Difficulties in Factoring a Number:
from the Perspective of Computation

電腦安全
海洋大學資訊工程系
丁培毅

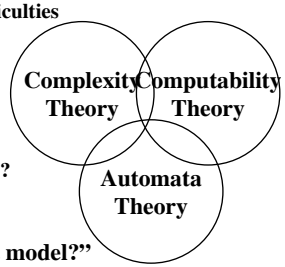
Prime Numbers

- **Prime number**: an integer p>1 that is divisible only by 1 and itself, ex. 2, 3, 5, 7, 11, 13, 17…
- **Composite number**: an integer n>1 that is not prime; can be expressible as a product a·b of integers with 1 < a, b< n; the prime factorization of n is unique
- **Fact**: there are infinitely many prime numbers. (by Euclid)
- **Prime Number Theorem**:
  the number of primes less than x, $\pi(x) \approx x/\ln x$

- How difficult is it to certify a prime number?
  How difficult is it to factor a composite number?

Computation Theory

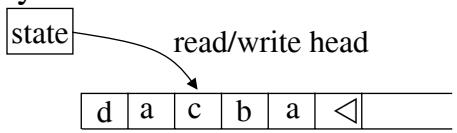
- **Complexity Theory:**
  central problem
  "What makes some problems computationally hard and others easy?"
  major achievements
  1. Schemes for classifying problems of different computational difficulties
  2. Options in confronting a difficult problem
  - **What is the most difficult part of a problem? Can we alter this part to avoid that problem?**
  - **Are there sub-optimal or heuristic solutions to a problem?**
  - **What kind of instance of a problem is hard?**
  - **Is there a randomized computable algorithm for a problem?**



- **Computability Theory:**
  central problem
  "What is computable? What is not computable? in what model?"
  major achievements
  1. Theoretical models of computers (ex. LBA, DTM, NTM, …)
  2. Classify problems as solvable or non-solvable
- **Automata Theory: definitions and properties of mathematical models of computation**
  - **Finite automata: text processing, compilers, H/W design**
  - **Push down automata: programming language, artificial intelligence**

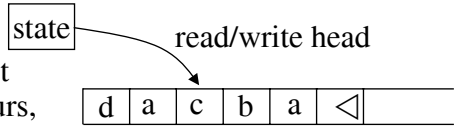
Turing Machine

- Complexity / Computability is defined w.r.t. a certain model of computation
- Turing Machine
  - Alan Turing, 1936
  - Similar to finite automaton but with an unlimited and unrestricted memory
  - Formally, a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$
    1. Q is the set of states
    2. Σ is the input alphabet not containing the special blank symbol $\triangleleft$
    3. Γ is the tape alphabet, where $\triangleleft \in \Gamma$ and $\Sigma \subseteq \Gamma$
    4. $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is the transition function
    5. $q_0 \in Q$ is the initial state
    6. $q_{accept} \in Q$ is the accept state
    7. $q_{reject} \in Q$ is the reject state, where $q_{reject} \neq q_{accept}$



Turing Machine (cont'd)

- TM computes as follows:
  - M's input $w = w_1 w_2 \ldots w_n \in \Sigma^*$ on the leftmost n squares of the tape, the rest of the tape are blanks $\triangleleft$ (the first $\triangleleft$ marks the end)
  - Initial state is q_0
  - read/write head starts on the leftmost square
  - Computation proceeds according to the transition function δ
  - If M tries to move its head to the left off the left hand end of the tape, the read/write head stays at the same place for that move
  - The computation continues until it enters either the accept or reject state. If neither occurs, M goes on forever.



DTM vs. NTM

- Deterministic Turing Machine: at any time, a DTM knows its next configuration (the state, the tape head, the tape content) for sure; a single configuration specified by its transition function

  $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

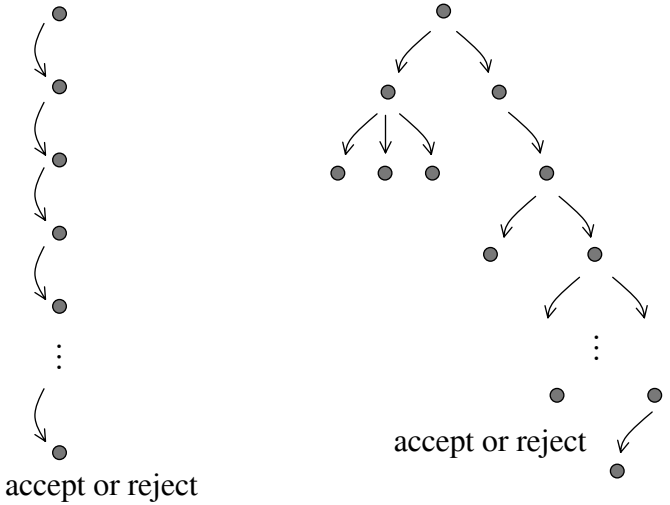
- Non-deterministic Turing Machine: at each moment, an NTM has several choices to proceed as the next configurations. i.e. the range of the transition function is modified to be a set:

  $\delta: Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R\})$

  - NTM has two equivalent evaluation ways if you only consider the capability:
    - Process in a parallel fashion
    - Process in a probabilistic fashion

    Probabilistic one seems slower. If you consider the time complexity, in polynomial time, the parallel one defines class NP, and the probabilistic one defines class BPP. Security professionals surely believe that BPP $\subset$ NP.
  - NTM can be proved to be equivalent to DTM

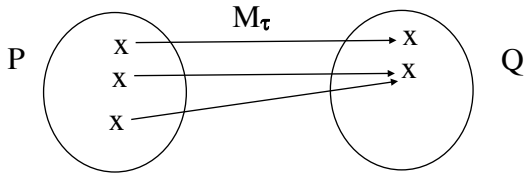
Deterministic vs. Nondeterministic



Note that an NTM decider halts on all branches.

