Notes for Lecture 5

In this lecture we prove the Karp-Lipton theorem that if all NP problems have polynomial size circuits then the polynomial hierarchy collapses. The next result we wish to prove is that all approximate combinatorial counting problem can be solved within the polynomial hierarchy. Before introducing counting problems and the hashing techniques that will yield this result (which we will do in the next class), we prove the Valiant-Vazirani theorem that solving SAT on instances with exactly one satisfying assignment is as hard as solving SAT in general. The proof of the Valiant-Vazirani theorem will introduce the main ideas that we will develop in the next class.

1 The Karp-Lipton-Sipser Theorem

Theorem 1 (Karp-Lipton-Sipser) *If* $\mathbf{NP} \subseteq \mathbf{SIZE}(n^{O(1)})$ *then* $\mathbf{PH} = \Sigma_2$. *In other words, the polynomial hierarchy would collapse to its second level.*

Before proving the above theorem, we first show a result that contains some of the ideas in the proof of the Karp-Lipton-Sipser theorem.

Lemma 2 *If* $\mathbf{NP} \subseteq \mathbf{SIZE}(n^{O(1)})$ *then there is a family of polynomial-size circuits that on input a 3CNF formula* φ *outputs a satisfying assignment for* φ *if one such assignment exists and a sequence of zeroes otherwise.*

PROOF: We define the circuits $C_n^1, \ldots, C_n^n$ as follows:

- C_n^1, on input a formula φ over n variables outputs 1 if and only if there is a satisfying assignment for φ where $x_1 = 1$,
- $\cdots$
- C_n^i, on input a formula φ over n variables and bits $b_1, \ldots, b_{i-1}$, outputs 1 if and only if there is a satisfying assignment for φ where $x_1 = b_1, \ldots, x_{i-1} = b_{i-1}, x_i = 1$
- $\cdots$
- C_n^n, on input a formula φ over n variables and bits $b_1, \ldots, b_{n-1}$, outputs 1 if and only if φ is satisfied by the assignment $x_1 = b_1, \ldots, x_{n-1} = b_{n-1}, x_n = 1$.

Also, each circuit realizes an **NP** computation, and so it can be built of polynomial size. Consider now the sequence $b_1 = C_n^1(\varphi)$, $b_2 = C_n^2(b_1, \varphi)$, $\ldots$, $b_n C_n^n(b_1, \ldots, b_{n-1}, \varphi)$. The reader should be able to convince himself that this is a satisfying assignment for φ if φ is satisfiable, and a sequence of zeroes otherwise. □

We now prove the Karp-Lipton-Sipser theorem.

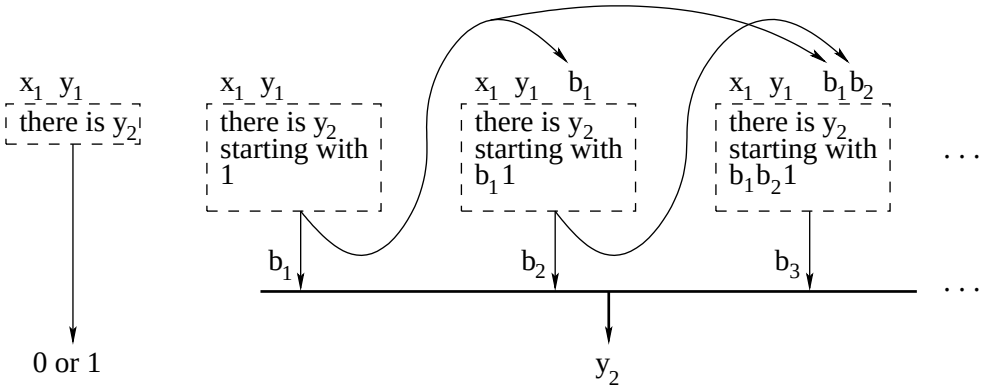

Figure 1: How to use decision problem solvers to find a witness to a search problem.

PROOF: [Of Theorem 1] We will show that if $\mathbf{NP} \subseteq \mathbf{SIZE}(n^{O(1)})$ then $\Pi_2 \subseteq \Sigma_2$. By a result in a previous lecture, this implies that $\mathbf{PH} = \Sigma_2$.

Let $L \in \Pi_2$, then there is a polynomial $p()$ and a polynomial-time computable $V()$ such that

$$x \in L \text{ iff } \forall y_1.|y_1| \leq p(|x|)\exists y_2.|y_2| \leq p(|x|).V(x, y_1, y_2) = 1$$

By adapting the proof of Lemma 2 (see Figure 1), or by using the statement of the Lemma and Cook's theorem, we can show that, for every n, there is a circuit C_n of size polynomial in n such that for every x and every y_1, $|y_1| \leq p(|x|)$,

$$\exists y_2.|y_2| \leq p(|x|) \wedge V(x, y_1, y_2) = 1 \text{ iff } V(x, y_1, C_n(x, y_1)) = 1$$

Let $q(n)$ be a polynomial upper bound to the size of C_n.

