The Iterated Mod Problem

Soham Chatterjee

Chennai Mathematical Institute

July 24, 2025

Contents

Introduction

- This paper is Iterated Mod Problem by Karloff and Ruzzo [KR89]
- Sequential algorithm for computing gcd is based on Euclidean Algorithm $r_0 = a$, $r_1 = b$. Then

$$r_2 = r_0 \bmod r_1, \quad r_3 = r_1 \bmod r_2, \quad \cdots$$

  gcd is the last nonzero r_i.
- But parallel complexity of gcd is poorly understood. Fastest parallel algorithm takes $O\left(\frac{n}{\log n}\right)$ time [CG90]
- gcd for polynomials is in NC
- The problem we will study related to the gcd problem. It is given integers or polynomials $x, m_n, m_{n-1}, \ldots, m_1$ find if

$$((x \bmod m_n) \bmod m_{n-1}) \cdots) \bmod m_1) = 0$$

Iterated Integer Mod Problem

Introduction

Problem:
Given positive integers $x, m_n, m_{n-1}, \ldots, m_1$ find if

$$((x \bmod m_n) \bmod m_{n-1}) \cdots) \bmod m_1) = 0$$

Theorem

Iterated Iinteger Mod $\in$ P

For any 2 numbers a and b, a **mod** b is in P. Here we are doing n iterated mods. So it still takes polynomial time. So IIM $\in$ P.

Circuit Value Problem

Theorem ([Lad75])

Circuit Value Problem is P-complete.

- Enough to take CVP for circuits with only NAND gates, NANDCVP

$$\text{Gates} \in [G]$$

$$\text{Input Variables} := y_i, i \in [r], \text{ Input Bits} := Y_i, i \in [r]$$

NANDCVP $\leq_l$ IIM

Log-Space Reduction

Let $n = 2G$.

- x is $n+1$-bit integer whose ith bit is Y_j if the ith edge is incident from the input y_j. Otherwise it is 1.
- $1 \leq g \leq G$

$$m_{2g} = 2^{2g} + 2^{2g-1} + \sum_{\substack{\text{jth edge} \\ \text{out-edge from g}}} 2^j \text{ and } m_{2g-1} = 2^{2g-1}$$

Remark: Here m_{2g} and m_{2g-1} simulate the gate g

NANDCVP $\leq_l$ IIM I

Correctness

Theorem

Let $x_{G+1} = x$. And for all $1 \leq g \leq G$ $x_g = ((\cdots((x \bmod m_{2G}) \bmod m_{2G-1}) \cdots \bmod m_{2g}) \bmod m_{2g-1}) = 0$.
Then:

1. For all $1 \leq g \leq G+1$, $x_g \leq 2^{2g-1}$
2. For all $1 \leq g \leq G+1$, $0 \leq j \leq 2g-1$ if the jth edge is an outgoing edge from an input node or from a gate h such that $h \geq g$ then x_g's jth bit is the value carried by jth edge otherwise 1

NANDCVP $\leq_l$ IIM II

Correctness

Prove by downward induction:

Base Case ($g = G + 1$): We have $x < 2^{2(G+1)-1} = 2^{2G+1} = 2^{n+1}$. True as x is n-bit number. And second condition follows by constuction. Let the theorem holds for all $g > k$.

NANDCVP $\leq_l$ IIM III

Correctness

Part (a):

$x_k = (x_{k+1} \bmod m_{2k}) \bmod m_{2k-1}$. $m_{2k-1} = 2^{2k-1}$. So x_k has $2k-1$ bits so $x_k < 2^{2k-1}$. So Part (a) is proved.

