Lecture 11

In the first part of this lecture we will show that solving **NP** problems in randomized polynomial time can be reduced to solving the problem Unique SAT in randomized polynomial time. In the second part of the lecture we will start an introduction to cryptography.

1 Unique SAT and NP

In the last lecture when we proved that $\#\mathbf{P} \subseteq \mathbf{BPP}^{\mathbf{NP}}$, we saw how we can use hash functions in order to reduce the number of satisfying assignments to a constant number. We will next see a similar, but still different, argument in order to reduce 3SAT to USAT (Unique SAT).

The problem USAT is defined as follows. Given a 3CNF formula φ,

- if φ is unsatisfiable, output NO.
- if φ has exactly one satisfying assignment, output YES.
- otherwise, any answer is acceptable.

EXERCISE 1 Show that if $USAT \in \mathbf{BPP}$ then $USAT \in \mathbf{RP}$.

THEOREM 1 (VALIANT-VAZIRANI)
If $USAT \in \mathbf{BPP}$ then $\mathbf{NP} = \mathbf{RP}$.

Open question: can the reduction be derandomized? I.e., is the following implication true: $USAT \in \mathbf{P} \Rightarrow \mathbf{P} = \mathbf{NP}$?

We now recall a definition of from the last lecture.

DEFINITION 1 *A family of functions H_k, where $h : \{0,1\}^n \to \{0,1\}^k$ for each $h \in H_k$, is a family of pairwise independent hash functions if for every $a, b \in \{0,1\}^n$ s.t. $a \neq b$, for a random $h \in H_k$, $h(a)$ and $h(b)$ are independent, uniformly distributed in $\{0,1\}^k$.*

The following lemma was given in the last lecture:

LEMMA 2
$\{h_A\}_{A\in\{0,1\}^{n\times k}}$ where $h_A(x) = A \cdot x^T$ is a family of pairwise independent hash functions.

LEMMA 3
Let φ be a 3CNF formula with n variables and t satisfying assignments. Let k be s.t. $2^k \leq t \leq 2^{k+1}$. Let H_{k+2} be a family of pairwise independent hash functions where $h : \{0,1\}^n \to \{0,1\}^{k+2}$ for each $h \in H_{k+2}$. Then

$$\mathbf{Pr}\left[\left|\left\{a : a \text{ satisfies } \varphi \text{ and } h(a) = \vec{0}\right\}\right| = 1\right] \geq \frac{1}{8}$$

We will first prove Theorem 1 using the above lemmas and then go back and prove Lemma 3.

PROOF OF THEOREM 1: Suppose $USAT \in \mathbf{BPP}$, then according to Exercise 1 there is an **RP** algorithm for USAT. We can then construct the following **RP** algorithm for 3SAT. Given a formula φ, pick k at random between 0 and $n-1$, and then pick a random matrix $A \in \{0,1\}^n \times (k+2)$. Define $h : \{0,1\}^n \to \{0,1\}^{k+2}$ as $h(x) = Ax^T$. Define a formula φ' s.t. the number of satisfying assignments for φ' is equal to the number of assignments a s.t. a satisfies φ and $h(a) = \vec{0}$. Return YES iff USAT algorithm outputs YES given φ'.
We construct φ' in the following way. φ has variables $x_1, \ldots, x_n$. We want $\varphi' \Leftrightarrow \varphi \wedge (Ax^T = \vec{0})$. The condition $Ax^T = \vec{0}$ amounts to:

$$\begin{array}{ccccccc} a_{11}x_1 & + & \ldots & + & a_{1n}x_n & = & 0 \pmod 2 \\ \vdots & & & & \vdots & & \\ a_{k+2,1}x_1 & + & \ldots & + & a_{k+2,n}x_n & = & 0 \pmod 2 \end{array}$$

We can enforce these $k+2$ equations by adding $(k+2)n$ new variables $y_{11}, \ldots, y_{k+2,n}$ and adding clauses that for each $1 \le i \le k+2$ enforce:

$$\begin{aligned} &y_{i1} = a_{i1}x_1 \\ &y_{ij} = y_{i,j-1} + a_{ij}x_j \pmod 2 \\ &y_{in} \text{ is false} \end{aligned}$$

Thus, we added $O(kn)$ clauses with 1,2, or 3 variables each. Notice that for every assignment a to x s.t. $h(a) = \vec{0}$ there is exactly one assignment to y that satisfies all the new clauses, and thus the number of assignments that satisfies φ' is exactly the same as the number of assignments to x that satisfy φ and s.t. $h(x) = \vec{0}$.