Many-One (Mapping) Reducibility

Def: Given two problems P and Q, P is reducible to Q iff there exists a TM M_τ (computable function, algorithm, program, etc) which can transform every instance in P to an instance of Q



Properties: capture the difficulties between problems

- If Q is solvable, then P is also solvable
- If P is a well known unsolvable problem and can be reduced to Q, then Q is also unsolvable

Extension: Efficient mapping reducibility - M_τ is poly-time

Turing Reducibility

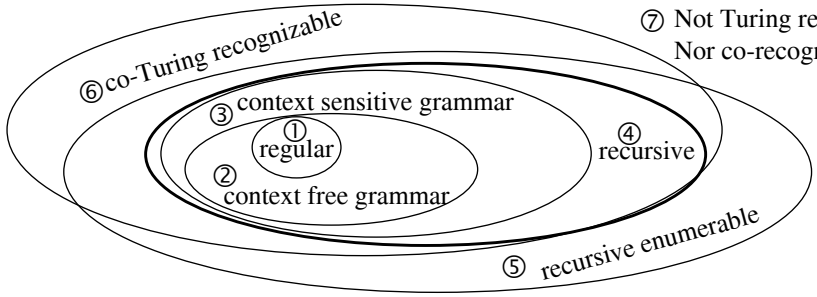
- There are some intuitive reducibility cases that cannot be captured by "mapping reduction", e.g.
  - A_{TM} and $\overline{A_{TM}}$ seem to be reducible to one another (a solution to either one could be used to solve the other by simply reversing the answer). However,
  - $\overline{A_{TM}}$ is not mapping reducible to A_{TM} because it is not Turing-recognizable (find a solution to map each unacceptable <M, w> to an acceptable <M, w> is clearly not possible)

Need a general notation that captures more problem reductions.

Def: Given two problems P and Q, P is Turing reducible to Q iff there exists an oracle TM M^A ,which given an oracle A for solving problem Q, can solve every instance in P

i.e. M^A using A as a subroutine (a blackbox) and can invoke A for (polynomially) many times

Language Classes



regular
regular expression ⊂ **context free grammar**
push down automata (PDA) ⊂ **context sensitive grammar**
linear bounded automata (LBA)

⊂ **solvable (problem)**
recursive (language)
Turing decidable
decidable
(TM can decide)
computable (function)
⊂ **Semi-solvable**
Recursive enumerable
Turing recognizable
Turing enumerable
? Primitive recursive function
? Unrestricted grammar
(TM can semi-decide)
⊂ **? μ-recursive function**

*** Unsolvable means undecidable (includes semi-solvable and totally unsolvable)**

Language Examples

① regular: closed under union, intersection, and complement
$\Sigma = \{0, 1\}$, 0^*10^*, $\Sigma^*1\Sigma^*$, $\Sigma^*001\Sigma^*$, $(\Sigma\Sigma)^*$, $01 \cup 10$, $(0 \cup \varepsilon)1^*$
$D=\{w|w$ has an equal number of occurrence of 01 and 10 as substrings$\}$,
$B_n=\{a^k \mid$ where k is a multiple of n, $n \geq 1\}$,
$C_n=\{x \mid x$ is a binary number that is a multiple of n$\}$

② CFL: closed under union
$B=\{0^n1^n \mid n \geq 0\}$, $C=\{w|w$ has equal number of 0's and 1's$\}$
$D=\{1^{n^2} \mid n \geq 0\}$, $E=\{0^i1^j \mid i \geq j\}$, $\{0^n1^m0^n \mid m, n \geq 0\}$
$\{0^m1^n \mid m \neq n\}$, $\{a^i b^j c^k \mid i,j,k \geq 0$ and i=j or i=k$\}$, $\{ww^R \mid w \in \{0,1\}^*\}$,
$\{w \mid w \in \{0,1\}^*$ and w is not a palindrome$\}$

③ CSL:
$\{w\,w \mid w \in \{0,1\}^*\}$, $\{w\,\#\,w \mid w \in \{0,1\}^*\}$, $\{w\,w\,w \mid w \in \{0,1\}^*\}$,
$\{a^n b^n c^n \mid n \geq 0\}$, $\{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$, $\{a^{2^n} \mid n \geq 0\}$,
$\{a^i b^j c^k \mid i \times j = k,\ i, j, k \geq 1\}$, $\{\#x_1\#x_2\#\ldots\#x_\ell \mid x_i \in \{0,1\}^*, x_i \neq x_j, \forall i \neq j\}$,
$\{<G> \mid G$ is a connected undirected graph$\}$, $\{w \mid w$ is a palindrome$\}$,
A_{REX}, E_{REX}, EQ_{REX}, A_{NFA}, E_{NFA}, EQ_{NFA}, A_{DFA}, E_{DFA}, A_{CFG}, E_{CFG}, A_{LBA}

Language Examples (cont'd)

④ Recursive (Turing computable, infinite tape required)

⑤ or ⑥ or ⑦
$\{p \mid p$ is a polynomial with two or more variables with an integral root$\}$ Hilbert's 10th
E_{LBA}, E_{TM}, $REGULAR_{TM}$, CFL_{TM} (Rice Thm: Testing any property (ex. CS, CF, regular, finite decidable) of L(M), M is a TM, is non-decidable), ALL_{CFG}, PCP, {incompressible strings}

⑤ Recursive Enumerable (Turing recognizable)
A_{TM}, $HALT_{TM}$

⑥ co-Turing recognizable
A_{TM}, $HALT_{TM}$, EQ_{CFG}, MIN_{TM}, Th(N, +, ×),

⑦ Not Turing recognizable nor co-Turing recognizable
EQ_{TM}, EQ_{TM},

Complexity Classes

- P: $O(n^k)$
  - the class of problems that can be polynomially decided by a DTM
- NP:
  - polynomially decided by an NTM (has a polynomial time verifier)
  - $P \subseteq NP$
  - worst case is difficult to solve, general instances might be easy
  - witness can be verified in polynomial time
- BPP:
  - polynomially decided by a probabilistic TM (a sort of NTM)
  - $BPP \subseteq NP$ (BPP =? NP)
- RP:
  - BPP with one-sided error probability (accept with error prob < 1/2, reject with prob. 1)
  - $RP \subset BPP \subseteq NP$