NANDCVP $\leq_l$ IIM IV

Correctness

Part (b):

- The only bits differ between x_{k+1} and x_k are the bits corresponding to edges incident on kth vertex (in and out). In x_{k+1} the jth bits are 1 if jth edge going out from gate k.
- The 2k and $2k-1$th edges are in edges of gate k. So in x_{k+1} the $(2k)$th and $(2k-1)$th bits are the value carried by the $(2k)$ and $(2k-1)$th edges. Two cases to consider:

NANDCVP $\leq_l$ IIM V

Correctness

Both $(2k)$ and $(2k+1)$th bits are 1:
$m_{2k} \leq x_{k+1} < 2m_{2k}$. So

$$(x_{k+1} \bmod m_{m_{2k}}) \bmod m_{2k-1} = x_{k+1} - m_{2k}$$

So in x_{2k} at output bits position of m_{2k} the 1 in replaced by 0

At least one of the bits is 0:

$$x_{k+1} < m_{2k} \implies x_{k+1} \bmod m_{2k} = x_{k+1}$$

So in x_{2k} at output bits position of m_{2k} has 1.

IIM is P-complete

$x_1 < 2^1$ is the value carried by the 0th edge, value of the CVP instance.

Theorem

NANDCVP $\leq_l$ Iterated Integer Mod

Theorem

Integer Iterated Mod Problem is P-complete

Super Increasing Knaspsack Problem (SIK)

Introduction

Definition (0-1 Knapsack Problem)

Given an integer w and a sequence of integers $w_1, w_2, \ldots, w_n$ is there a sequence of $0-1$ valued variables $x_1, \ldots x_n$ such that $w = \sum_{i=1}^{n} x_i w_i$.

- 0-1 Knapsack Problem is known to be NP-complete. [GJ90]
- A knapsack instance is called super increasing (SIK) if each weight w_i is larger than the sum of the previous weights i.e. for all $2 \leq i \leq n$ we have $w_i > \sum_{j=1}^{i-1} w_j$

Super Increasing Knaspsack Problem (SIK)

Introduction

Theorem

Super Increasing Knaspsack Problem $\in$ P

Greedy strategy considering the w_i in decreasing order gives a linear time algorithm for solving super increasing knapsack problem.

SIK is P-complete I

Theorem

If $w_1, \ldots, w_n$ are such that $\forall\, i \in [n-1] \sum_{k=1}^{i} w_k < w_{i+1}$ then there is a 0-1 sequence of variables $x_1, \ldots, x_n$ such that $\sum_{i=1}^{n} x_i w_i = w$ iff

$$((\cdots((w \bmod w_n) \bmod w_{n-1})\cdots) \bmod w_2) \bmod w_1 = 1$$

SIK is P-complete II

Observe: The previous reduction the modulo numbers doesn't satisfy super increasing knapsack condition.

- Need to find another reduction of NANDCVP to IIM where modulo numbers are super increasing to work with above theorem !!

SIK is P-complete III

- Let x is $n+1$-length base 4 number whose ith digit is Y_j if the ith edge is incident from the input y_j. Otherwise it is 1.
- $1 \leq g \leq G$

$$m_{2g} = 4^{2g} + 4^{2g-1} + \sum_{\substack{\text{jth edge} \\ \text{out-edge from g}}} 4^j$$

$$m_{2g-0.5} = 4^{2g} - 4^{2g-1} = 3 \times 4^{2g-1}, \quad m_{2g-1} = 4^{2g-1}$$

SIK is P-complete IV

Define for all $1 \leq g \leq G$,
$x_g = (((\cdots(((x \bmod m_{2G}) \bmod m_{2G-0.5}) \bmod m_{2G-1})\cdots) \bmod m_{2g}) \bmod m_{2g-0.5}) \bmod m_{2g-1} = 0$ and $x_{G+1} = x$.