We now analyze the correctness of the algorithm. Suppose φ has $t > 0$ satisfying assignments. Then with probability $\frac{1}{n}$ we chose a k s.t. $2^k \le t \le 2^{k+1}$. If k was chosen as such, then, according to Lemma 3, with probability $\ge \frac{1}{8}$ φ has a unique satisfying assignment a s.t. $h(a) = 0$. Thus, if $t > 0$ then with probability $\ge \frac{1}{8n}$ φ' is a YES instance of USAT. If $t = 0$ then clearly φ' has no satisfying assignment, and therefore it is a NO instance of USAT. Hence, when we apply the **RP** algorithm for USAT on φ' we will always output NO if φ is not satisfiable. If φ is satisfiable, we will output YES with probability $\ge \frac{1}{8n} \cdot \frac{1}{2} = \frac{1}{16n}$. As usual, we can enhance this probability by repeating the whole algorithm a number of times. $\Box$

PROOF OF LEMMA 3: Recall that φ has t satisfying assignments, that $2^k \le t \le 2^{k+1}$, and that the hash functions are $h : \{0,1\}^n \to \{0,1\}^{k+2}$. Let a be some satisfying assignment for φ. Then

$$\begin{aligned} &\mathbf{Pr}_{h \in H_{k+2}}[a \text{ is the unique sat. assignment of } \varphi \text{ s.t. } h(a) = \vec{0}] \; = \\ = \;& \mathbf{Pr}_h[h(a) = \vec{0} \wedge \forall b \neq a \;\; b \text{ sat. } \varphi \Rightarrow h(b) \neq \vec{0}] \; = \\ = \;& \mathbf{Pr}_h[\forall b \neq a \;\; b \text{ sat. } \varphi \Rightarrow h(b) \neq \vec{0} \mid h(a) = \vec{0}] \cdot \mathbf{Pr}_h[h(a) = \vec{0}] \end{aligned}$$

Clearly, $\mathbf{Pr}_h[h(a) = \vec{0}] = \frac{1}{2^{k+2}}$. As for the first term,

$$\begin{aligned}
&\mathbf{Pr}_h[\forall b \neq a \;\; b \text{ sat. } \varphi \Rightarrow h(b) \neq \vec{0} \mid h(a) = \vec{0}] \;\; = \\
&= \;\; 1 - \mathbf{Pr}_h[\exists b \neq a \;\; b \text{ sat. } \varphi \wedge h(b) = \vec{0} \mid h(a) = \vec{0}] \;\; \geq \\
&\geq \;\; 1 - \sum_{b \neq a \text{ sat. } \varphi} \mathbf{Pr}_h[h(b) = \vec{0} | h(a) = \vec{0}] \;\; =
\end{aligned}$$

using the pairwise independence and fact that $t \leq 2^{k+1}$ we get:

$$= \;\; 1 - \sum_{b \neq a \text{ sat. } \varphi} \mathbf{Pr}_h[h(b) = \vec{0}] \;\; \geq \;\; 1 \; - \; 2^{k+1} \cdot \frac{1}{2^{k+2}} \;\; = \;\; \frac{1}{2}$$

Thus,

$$\mathbf{Pr}_h[a \text{ is the unique sat. assignment of } \varphi \text{ s.t. } h(a) = \vec{0}] \;\; \geq \;\; \frac{1}{2^{k+3}}$$

We can now finish the proof of the Lemma. Notice that the events "a is the unique satisfying assignment s.t. $h(a) = \vec{0}$", where a varies over all satisfying assignments, are all disjoint. Therefore,

$$\begin{aligned}
&\mathbf{Pr}_h[\varphi \text{ has a unique sat. assignment } a : h(a) = \vec{0}] \;\; = \\
&= \;\; \sum_{a \text{ sat. } \varphi} \mathbf{Pr}_h[a \text{ is the unique sat. assignment of } \varphi \text{ s.t. } h(a) = \vec{0}] \;\; \geq \\
&\geq \;\; 2^k \cdot \frac{1}{2^{k+3}} \;\; = \;\; \frac{1}{8}
\end{aligned}$$

□

2 Cryptography

We now start an introduction to Cryptography. We will discuss private-key security and pseudorandomness.