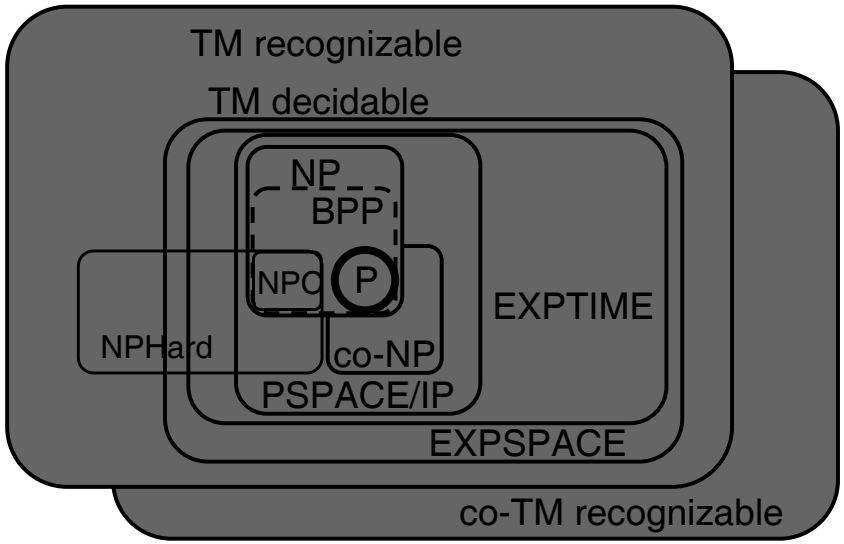
Complexity Classes (cont'd)

- EXPTIME: $O(2^{n^k})$
  - decided by a DTM in exponential time steps
  - $P \subseteq NP \subseteq EXPTIME$
  - n! is larger than e^n; however TSP $\in$ EXPTIME
- NP-hard: (non-deterministic polynomial time hard)
  - For all decision problems in NP, there is a polynomial-time many-one reduction to H, which is in NP-hard
  - the problem H is NP-hard if for every decision problem L in NP there is an oracle machine that has an oracle for solving H and this oracle machine can solve L in polynomial time (poly-time Turing reduction)
- NP-complete: NP-hard $\cap$ NP

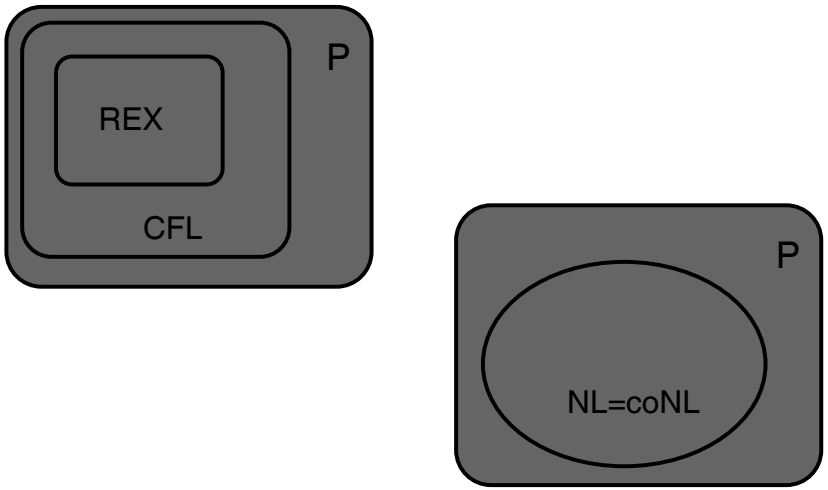
Complexity Classes (cont'd)

- PSPACE: can be solved by a deterministic TM with the memory requirement a polynomial in n
- NPSPACE: nondeterministic TM, polynomial space
  - $P \subseteq NP \subseteq PSPACE=NPSPACE \subseteq EXPTIME$
  - PSPACE-complete
- IP:
  - IP = PSPACE
- EXPSPACE: exponential space is required
- L: sublinear space, deterministic TM
- NL: sublinear space, nondeterministic TM
  - NL-complete

Refining Complexity Classes

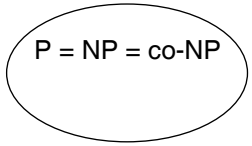


Refining Complexity Classes

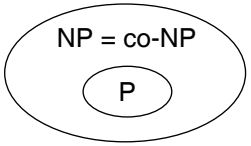


NP vs. co-NP

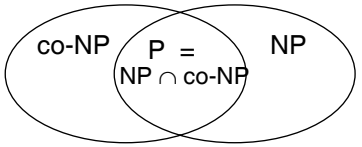
Four possibilities:

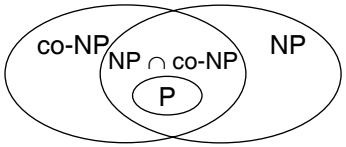


most unlikely



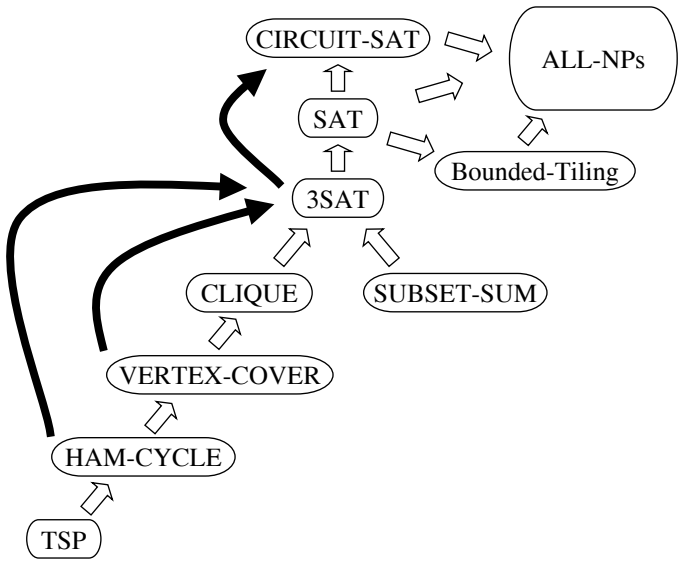
NP might be closed under complement





most likely

NP-Complete problems



Problem Definitions

- PRIMES:
  - the problem to decide if an integer is a prime number
  - in terms of language decidability, the language L is {the set of all prime numbers}
- COMPOSITES:
  - the problem to decide if an integer is composite (i.e. not prime)
- PAPP: (Prime $\vee$ Absolute Pseudo Prime)
  - the problem to decide if an integer passes Fermat tests to all bases (i.e. an absolute pseudoprime (Carmichael number) or a prime)
- FACTORING: (this is a search problem)
  - the problem to find factors of a composite integer

