}.$$

Its expansion is

$$\text{OR}(x_1, x_2) = -\tfrac{1}{2} + \tfrac{1}{2}x_1 + \tfrac{1}{2}x_2 + \tfrac{1}{2}x_1 x_2.$$

2.2 Inner Products and Expectations

We define the expectation inner product between two functions $f, g : \{-1,1\}^n \to \mathbb{R}$ by

$$\langle f, g\rangle := \mathop{\mathbf{E}}_{x\sim\{-1,1\}^n}[f(x)g(x)] = \frac{1}{2^n}\sum_{x\in\{-1,1\}^n} f(x)g(x).$$

Theorem 4 (Orthogonality of Characters)**.** *For subsets $S, T \subseteq [n]$, we have*

$$\langle \chi_S, \chi_T\rangle = \begin{cases} 1 & S = T, \\ 0 & S \neq T. \end{cases}$$

Proof. If $S = T$, then $\chi_S(x)\chi_T(x) = 1$ for all x, so the average is 1. If $S \neq T$, then there exists an index i in the symmetric difference $S \triangle T$. For half of the inputs x, we have $x_i = 1$, and for the other half $x_i = -1$. Thus $\mathbf{E}[\chi_S(x)\chi_T(x)] = 0$. □

By orthogonality, we obtain

$$\widehat{f}(S) = \langle f, \chi_S\rangle = \mathbf{E}[f(x)\chi_S(x)].$$

Theorem 5 (Average Value)**.** *The coefficient $\widehat{f}(\varnothing)$ equals the average value of f:*

$$\widehat{f}(\varnothing) = \mathbf{E}[f(x)].$$

Proof. By definition,

$$\widehat{f}(\varnothing) = \langle f, \chi_\varnothing\rangle = \mathbf{E}[f(x)\cdot 1] = \mathbf{E}[f(x)].$$

□

Theorem 6 (Inner Product Formula)**.** *For all $f, g : \{-1,1\}^n \to \mathbb{R}$,*

$$\langle f, g\rangle = \sum_{S\subseteq[n]} \widehat{f}(S)\widehat{g}(S).$$

Proof. Expanding both f and g,

$$\langle f, g\rangle = \Big\langle \sum_S \widehat{f}(S)\chi_S, \sum_T \widehat{g}(T)\chi_T \Big\rangle.$$

By bilinearity and orthogonality,

$$\langle f, g\rangle = \sum_{S,T} \widehat{f}(S)\widehat{g}(T)\langle\chi_S, \chi_T\rangle = \sum_S \widehat{f}(S)\widehat{g}(S).$$

□

Corollary 7 (Parseval's Identity)**.** *For all $f : \{-1,1\}^n \to \mathbb{R}$,*

$$\mathbf{E}[f(x)^2] = \sum_{S\subseteq[n]} \widehat{f}(S)^2.$$

Proof. We compute

$$\mathbf{E}[f(x)^2] = \langle f, f\rangle = \sum_S \widehat{f}(S)^2$$

□

Theorem 8 (Variance)**.** *For all* $f : \{-1,1\}^n \to \mathbb{R}$*,*

$$\mathbf{Var}[f(x)] = \sum_{\varnothing \neq S \subseteq [n]} \widehat{f}(S)^2$$

Proof. We compute

$$\mathbf{Var}[f(x)] = \mathbf{E}[f(x)^2] - \mathbf{E}[f(x)]^2 = \left(\sum_{S \subseteq [n]} \widehat{f}(S)^2 \right) - \widehat{f}(\varnothing)^2 = \sum_{\varnothing \neq S \subseteq [n]} \widehat{f}(S)^2. \qquad \square$$

2.3 The Boolean Hypercube and Hadamard Matrices

Another important structure related to the domain $\{-1,1\}^n$ is the *Boolean hypercube.* This is the graph on the vertex set $\{-1,1\}^n$ such that $x, y \in \{-1,1\}^n$ have an edge between them if their Hamming distance $\Delta(x,y)$ is exactly 1. The Hamming distance is defined by

$$\Delta(x,y) := |\, \{i \in [n] : x_i \neq y_i\} \,|.$$

The Hadamard matrix is the change-of-basis matrix from the standard basis (delta functions on $\{-1,1\}^n$) to the Fourier basis (characters).

Example (4 × 4 Hadamard matrix)**.** The 4 × 4 Hadamard matrix is given by

$$H_4 = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

It is easy to verify, for example, that $\delta_{(1,-1)} = \frac{1}{4}(\chi_\varnothing + \chi_{\{1\}} - \chi_{\{2\}} - \chi_{\{1,2\}})$.