So now we have that for inputs x of length n,

$$x \in L \text{ iff } \exists C_n.|C_n| \leq q(n).\forall y_1.|y_1| \leq p(n).V(x, y_1, C_n(x, y_1)) = 1$$

which shows that L is in Σ_2. □

2 The Valiant-Vazirani Reduction

In this section we show the following: suppose there is an algorithm for the satisfiability problem that always find a satisfying assignment for formulae that have exactly one satisfiable assignment (and behaves arbitrarily on other instances): then we can get an **RP** algorithm for the general satisfiability problem, and so $\mathbf{NP} = \mathbf{RP}$.

We prove the result by presenting a randomized reduction that given in input a CNF formula φ produces in output a polynomial number of formulae $\psi_0, \ldots, \psi_n$. If φ is satisfiable, then (with high probability) at least one of the ψ_i is satisfiable and has exactly one satisfying assignment; if φ is not satisfiable, then (with probability one) all ψ_i are unsatisfiable.

The idea for the reduction is the following. Suppose φ is a satisfiable formula with n variables that has about 2^k satisfying assignments, and let $h : \{0,1\}^n \to \{0,1\}^k$ be a hash

function picked from a family of pairwise independent hash functions: then the average number of assignments x such that $\varphi(x)$ is true and $h(x) = (0, \ldots, 0)$ is about one. Indeed, we can prove formally that with constant probability there is exactly one such assignment,[1] and that there is CNF formula ψ (easily constructed from φ and h) that is satisfied precisely by that assignment. By doing the above construction for values of k ranging from 0 to n, we obtain the desired reduction. Details follow.

Definition 1 *Let H be a family of functions of the form $h : \{0,1\}^n \to \{0,1\}^m$. We say that H is a family of pair-wise independent hash functions if for every two different inputs $x, y \in \{0,1\}^n$ and for every two possible outputs $a, b \in \{0,1\}^m$ we have*

$$\mathbf{Pr}_{h \in H}[h(x) = a \wedge h(y) = b] = \frac{1}{2^{2m}}$$

Another way to look at the definition is that for every $x \neq y$, when we pick h at random then the random variables $h(x)$ and $h(y)$ are independent and uniformly distributed. In particular, for every $x \neq y$ and for every a, b we have $\mathbf{Pr}_h[h(x) = a | h(y) = b] = \mathbf{Pr}_h[h(x) = a]$.

For m vectors $a_1, \ldots, a_m \in \{0,1\}^m$ and m bits $b_1, \ldots, b_m$, define $h_{a_1,\ldots,a_m,b_1,\ldots,b_m} : \{0,1\}^n \to \{0,1\}^m$ as $h_{\mathbf{a},b}(x) = (a_1 \cdot x + b_1, \ldots, a_m \cdot x + b_m)$, and let H_{AFF} be the family of functions defined this way. Then it is not hard to see that H_{AFF} is a family of pairwise independent hash functions.

Lemma 3 *Let $T \subseteq \{0,1\}^n$ be a set such that $2^k \leq |T| < 2^{k+1}$ and let H be a family of pairwise independent hash functions of the form $h : \{0,1\}^n \to \{0,1\}^{k+2}$. Then if we pick h at random from H, there is a constant probability that there is a unique element $x \in T$ such that $h(x) = \mathbf{0}$. Precisely,*

$$\mathbf{Pr}_{h \in H}[|\{x \in T : h(x) = \mathbf{0}\}| = 1] \geq \frac{1}{8}$$

PROOF: Let us fix an element $x \in T$. We want to compute the probability that x is the *unique* element of T mapped into $\mathbf{0}$ by h. Clearly,

$$\mathbf{Pr}_h[h(x) = \mathbf{0} \wedge \forall y \in T - \{x\}.h(y) \neq \mathbf{0}] = \mathbf{Pr}_h[h(x) = \mathbf{0}] \cdot \mathbf{Pr}_h[\forall y \in T - \{x\}.h(y) \neq \mathbf{0} | h(x) = \mathbf{0}]$$

and we know that

$$\mathbf{Pr}_h[h(x) = \mathbf{0}] = \frac{1}{2^{k+2}}$$

The difficult part is to estimate the other probability. First, we write

$$\mathbf{Pr}_h[\forall y \in T - \{x\}.h(y) \neq \mathbf{0} | h(x) = \mathbf{0}] = 1 - \mathbf{Pr}_h[\exists y \in T - \{x\}.h(y) = \mathbf{0} | h(x) = \mathbf{0}]$$

[1] For technical reasons, it will be easier to prove that this is the case when picking a hash function $h : \{0,1\}^n \to \{0,1\}^{k+2}$.