Figure 1 illustrates the scenario in the private-key security framework. There are two parties, conventionally called Alice and Bob, who wish to transfer messages between themselves. A third party, conventionally called Eve (for eavesdropper), tries to listen to the messages, inject new content into them, or even destroy them. Alice and Bob share a private key (a sequence of random bits) known only to them.

We wish to come up with communication protocols having the following properties:

- Privacy (encryption) - Eve should have no information on the content of messages exchanged by Alice and Bob.
- Integrity (authentication) - Eve can not impersonate Alice/Bob and can not change the content of their messages.

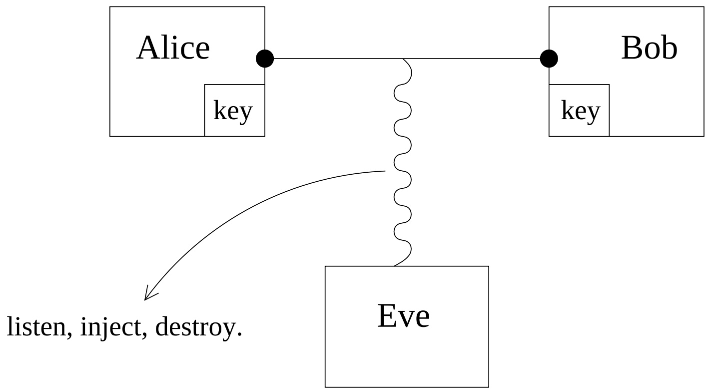

Figure 1: The setting for private-key security

- We will not deal with the possibility of Eve destroying the messages completely.

Constructing such protocols is related to constructing a pseudorandom generator. As illustrated in Figure 2, a pseudorandom generator takes a t bits long random seed and outputs an $m > t$ bits long sequence which "looks random".

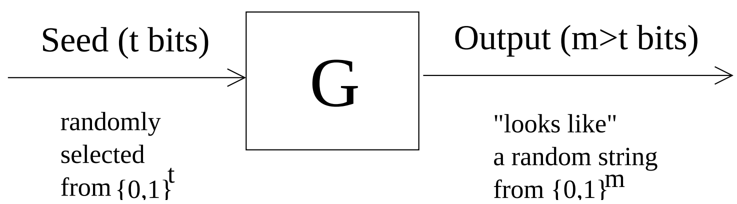

Figure 2: Pseudorandom generator

We now discuss what "looks random" means. We might require that the output sequence have some of the properties truly random sequences have. For instance:

- number of 0's and 1's differ by $O(\sqrt{m})$.
- the xor of a fixed subset of the bits is unbiased.
- there is a probability $\Theta(\frac{1}{m})$ that the output is a prime number.
- ...

The question is which and how many properties are enough to consider before concluding that the sequence looks random? As we will see next, we in fact consider all the possible properties which are computable under certain restrictions on the computation power.

Definition 2 *$G : \{0,1\}^t \to \{0,1\}^m$ is an (S, ϵ)-pseudorandom generator if for every property computable by a circuit C of size $\leq S$*

$$\left| \mathbf{Pr}_{x\in\{0,1\}^t}[C(G(x)) = 1] \;-\; \mathbf{Pr}_{z\in\{0,1\}^m}[C(z) = 1] \right| \;\leq\; \epsilon$$

DEFINITION 3 *$\{G_t\}$ $G_t : \{0,1\}^t \rightarrow \{0,1\}^m$ with $m > t$ is a pseudorandom generator if G_t is computable in poly(t) time and there are functions $S(t)$,$\epsilon(t)$ s.t. S and $\frac{1}{\epsilon}$ are super-polynomial and G_t is an $(S(t), \epsilon(t))$-pseudorandom generator.*

LEMMA 4
If $\mathbf{P} = \mathbf{NP}$ then there is no pseudorandom generator.

PROOF: A pseudorandom generator must be some $G_t : \{0,1\}^t \rightarrow \{0,1\}^m$ with $m \geq t+1$ where G_t is computable in polynomial time. We can define the property "being a possible output of G_t". The property is in **NP** and therefore in **P**. This property is true with probability 1 on output of the generator, but with probability $\leq \frac{1}{2}$ on the uniform distribution of m bits. $\Box$

Notice that in fact, according to our definition of pseudorandom generator, even a weaker condition implies there is no pseudorandom generator, namely, if polynomial size circuits can solve all **NP** problems on average, i.e. for inverse polynomial fraction of the inputs, then there is no pseudorandom generator.