Pseudoprimes

- Def: a pseudoprime to the base b is a *composite* positive integer **n** such that the integer b satisfies $b^{n-1} \equiv 1 \pmod n$
- Ex. 341=11·31, 561=3·11·17, and 645=3·5·43 are pseudo-primes to the base 2
  - There are 455,052,512 primes less than 10^{10} but only 14884 pseudoprimes to the base 2.
  - There are infinitely many pseudoprimes to any given base.
  - Note: 341 is not a pseudoprime to the base 7
- Def: a Carmichael Number (an absolute pseudoprime) is a *composite* integer that satisfies $b^{n-1} \equiv 1 \pmod n$ for all positive integers b where gcd(b,n) = 1

Pseudoprimes (cont'd)

- Ex. 561=3·11·17 and 6601=7·23·41 are Carmichael numbers

If gcd(b, 561) = 1 then gcd(b,3)=gcd(b,11)=gcd(b,17)=1.
From Fermat's Little Theorem, $b^2 \equiv_3 1$, $b^{10} \equiv_{11} 1$, and $b^{16} \equiv_{17} 1$.
Consequently, $b^{560} \equiv_3 (b^2)^{280} \equiv_3 1$, $b^{560} \equiv_{11} (b^{10})^{56} \equiv_{11} 1$, and $b^{560} \equiv_{17} (b^{16})^{35} \equiv_{17} 1$.
By CRT, $b^{560} \equiv_{561} 1$

  - If $n=q_1q_2\ldots q_k$, q_j are distinct primes that satisfy $(q_j-1)|(n-1)$ for all j, then n is a Carmichael number
  - There are infinitely many Carmichael numbers. (conjectured 1912 by Carmichael, proved 1992 Alford, Granville and Pomerance)
  - 43 Carmichael numbers not exceeding 10^6, and 105,212 of them not exceeding 10^{15}
  - Carmichael numbers cannot be distinguished from a prime number by "Fermat Test" with respect to any integer base

PAPP

- Def:
  PAPP={p|p is a prime number or an absolute pseudoprime}
- Claim: PAPP is a decidable problem
- **Fermat Test** … a probabilistic poly-time algorithm to decide PAPP: given an integer p,

step 1. randomly pick a < p and compute $b \equiv_p a^{p-1}$
step 2. if $b \neq_p 1$ reject (i.e. declare $p \notin$ PAPP),
else repeat for k times
step 3. accept (declare $p \in$ PAPP) otherwise

- This PPT algorithm decides PAPP with a one-sided error rate, $\Pr\{\text{Fermat Test declares } x \in \text{PAPP} | x \notin \text{PAPP}\}=2^{-k}$
  $\Pr\{\text{Fermat Test declares } x \notin \text{PAPP} | x \in \text{PAPP}\}=0$

Error Probability of the Fermat Test

- Lemma: for any integer n > 1, if n fails the Fermat test to some base **a** in Z_n, then n fails the Fermat test to at least half of all numbers in Z_n

i.e. $n \notin$ PAPP

Proof:

given $a \in Z_n$ such that $a^{n-1} \neq_n 1$, (i.e. a is a witness for the composite number n)

we want to prove that for any non-witness h, i.e. $h^{n-1} \equiv_n 1$, there exists a unique witness t such that $t^{n-1} \neq_n 1$ i.e. #witnesses≥n/2

let $n = q \cdot r$ and gcd(q, r) = 1 (for applying CRT)

1. Construct $t \equiv_q h \equiv_r a$, in that case, $t^{n-1} \equiv_q 1 \neq_r 1$ i.e. t is a witness ($t^{n-1} \neq_n 1$)
   (note that we assume $a^{n-1} \neq_n kr + 1$; otherwise construct $t \equiv_r h \equiv_q a$)
2. if h' ≠ h then t' ≠ t from CRT, i.e. t is a distinct witness

Error Probability (cont'd)

- The previous lemma implies that for an $\mathbf{n} \notin$ PAPP
  if you randomly pick a number $\mathbf{a} \in Z_n$ and perform the Fermat test to this base on n, you have a probability greater than 0.5 for getting a witness in Z_n
  i.e.

  $\Pr\{\text{a single repetition of FT declares } n \notin \text{PAPP} \mid n \notin \text{PAPP}\} \geq 1/2$

  with k repetitions (each picks independently a base),

  $\Pr\{\text{Fermat Test declares } n \in \text{PAPP} \mid n \notin \text{PAPP}\} \leq 2^{-k}$

Miller-Rabin Test

- Fermat test cannot distinguish Carmichael numbers from true prime numbers while the "Miller-Rabin Test" can.
- Miller-Rabin test for primality utilizes another number theory property:

  - The number 1 has exactly two square roots, 1 and –1, modulo any prime number p
  - For a composite number c, could be a Carmichael number, 1 has four or more square roots modulo c

  - One pass in Miller-Rabin test:
    "if a number ***p*** passes the Fermat test to the base ***a***, the algorithm finds one of the square roots of 1 modulo p at random and determines whether that square root is 1 or -1. If it is not, we know that the number ***p*** is not a prime"
    i.e. starting from $1 \equiv_p a^{p-1}$, $a^{(p-1)/2}$ is a square root of 1, $a^{(p-1)/4}$ …

Basic Factoring Principle

- Let n be an integer and suppose there exist integers x and y with $x^2 \equiv y^2 \pmod n$, but $x \neq \pm y \pmod n$. Then ❶ n is composite, ❷ both gcd(x-y, n) and gcd(x+y, n) are nontrivial factors of n.

Proof:

let d = gcd(x-y, n).

