Math 261C: Randomized Algorithms

Lecture topic: Primality Testing, and Randomized Petting Zoo

Lecturer: Sam Buss
Scribe notes by: Michelle Bodnar
Date: May 5, 2014

No lecture May 21, 23. We will have a two hour lecture on Friday May 30.

1. Miller-Rabin Primality Testing

The first version was a deterministic algorithm done by Miller in 1976, but required the assumption of the Extended Riemann Hypothesis (ERH). The second version was a randomized algorithm done by Rabin in 1980.

Recall that n is a **Carmichael number** if $a^{n-1} \equiv 1 \bmod n$ for all $a \in \mathbb{Z}_n^*$.

Lemma 1. *An integer n with prime factorization $n = p_1 p_2 \cdots p_l$ is a Carmichael number if and only if each p_i is distinct, and $(p_i - 1)|(n-1)$ for $i = 1, 2, \ldots, l$.*

Proof. Suppose n is a Carmichael number. Let $n = p_1^{k_1} \cdots p_l^{k_l}$ be the prime factorization of n. Since n is not prime, either $k_1 > 1$ or $l > 1$. Then we have $a^{n-1} \equiv 1 \bmod n$, so $a^{n-1} \equiv 1 \bmod p_i$ for each i. Since $\mathbb{Z}_{p_i^{k_i}}$ is cyclic, we have $\phi(p_i^{k_i})|(n-1)$, so $p_i^{k_i-1}(p_i - 1)|(n-1)$. If $k_i > 1$ then $p_i|(n-1)$, contradicting the fact that $p_i|n$.

The reverse direction is easier and left as an exercise. □

The Miller-Rabin Primality Algorithm is shown on the next page.

Miller-Rabin(n)
Input: $n \geq 2$ **is an odd integer.;**
Output: "Not prime" or "Not sure". Write $n-1 = 2^r R$ **where** R **is odd;**
Choose $a \in \mathbb{Z}_n \backslash \{0\}$ **at random;**
if $gcd(a, n) \neq 1$, **then**
  Output "Not prime";
for $k = 0$ *to* r **do**
  $b_k = a^{\frac{n-1}{2^k}}$ **mod** n**;**
end
if $b_0 \not\equiv 1$ *mod* n **then**
  Output "Not prime";
else
  Find the maximum j **such that** $b_j \equiv 1$ **mod** n**;**
  if $j = r$ **then**
    Output "Not sure";
  else if $b_{j+1} \neq -1$ *mod* n **then**
    Output "Not prime";
  else
    Output "Not sure";
  end
end

Algorithm 1: Miller-Rabin Algorithm

Claim: If n is composite, the algorithm outputs "Not prime" with probability at least $1/4$.

Proof. If n is not Carmichael, steps (2) and (3) output "Not prime" with probability at least $1/2$ by Solovay Strassen. If n is Carmichael, write $p_i - 1 = 2^{r_i} R_i$ where R_i is odd. Without loss of generality, assume $r_1 \geq r_2, \ldots, r_l$. Take $k = r - r_1$. Then $\frac{n-1}{2^k}$ is a multiple of $p_1 - 1$ but $\frac{n-1}{2^{k+1}}$ is not. Since $p_i - 1 | \frac{n-1}{2^k}$ for each i, $b_k \equiv 1 \bmod p_i$ so $b_k \equiv 1 \bmod n$.

$b_{k+1} \equiv 1 \bmod n$ if and only if either

(1) $a^{\frac{n-1}{2^{k+1}}} \equiv 1 \bmod p_i$ for all i or

(2) $a^{\frac{n-1}{2^{k+1}}} \equiv -1 \bmod p_i$ for all i.

(1) happens with probability $\leq 1/2$, since it happens with probability $1/2$ already for p_1.

(2) happens with probability $\leq 1/4$. To see this, note that if $r_1 > r_2, \ldots, r_l$ then (2) happens with probability 0 since $\frac{n-1}{2^{k+1}}$ is a multiple of $p_i - 1$ for $i \geq 2$. Otherwise $r_1 = r_2$, so $\text{Prob}(a^{\frac{n-1}{2^{k+1}}}) \equiv -1 \bmod p_i$ is equal to $1/2$ for $i = 1, 2$ independently. □

If the algorithm outputs "Not prime," then n is guaranteed to be composite. Running the algorithm multiple times will give us success with probability as close to 1 as we like. If the Extended Riemann Hypothesis (ERH) is true, then the algorithm can derandomized, as, assuming ERH, it suffices to consider only $a \leq O(\log(n^2))$. In this case, we could remove the randomization by checking just these small values.

Agrawal, Kayal, and Saxena gave a polynomial time deterministic algorithm for primality in 2004.

2. Randomized Complexity Mini Zoo (Petting Zoo)

We'll consider randomized, polynomial time algorithms which give yes/no answers (i.e., they accept or reject), and we'll work with languages $L \subseteq \{0,1\}^*$. By convention, R accepts if and only if R does not reject.

We say L is in PP if and only if for some deterministic polynomial time $R(x,y)$ and polynomial p we have that for all $x \in \{0,1\}^*$,

$$x \in L \iff \text{Prob}_{\substack{y \in \{0,1\}^* \\ |y|=p(|x|)}} (R(x,y) \text{ accepts}) \geq \frac{1}{2}.$$

We say L is in RP if and only if for some deterministic polynomial time $R(x,y)$ and polynomial p we have that for all $x \in \{0,1\}^*$,

$$x \in L \implies \text{Prob}_{\substack{y \in \{0,1\}^* \\ |y|=p(|x|)}} (R(x,y) \text{ accepts}) \geq \frac{1}{2}$$

$$x \notin L \implies \text{Prob}_{\substack{y \in \{0,1\}^* \\ |y|=p(|x|)}} (R(x,y) \text{ accepts}) = 0.$$

This is an example of one-sided error because $R(x,y)$ could reject even if $x \in L$, however $R(x,y)$ will never accept if $x \notin L$. We've shown that compositness is in RP. If you run an RP test k times, the probabilites become $1 - 2^{-k}$ and 0, instead of $1/2$ and 0.

We say L is in BPP if and only if for some deterministic polynomial time $R(x,y)$ and polynomial p we have that for all $x \in \{0,1\}^*$,

$$x \in L \implies \text{Prob}_{\substack{y \in \{0,1\}^* \\ |y|=p(|x|)}} (R(x,y) \text{ accepts}) \geq \frac{2}{3}$$

$$x \notin L \implies \text{Prob}_{\substack{y \in \{0,1\}^* \\ |y|=p(|x|)}} (R(x,y) \text{ accepts}) \leq \frac{1}{3}.$$

This is an example of two-sided error. Note that $RP \subseteq BPP$ and $RP \subseteq NP$, but it is not known if $BPP \subseteq NP$. It is conjectured that $BPP = P$.

By repeatedly choosing random y's, checking if $R(x, y)$ holds, and taking the majority answer, we can get a BPP algorithm to be polynomial time with probabilities amplified to $1 - 2^{-n}$ and 2^{-n}.

We say $f : \{0,1\}^*$ is in $\sharp P$ if and only if for some deterministic polynomial time $R(x, y)$ and polynomial p we have $f(x) = |\{y : |y| = p(|x|) \text{ and } R(x, y) \text{ accepts}\}|$.