- $x_g \leq 4^{2g-1}$ for all $1 \leq g \leq G+1$

SIK is P-complete V

Theorem

For all $1 \leq g \leq G + 1$, $0 \leq j \leq 2g - 1$ if the jth edge is an outgoing edge from an input node or from a gate h such that $h \geq g$ then x_g's jth bit is the value carried by jth edge otherwise 1

SIK is P-complete VI

- Prove by downward induction. Base case $g = G + 1$ is true.
- x_{k+1} and x_k differs at the positions corresponding to the edges incident on kth vertex.
- $2k$ and $2k - 1$th edges are in-edges of vertex k so they are the values carried by $2k$ and $2k - 1$th edges

SIK is P-complete VII

If both of them 1:

$$4m_{2k} > x_{k+1} \geq m_{2k} \implies x_{k+1} \bmod m_{2k} = x_{k+1} - m_{2k} < 4^{2k-1}$$

$$(x_{k+1} - m_{2k} \bmod m_{2k-0.5}) \bmod m_{2k-1} = x_{k+1} - m_{2k}$$

In x_k the positions where m_{2k} has 1 will have 0.

SIK is P-complete VIII

If at least one of them 0:
$x_{k+1} \bmod m_{2k} = x_{k+1}$. In x_k positions where m_{2k} has 1 will have 1.

$$x_{k+1} = a \times 4^{2k} + b \times 4^{2k-1} + c \text{ where } a, b \in \{0, 1\}$$

- $a = 1$, $b = 0$:

  $$(x_{k+1} \bmod m_{2k-0.5}) \bmod m_{2k-1} = 1 \times 4^{2k-1} + c \bmod m_{2k-1} = c$$

- $a = 0$:

  $$(x_{k+1} \bmod m_{2k-0.5}) \bmod m_{2k-1} = b \times 4^{2k-1} + c \bmod m_{2k-1} = c$$

SIK is P-complete IX

After m_1, $x_1 < 2^1$ is the value carried by the 0th edge, the value of the CVP.

- Notice: The modulos satisfies the super increasing knapsack problem.

Since

$$\sum_{g=1}^{k} m_{2g} + m_{2g-0.5} + m_{2g-1} = \sum_{g=1}^{k} m_{2g} + 4^{2g} < 4^{2k+1} = m_{2(k+1)-1}$$

SIK is P-complete X

1. Sum of weights till m_{2k} is strictly $< m_{2(k+1)-1}$
2. $$\begin{aligned} & \text{Sum of weights till } m_{2(k+1)-1} \\ = \; & (\text{sum of weights till } m_{2k}) + m_{2(k+1)-1} \\ < \; & 2 \times 4^{2(k+1)-1} < 3 \times 4^{2(k+1)-1} = m_{2(k+1)-0.5} \end{aligned}$$
3. $$\begin{aligned} & \text{Sum of weights till } m_{2(k+1)-0.5} \\ = \; & (\text{sum of weights till } m_{2k}) + m_{2(k+1)-1} + m_{2(k+1)-0.5} \\ < \; & 2 \times 4^{2(k+1)-1} + 3 \times 4^{2(k+1)+1} \\ = \; & 4^{2(k+1)} + 4^{2(k+1)-1} < m_{2(k+1)} \end{aligned}$$

SIK is P-complete XI

Theorem

NANDCVP $\leq_l$ Super Increasing Knapsack

Theorem

Super Increasing Knapsack Problem is P-complete.

Polynomial Iterated Mod Problem

Introduction

Definition (Polynomial Iterated Mod Problem)

Given univariate polynomials $a(x)$, $b_1(x), \ldots, b_n(x)$ over a field $\mathbb{F}$ compute the residue
$((\cdots((a(x) \bmod b_1(x)) \bmod b_2(x))\cdots) \bmod b_{n-1}(x)) \bmod b_n(x)$

- A polynomial mod can't test for two bits

$$(10)_2 \bmod (11)_2 = (10)_2 \text{ but } (x^2 + 0x) \bmod (x^2 + x) = 0x^2 - x$$

Theorem

Polynomial Iterated Mod Problem is in P

Lower Triangular Matrix Inversion

Theorem ([Hel74],[Hel78])

For any field $\mathbb{F}$, lower triangular matrix inversion is in $\text{Arithmetic} - \text{NC}$

Theorem ([BvzGH82],[BCP84])