Case 1: assume d = n $\Rightarrow$ $x \equiv y \pmod n$ contradiction

Case 2: assume d is 1 (the trivial factor)

$x^2 \equiv y^2 \pmod n \Rightarrow x^2 - y^2 = (x-y)(x+y) = k \cdot n$

d=1 means gcd(x-y, n)=1 $\Rightarrow$

$n \mid x+y \Rightarrow x \equiv -y \pmod n$ contradiction

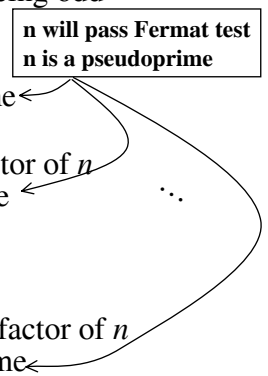
Case 1 and 2 implies that $1 < d < n$

i.e. d must be a nontrivial factor of n

One Pass of Miller-Rabin Primality Test

Is ***n*** a composite number?

- Let $n > 1$ be odd, write $n\text{-}1 = 2^k \cdot m$ with m being odd
- Choose a random integer ***a*** with $1 < a < n\text{-}1$
- Compute $b_0 \equiv a^m \pmod n$
  if $b_0 \equiv \pm 1 \pmod n$, stop, n is probably prime
- Compute $b_1 \equiv b_0^2 \pmod n$
  if $b_1 \equiv 1 \pmod n$, stop, $\gcd(b_0\text{-}1, n)$ is a factor of n
  if $b_1 \equiv -1 \pmod n$, stop, n is probably prime
- Compute $b_2 \equiv b_1^2 \pmod n$
  ……..
- Compute $b_{k-1} \equiv b_{k-2}^2 \pmod n$
  if $b_{k-1} \equiv 1 \pmod n$, stop, $\gcd(b_{k-2}\text{-}1, n)$ is a factor of n
  if $b_{k-1} \equiv -1 \pmod n$, stop, n is probably prime
- Compute $b_k \equiv b_{k-1}^2 \pmod n$
  if $b_k \equiv 1 \pmod n$, stop, $\gcd(b_{k-1}\text{-}1, n)$ is a factor of n
  otherwise n is composite (Fermat Little Thm, $b_k \equiv a^{n-1} \pmod n$)



One Pass of MRP Test (cont'd)

- In summary: there are 4 possible sorts of sequences for
  $b_0, b_1, b_2, \ldots b_{i-1}, b_i, \ldots b_k$:

342, 22, 5, 1, 1, 1, 1, …, 1	composite, factored
45, 5634, 325, 213, -1, 1, …, 1	possibly prime
±1, 1, 1, …, 1	possibly prime
214, 987, …, 8931, 321, 134	composite

Strong Pseudoprime

- If n passes the Miller-Rabin test with base ***a*** (without being identified as a composite), we say that n is a strong pseudoprime number to the base ***a***.
  - Ex. 2047 is a strong pseudoprime to the base 2
- Up to 10^{10}, there are only 3291 strong pseudoprime numbers *to the base 2*
- There are infinitely many strong pseudoprimes to the base 2
- There is no parallel set in strong pseudoprimes to the Carmichael numbers as to the pseudoprime.

Error Probability of MRP-Test

- Def: **PRIMES = {p|p is a prime number}**
- The Miller Rabin Primality test selects $a_1, \ldots, a_k$ randomly in Z_p, and repeats the previous square root test for k times, is a probabilistic polynomial time algorithm
- The maximum error probability is

$$\Pr\{\text{MR declares } x \in \text{PRIMES} \mid x \notin \text{PRIMES}\} = 2^{-k}$$

even stronger

$$\Pr\{\text{MR declares } x \in \text{PRIMES} \mid x \notin \text{PRIMES}\} = 4^{-k}$$

- On the other hand

$$\Pr\{\text{MR declares } x \in \text{PRIMES} \mid x \in \text{PRIMES}\} = 1$$

Error Probability (cont'd)

- Lemma 1: **$\Pr\{\text{MR declares } x \in \text{PRIMES} \mid x \in \text{PRIMES}\} = 1$**

  the MR algorithm rejects x only when 1) $a^{x-1} \neq_x 1$ and 2) successive square roots of a^{x-1}ever $\neq_x 1$; however, both cases imply that x must be a composite, contradiction with the assumption $x \in$ PRIME

- Lemma 2: **$\Pr\{\text{MR declares } x \in \text{PRIMES} \mid x \notin \text{PRIMES}\} = 2^{-k}$**

  We want to show that if p is an odd composite number and a is selected randomly in Z_p,

  $\Pr\{a \text{ is a composite witness}\} > 1/2$

  i.e. we would like to demonstrate that at least as many witnesses as non-witnesses exist in Z_p; we could prove that for any non-witness h, i.e. , there exists a unique witness b i.e. #witnesses>p/2

Error Probability (cont'd)

Ken Rose, Elementary Number Theory, 4-th Ed. A/W

- Thm 6.10 (in Ken. Rosen): If n is an odd composite positive integer, then n passes Miller-Rabin's test for at most (n-1)/4 bases b with $1 \le b \le n-1$
  - Stronger convergence property
- Thm 6.11:

$$\Pr\{\text{MR declares } x \in \text{PRIMES} \mid x \notin \text{PRIMES}\} = 4^{-k}$$

- Conjecture 6.1: Generalized Riemann hypothesis

  For every composite positive integer n, there is a base b with $b < 2(\log_2 n)^2$, such that n fails Miller-Rabin's test for the base b

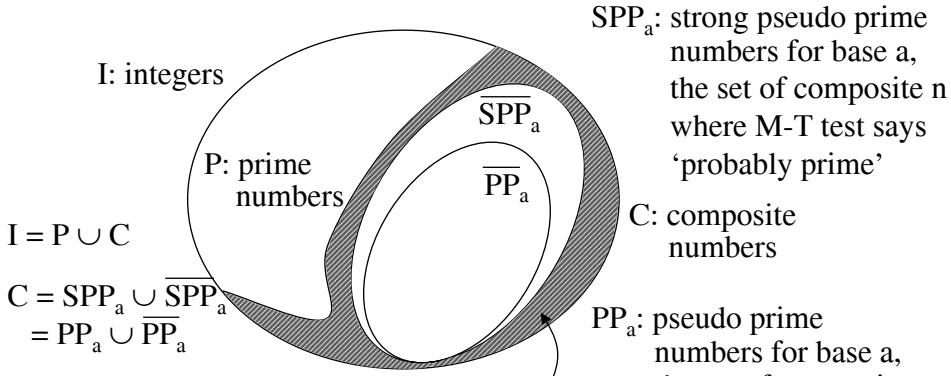
Error Probability (cont'd)