And then observe that

$$\begin{aligned}
& \mathbf{Pr}_h[\exists y \in T - \{x\}.h(y) = \mathbf{0} | h(x) = \mathbf{0}] \\
\leq & \sum_{y \in |T| - \{x\}} \mathbf{Pr}_h[h(y) = \mathbf{0} | h(x) = \mathbf{0}] \\
= & \sum_{y \in |T| - \{x\}} \mathbf{Pr}_h[h(y) = \mathbf{0}] \\
= & \frac{|T| - 1}{2^{k+2}} \\
\leq & \frac{1}{2}
\end{aligned}$$

Notice how we used the fact that the value of $h(y)$ is independent of the value of $h(x)$ when $x \neq y$.

Putting everything together, we have

$$\mathbf{Pr}_h[\forall y \in T - \{x\}.h(y) \neq \mathbf{0} | h(x) = \mathbf{0}] \geq \frac{1}{2}$$

and so

$$\mathbf{Pr}_h[h(x) = \mathbf{0} \wedge \forall y \in T - \{x\}.h(y) \neq \mathbf{0}] \geq \frac{1}{2^{k+3}}$$

To conclude the argument, we observe that the probability that there is a unique element of T mapped into $\mathbf{0}$ is given by the sum over $x \in T$ of the probability that x is the unique element mapped into $\mathbf{0}$ (all this events are disjoint, so the probability of their union is the sum of the probabilities). The probability of a unique element mapped into $\mathbf{0}$ is then at least $|T|/2^{k+3} > 1/8$. □

Lemma 4 *There is a probabilistic polynomial time algorithm that on input a CNF formula φ and an integer k outputs a formula ψ such that*

- *If φ is unsatisfiable then ψ is unsatisfiable.*
- *If φ has at least 2^k and less than 2^{k+1} satifying assignments, then there is a probability at least $1/8$ then the formula ψ has exactly one satisfying assignment.*

PROOF: Say that φ is a formula over n variables. The algorithm picks at random vectors $a_1, \ldots, a_{k+2} \in \{0,1\}^n$ and bits $b_1, \ldots, b_{k+2}$ and produces a formula ψ that is equivalent to the expression $\varphi(x) \wedge (a_1 \cdot x + b_1 = 0) \wedge \ldots \wedge (a_{k+2} \cdot x + b_{k+2} = 0)$. Indeed, there is no compact CNF expression to compute $a \cdot x$ if a has a lot of ones, but we can proceed as follows: for each i we add auxiliary variables $y_1^i, \ldots, y_n^i$ and then write a CNF condition equivalent to $(y_1^i = x_1 \wedge a_i[1]) \wedge \cdots \wedge (y_n^i = y_{n-1}^i \oplus (x_n \wedge a_i[n] \oplus b_i)))$. Then ψ is the AND of the clauses in φ plus all the above expressions for $i = 1, 2, \ldots, k+2$.

By construction, the number of satisfying assignments of ψ is equal to the number of satisfying assignments x of φ such that $h_{a_1,\ldots,a_{k+2},b_1,\ldots,b_{k+2}}(x) = \mathbf{0}$. If φ is unsatisfiable, then, for every possible choice of the a_i, ψ is also unsatisfiable.

If φ has between 2^k and 2^{k+1} assignments, then Lemma 3 implies that with probability at least $1/8$ there is exactly one satisfying assignment for ψ. □

Theorem 5 (Valiant-Vazirani) *Suppose there is a polynomial time algorithm that on input a CNF formula having exactly one satisfying assignment finds that assignment. (We make no assumption on the behaviour of the algorithm on other inputs.) Then* **NP** = **RP**.

PROOF: It is enough to show that, under the assumption of the Theorem, 3SAT has an **RP** algorithm.

On input a formula φ, we construct formulae $\psi_0, \ldots, \psi_n$ by using the algorithm of Lemma 4 with parameters $k = 0, \ldots, n$. We submit all formulae $\psi_0, \ldots, \psi_n$ to the algorithm in the assumption of the Theorem, and accept if the algorithm can find a satisfying assignment for at least one of the formulae. If φ is unsatisfiable, then all the formulae are always unsatisfiable, and so the algorithm has a probability zero of accepting. If φ is satisfiable, then for some k it has between 2^k and 2^{k+1} satisfying assignments, and there is a probability at least $1/8$ that ψ_k has exactly one satisfying assignment and that the algorithm accepts. If we repeat the above procedure t times, and accept if at least one iteration accepts, then if φ is unsatisfiable we still have probability zero of accepting, otherwise we have probability at least $1 - (7/8)^t$ of accepting, which is more than $1/2$ already for $t = 6$. □

3 References

The Karp-Lipton-Sipser theorem appears in [KL80]. The Valiant-Vazirani result is from [VV86].

References

[KL80] R.M. Karp and R.J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 302–309, 1980. 5

[VV86] Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986. 5

Exercises

1. Define $\mathbf{EXP} = \mathbf{DTIME}(2^{n^{O(1)}})$. Prove that if $\mathbf{EXP} \subseteq \mathbf{SIZE}(n^{O(1)})$ then $\mathbf{EXP} = \Sigma_2$.