Lower triangular matrix inversion is in NC over finite fields and $\mathbb{Q}$

Reduction I

Given $a(x), b_1(x), \ldots, b_n(x)$ over $\mathbb{F}$.
$b_0(x) = r_0(x) = a(x)$ and $d_i = \deg b_i(x)$ for all $0 \leq i \leq n$.
Assume $d_0 \geq d_1 > \cdots > d_n$

$$\begin{aligned} a(x) &= q_1(x)b_1(x) + r_1(x) \\ &= q_1(x)b_1(x) + q_2(x)b_2(x) + r_2(x) \\ &\vdots \\ &= q_1(x)b_1(x) + \cdots + q_n(x)b_n(x) + r_n(x) \end{aligned}$$

$r_{i-1}(x) = q_i(x) \cdot b_i(x) + r_i(x)$ with $\deg r_i < \deg b_i = d_i$ or $r_i = 0$

Reduction II

The coefficient of x^j in $a(x), b_i(x), q_i(x), r_i(x)$ are a_j, $b_{i,j}$, $q_{i,j}$, $r_{i,j}$.

- $\deg q_1 = d_0 - d_1$, $\deg q_i \leq d_{i-1} - d_i - 1$
- Compare the coefficients of x^j in both direction.
- $(d_0 + 1) \times (d_0 + 1)$ matrix M. Denote the variable matrix for coefficients of q_i and r_n as X

Reduction III

$d_0 -$ i-th entry of MX is coefficient of degree i. $d_k \leq i < d_{k-1}$.

$r_n(x) + \sum_{i=K+1}^{n} q_i(x)b_i(x)$ doesn't take part in coefficient of x^i.

$$i = d_k + (d_{k-1} - d_k - 1 - (d_{k-1} - 1 - i)) = d_k + (i - d_k)$$

Can't go lower $(d_{k-1} - d_k - 1 - (d_{k-1} - 1 - i))$ for coefficient of q_k

$$d_0 - i = (d_0 - d_1 + 1) + (d_1 - d_2) + \cdots (d_{k-2} - d_{k-1}) + (d_{k-1} - 1 - i)$$

So M has at $(d_0 - i, d_0 - i)$th entry b_{k,d_k} and after that all entries are 0 in that row. Hence M is lower triangular.

Matrix is non-singular since the diagonal entries are the leading coefficients of $b_i(x)$

Reduction IV

We need to inverse M which is in Arithmetic – NC for general fields and for finite fields, $\mathbb{Q}$ it is in NC.

Theorem

Iterated Polynomial Mod Problem is in NC for finite field and $\mathbb{Q}$ and in Arithmetic – NC for general field.

Thank You!

References I

[BCP84] A Borodin, S Cook, and N Pippenger.
Parallel computation for well-endowed rings and space-bounded probabilistic machines.
Inf. Control, 58(1–3):113–136, jul 1984.

[BvzGH82] Allan Borodin, Joachim von zur Gathen, and John Hopcroft.
Fast parallel matrix and gcd computations.
In Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, SFCS '82, page 65–71, USA, 1982. IEEE Computer Society.

[CG90] Benny Chor and Oded Goldreich.
An improved parallel algorithm for integer GCD.
Algorithmica, 5(1):1–10, 1990.

References II

[GJ90] Michael R. Garey and David S. Johnson.
Computers and Intractability; A Guide to the Theory of NP-Completeness.
W. H. Freeman & Co., USA, 1990.

[Hel74] Don Heller.
A determinant theorem with applications to parallel algorithms.
SIAM J. Numer. Anal., 11(3):559–568, jun 1974.

[Hel78] Don Heller.
A survey of parallel algorithms in numerical linear algebra.
SIAM Rev., 20(4):740–777, oct 1978.

[KR89] Howard J. Karloff and Walter L. Ruzzo.
The iterated mod problem.
Information and Computation, 80(3):193–204, 1989.

References III

[Lad75] Richard E. Ladner.
The circuit value problem is log space complete for p.
SIGACT News, 7(1):18–20, jan 1975.