- Thm 6.12:

  If the generalized Riemann hypothesis is valid, then there is an algorithm to determine whether a positive integer n is prime using $O((\log_2 n)^2)$ bit operations

One Pass of Miller-Rabin Primality Test

- Both of these two tests can identify subsets of composite numbers



$I = P \cup C$

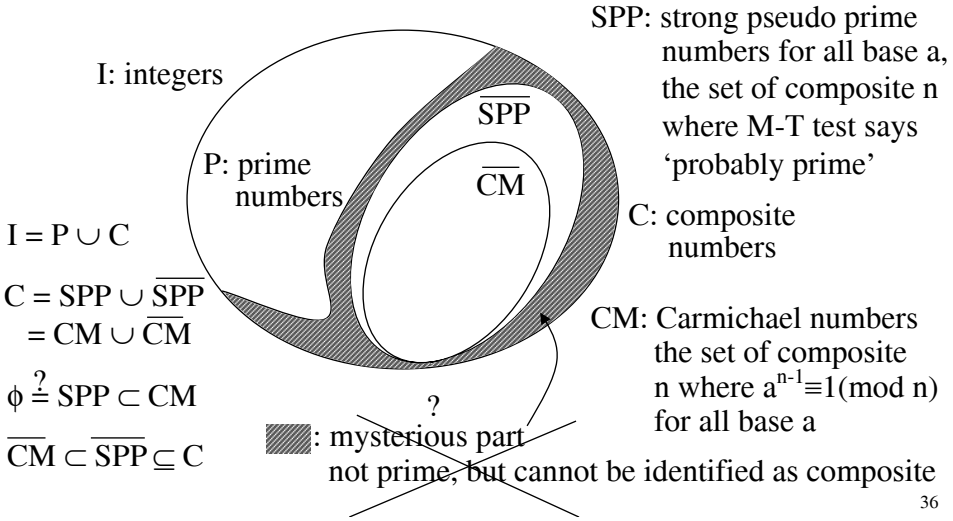
$C = SPP_a \cup \overline{SPP_a}$
$= PP_a \cup \overline{PP_a}$

$SPP_a \subset PP_a$

$\overline{PP_a} \subset \overline{SPP_a} \subset C$

Miller-Rabin Primality Test

- Both of these two tests can identify subsets of composite numbers



$I = P \cup C$

$C = SPP \cup \overline{SPP}$
$= CM \cup \overline{CM}$

$\phi \stackrel{?}{=} SPP \subset CM$

$\overline{CM} \subset \overline{SPP} \subseteq C$

Practical Question

- Consider a composite number $n = p \cdot q$, where p and q are two large prime numbers, each with k/2 bits
- Applying Miller-Rabin test on n for k times, the probability that n is not detected as a composite is less than 2^{-k} which is extremely small if k is say 1024
  - Note that n must at least satisfy $n \notin$ PAPP otherwise Miller-Rabin test will factor n in the process of identifying its compositeness
  - But there is still some chance that for some base a, n passes the Fermat test but detected by the Miller-Rabin test
- Is n still hard to be factored?
  Actually, factoring n is a hard non-poly time problem:

  GNFS: $\exp\{(1.923+O(1))\}(\ln(n))^{1/3}\ (\ln(\ln(n)))^{2/3}$

COMPOSITES

- COMPOSITES $\in$ NP
  - There are several kinds of witnesses for a composite number (an instance of COMPOSITES), ex:
    - A factor of it (one of them is enough) or
    - A positive integer a such that $a^{n-1} \neq 1 \pmod n$ or
    - A positive integer a such that $a^{n-1} \equiv 1 \pmod n$ and
      $a^{s2^j} \neq \pm 1 \pmod n$ and $a^{s2^{j+1}} \equiv 1 \pmod n$
      where $n-1 = s \cdot 2^k$ and s is an odd integer, $0 \leq j < k$
  - actually, COMPOSITES $\in$ RP $\subset$ BPP $\subset$ NP
    - use the probabilistic Miller-Rabin algorithm to decide if a number is a composite number
    - the error probability:
      - If $x \in$ COMPOSITES, $\Pr\{\text{accept } x\} > 1/2$
      - If $x \notin$ COMPOSITES, $\Pr\{\text{reject } x\} = 1$

PRIMES

- The complement of COMPOSITES
  - PRIMES $\in$ CoNP by definition
- PRIMES $\in$ NP
  - There are several kinds of witnesses for a prime number (an instance of PRIMES) ex.
    - Pratt certificate
    - Atkin-Goldwasser-Kilian-Morain certificate
  - PRIMES $\in$ RP $\subset$ BPP $\subset$ NP
    - use the probabilistic Miller-Rabin algorithm to decide if a number is a prime number
    - the error probability:
      - If $x \in$ PRIMES, $\Pr\{\text{accept } x\} = 1$
      - If $x \notin$ PRIMES, $\Pr\{\text{reject } x\} > 1/2$

Prime Witness: Pratt Certificate

- By applying Fermat's little theorem converse to n and recursively to each purported factor of n-1, a certificate for a given prime number n can be generated. (for prime $< 10^{10}$)
- ex. $n = 7919$, $n-1 = 7918 = 2 \cdot 37 \cdot 107$, let $a = 7$

  $7^{7918} \equiv_{7919} 1$, $7^{7918/2} \neq_{7919} 1$, $7^{7918/37} \neq_{7919} 1$, $7^{7918/107} \neq_{7919} 1$

  n = 2 is called "self-witness"

  $n = 37$, $n-1 = 36 = 2^2 \cdot 3^2$,

  let $a = 2$, $2^{36} \equiv_{37} 1$, $2^{36/2} \neq_{37} 1$, $2^{36/3} \neq_{37} 1$

  $n = 107$, $n-1 = 106 = 2 \cdot 53$,

  let $a = 2$, $2^{106} \equiv_{107} 1$, $2^{106/2} \neq_{107} 1$, $2^{106/53} \neq_{107} 1$

  $n = 53$, $n-1 = 52 = 2^2 \cdot 13$

  let $a = 2$, $2^{52} \equiv_{53} 1$, $2^{52/2} \neq_{53} 1$, $2^{52/13} \neq_{53} 1$

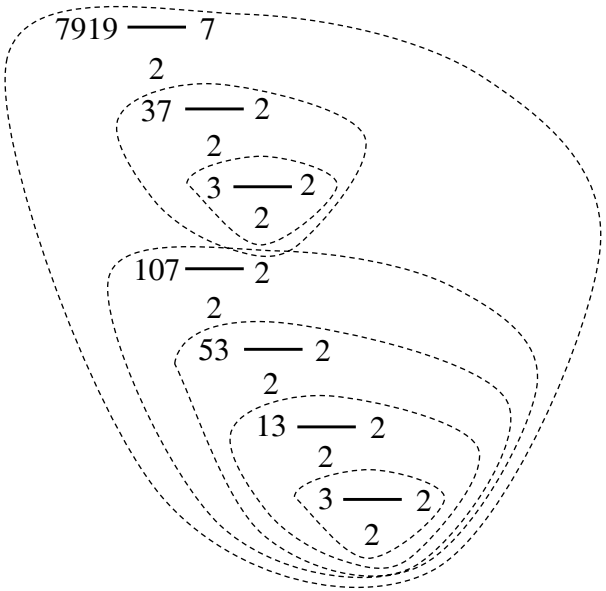
  $n = 13$, $n-1 = 12 = 2^2 \cdot 3$

  let $a = 2$, $2^{12} \equiv_{13} 1$, $2^{12/2} \neq_{13} 1$, $2^{12/3} \neq_{13} 1$

  $n = 3$, $n-1 = 2 = 2$

  let $a = 2$, $2^2 \equiv_3 1$, $2^{2/2} \neq_3 1$

Pratt Certificate: an example



7918=2·37·107
36=2^2·3^2
106=2·53
52=2^2·13
12=2^2·3
2 is a self witness

Atkin-Goldwasser-Kilian-Morain Certificate

- A recursive primality certificate: (for prime > 10^{10})
  - A point on an elliptic curve C

    $y^2 = x^3 + g_2 x + g_3 \pmod p$ for some number g_2 and g_3
  - A prime q with $q > (p^{1/4} + 1)^2$, such that for some other number k and m=kq with k≠1, $mC(x,y,g_{2,g3},p)$ is the identity on the curve, but $kC(x,y,g_2,g_3,p)$ is not the identity. This guarantees primality of p by a theorem of Goldwasser and Killian (1986).
  - Each q has its recursive certificate following it. So if the smallest q is known to be prime, all the numbers are certified prime up the chain.

Related Theorems

- Fermat's Little Theorem
- Euler's Theorem
- Carmichael Theorem
- Fermat Little Theorem Converse

("Fair-MAH")

Fermat's Little Theorem

- If p is a prime, p∤a then $a^{p-1} \equiv 1 \pmod p$

Proof:
- let S = {1, 2, 3, …, p-1} (Z_p^*), define $\psi(x) \equiv a \cdot x \pmod p$ be a mapping ψ: S→Z
- ∀x ∈ S, $\psi(x) \neq 0 \pmod p \Rightarrow$ ∀x ∈ S, ψ(x) ∈ S, i.e. ψ: S→S
  - if $\psi(x) \equiv a \cdot x \equiv 0 \pmod p \Rightarrow x \equiv 0 \pmod p$ since gcd(a, p) = 1
- ∀ x, y ∈ S, if x ≠ y then ψ(x) ≠ ψ(y) since
  - if $\psi(x) \equiv \psi(y) \Rightarrow a \cdot x \equiv a \cdot y \Rightarrow x \equiv y$ since gcd(a, p) = 1
- from the above two observations, ψ(1), ψ(2),... ψ(p-1) are distinct elements of S
- $1 \cdot 2 \cdot \ldots \cdot (p-1) \equiv \psi(1) \cdot \psi(2) \cdot \ldots \cdot \psi(p-1) \equiv (a \cdot 1) \cdot (a \cdot 2) \cdot \ldots \cdot (a \cdot (p-1))$
  $\equiv a^{p-1} (1 \cdot 2 \cdot \ldots \cdot (p-1)) \pmod p$
- since gcd(j, p) = 1 for j ∈ S, we can divide both side by 1, 2, 3, … p-1, and obtain $a^{p-1} \equiv 1 \pmod p$

Fermat's Little Theorem Converse

For an odd integer n, if $\exists\, a$, $a^{n-1} \equiv 1 \pmod n$ and
$\forall\, p_i$, where $n\text{-}1 = \Pi_i\, p_i^{r_i}$, $a^{(n-1)/p_i} \neq 1 \pmod n$
then
1. **$\text{ord}_n(a) = n\text{-}1$**
2. **n is a prime number**
3. **a is a primitive in Z_n^***

Proof: let $\text{ord}_n(a)$ be the smallest integer d such that $a^d \equiv_n 1$,

i.e. $a^{\text{ord}_n(a)} \equiv_n 1$, $\text{ord}_n(a) \le n-1$, let $n-1 = k \cdot \text{ord}_n(a) + r$

$a^{n-1} \equiv_n 1 \Rightarrow a^{n-1} \equiv_n a^{k \cdot \text{ord}_n(a)+r} \equiv_n 1 \Rightarrow 1 \equiv_n 1^k \cdot a^r \Rightarrow r=0$ i.e. $\text{ord}_n(a) \mid (n-1)$

$\Rightarrow \text{ord}_n(a)=n-1$ or

$\exists\, p_i$, $n-1=\Pi_i\, p_i^{r_i}$ s.t. $\text{ord}_n(a) \mid (n-1)/p_i$ i.e. $a^{(n-1)/p_i} \equiv_n (a^{\text{ord}_n(a)})^k \equiv_n 1$

$\Rightarrow$ $a^{n-1} \equiv_n 1$ and $\forall p_i$, where $n-1=\Pi_i\, p_i^{r_i}$, $a^{(n-1)/p_i} \neq_n 1 \Rightarrow \text{ord}_n(a)=n-1$

$\Rightarrow$ n is a prime number (for a composite number, the order of any a is at most $\phi(n)$, which is strictly less than n-1) and a is a primitive

Euler's Theorem

This is true even when $n = p^2$

✧ If gcd(a,n)=1 then $a^{\phi(n)} \equiv 1 \pmod n$

Proof:
- ✡ let S be the set of integers $1 \le x \le n$, with gcd(x, n) = 1, define $\psi(x) \equiv a \cdot x \pmod n$ be a mapping ψ: S→Z
- ✡ $\forall x \in S$ and gcd(a, n) = 1, $\psi(x) \neq 0 \pmod n$, $\gcd(\psi(x), n) = 1$ $\Rightarrow \forall x \in S$, $\psi(x) \in S$, i.e. ψ: S→S
  - if $\psi(x) \equiv a \cdot x \equiv 0 \pmod n \Rightarrow x \equiv 0 \pmod n$
  - gcd(a, n)=1 and gcd(x, n) = 1
- ✡ $\forall$ x, y ∈ S, 'if $x \neq y$ then $\psi(x) \neq \psi(y) \pmod n$'
  - if $\psi(x) \equiv \psi(y) \Rightarrow a \cdot x \equiv a \cdot y \Rightarrow x \equiv y$ since gcd(a, n) = 1
- ✡ from the above two observations, $\forall x \in S$, $\psi(x)$ are distinct elements of S (i.e. $\{\psi(x) \mid \forall x \in S\}$ is S)
- ✡ $\prod_{x \in S} x \equiv \prod_{x \in S} \psi(x) \equiv a^{\phi(n)} \prod_{x \in S} x \pmod n$
- ✡ since gcd(x, n) = 1 for x ∈ S, we can divide both side by x ∈ S one after another, and obtain $a^{\phi(n)} \equiv 1 \pmod n$

Carmichael Theorem

Carmichael's Theorem:

$\forall a \in Z_n^*$, $a^{\lambda(n)} \equiv 1 \pmod n$ and $a^{n \cdot \lambda(n)} \equiv 1 \pmod{n^2}$
where $n=p \cdot q$, $p \neq q$, $\lambda(n) = \text{lcm}(p-1, q-1)$, $\lambda(n) \mid \phi(n)$

✧ like Euler's Theorem, we can prove it through Fermat's Little Theorem, consider $n = p \cdot q$, where $p \neq q$,

$\forall a \in Z_p^*$, $a^{p-1} \equiv 1 \pmod p \Rightarrow (a^{p-1})^{(q-1)/\gcd(p-1,q-1)} \equiv a^{\lambda(n)} \equiv 1 \pmod p$

$\forall a \in Z_q^*$, $a^{q-1} \equiv 1 \pmod q \Rightarrow (a^{q-1})^{(p-1)/\gcd(p-1,q-1)} \equiv a^{\lambda(n)} \equiv 1 \pmod q$

from CRT, $\forall a \in Z_p^* \cap Z_q^* = Z_n^*$, $a^{\lambda(n)} \equiv 1 \pmod n$

therefore, $\forall a \in Z_n^*$, $a^{\lambda(n)} = 1 + k \cdot n$

raise both side to the n-th power, we get $a^{n \cdot \lambda(n)} = (1 + k \cdot n)^n$,

$\Rightarrow a^{n \cdot \lambda(n)} = 1 + n \cdot k \cdot n + \ldots \Rightarrow \forall a \in Z_n^*$ (or $Z_{n^2}^*$), $a^{n \cdot \lambda(n)} \equiv 1 \pmod{n^2}$

Primitive Roots modulo p

✧ When p is a prime number, a primitive root modulo p is a number whose powers yield every nonzero element mod p. (equivalently, the order of a primitive root is p-1)

✧ ex: $3^1 \equiv 3$, $3^2 \equiv 2$, $3^3 \equiv 6$, $3^4 \equiv 4$, $3^5 \equiv 5$, $3^6 \equiv 1 \pmod 7$
3 is a primitive root mod 7

✧ sometimes called a multiplicative generator

✧ there are plenty of primitive roots, actually $\phi(p-1)$
- ✶ ex. p=101, $\phi(p-1)=100 \cdot (1-1/2) \cdot (1-1/5)=40$
  p=143537, $\phi(p-1)=143536 \cdot (1-1/2) \cdot (1-1/8971)=71760$

Primitive Testing Procedure

- How do we test whether h is a primitive root modulo p?
  - naïve method:
    go through all powers h^2, h^3, …, h^{p-2}, and make sure $\neq 1$ modulo p
  - faster method:
    assume p-1 has prime factors q_1, q_2, …, q_n,
    for all q_i, make sure $h^{(p-1)/q_i}$ modulo p is not 1,
    then h is a primitive root

Intuition: let $h \equiv g^a \pmod p$, if gcd(a, p-1)=d (i.e. g^a is not a primitive root), $(g^a)^{(p-1)/q_i} \equiv (g^{a/q_i})^{(p-1)} \equiv 1 \pmod p$ for some $q_i \mid d$

Primitive Testing Procedure (cont'd)

- Procedure to test a primitive g:

> assuming p-1 has prime factors q_1, q_2, …, q_n, (i.e. $p-1 = q_1^{r_1} \ldots q_n^{r_n}$)
> for all q_i, make sure $g^{(p-1)/q_i} \pmod p$ is not 1

Proof:

(a) by definition, $g^{ord_p(g)} \equiv 1 \pmod p$, $g^{\phi(p)} \equiv 1 \pmod p$ therefore $ord_p(g) \leq \phi(p)$
if $\phi(p) = ord_p(g) * k + s$ with $s < ord_p(g)$
$g^{\phi(p)} \equiv g^{ord_p(g) * k} g^s \equiv g^s \equiv 1 \pmod p$, but $s < ord_p(g) \Rightarrow s = 0$
$\Rightarrow$ $ord_p(g) \mid \phi(p)$ and $ord_p(g) \leq \phi(p)$

(b) assume g is not a primitive root i.e $ord_p(g) < \phi(p)=p-1$
then $\exists$ i, such that $ord_p(g) \mid (p-1)/q_i$ i.e. $g^{(p-1)/q_i} \equiv 1 \pmod p$ for some q_i

(c) if for all q_i, $g^{(p-1)/q_i} \neq 1 \pmod p$
then $ord_p(g) = \phi(p)$ and g is a primitive root modulo p