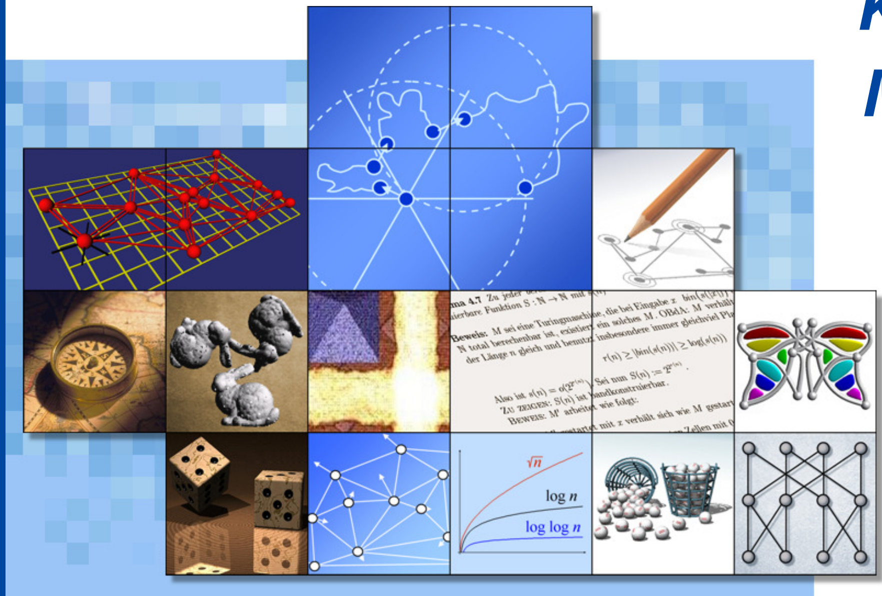
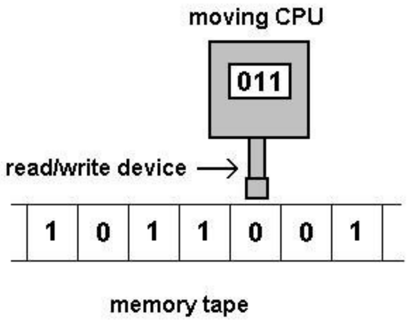
HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

On Faster Integer Calculations using Non-Arithmetic Primitives

Katharina Lürwer-Brüggemeier
Martin Ziegler

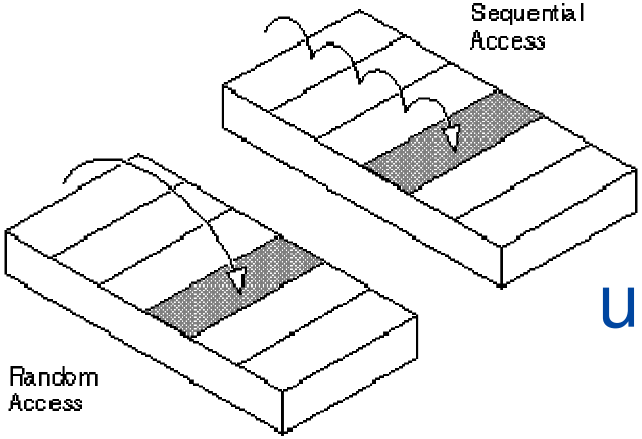


Turing machine



bit cost model

RAM



unit cost model

Compute the number 2^{2^k}.

In the bit cost model $\mathcal{O}(\mathbf{2^k})$

1	0	0	0	…	0	0	0	0

$$\underbrace{0\ 0\ 0\ \ldots\ 0\ 0\ 0\ 0}_{2^k}$$

In the unit cost model $\mathcal{O}(\mathbf{k})$

$$(((2\underbrace{^2)^2)^2\ldots)^2}_{k}$$

Is the number $x \in \mathbb{N}$ even ?

It can be decided in $\mathcal{O}$**(log x)** steps over $\{+,-,\times, =\}$.

Find k with $2^k > x$

For i:=k-1 to 1

If $2^i \leq x$; x:= x - 2^i next

If x=0 accept else reject

It can be decided in $\mathcal{O}$**(1)** steps over $\{+,-,\times,$**div**$, =\}$.

If x mod 2 = x - 2·(x div 2) = 0 accept else reject

Examples

a) Given $a, k \in \mathbb{N}$ and some arbitrary $b \in \mathbb{N}$, $b>a^{2^k}$, one can compute a^{2^k} over $\{+,-, \times, \text{div}\}$ in $\mathcal{O}(\sqrt{k})$ steps.

b) Over $\{+,-,\times_c, \text{div}, \leq \}$, not only primality test but even factorization of a given x is possible in time $\mathcal{O}(\log x)$ linear in the binary length.

c) Over $\{+,-, \times, \text{div} ,\leq \}$and using indirect addressing the greatest common divisor gcd(x,y) of given integers, $x,y \leq N$ can be calculated in $\mathcal{O}(\log N/\log \log N)$.

d) Over $\{+,-,\leq,\&, \rightarrow,\leftarrow\}$(but without indirect addressing as for Bucket Sort) n given integers $x_1,\ldots, x_n$ can be sorted in $\mathcal{O}(n)$.

Polynomial Evaluation

With Horner's rule, a polynomial $\sum_{n=0}^{d} p_n x^n = p_0 + x(p_1 + x(p_2 + \ldots x(p_{d-1} + p_d x)\ldots))$ can be calculated using $\mathcal{O}(d)$ operations over $\{+,\times\}$.

Given $p_0,\ldots,p_{d-1} \in \mathbb{Z}$, $|p_n| < P$ and $x \in \mathbb{Z}$, a polynomial $\sum_{n=0}^{d} p_n x^n$ can be calculated using $\mathcal{O}(d/\log_p d)$ operations over $\{+,-,\times,=\}$.

Proof

wlog $p_n \geq 0$ For $k \in \mathbb{N}$ decompose p into $\lceil d/k \rceil$ polynomials $q_i \in \mathbb{N}[X]$ with deg $q_i < k$.

P^k distinct polynomials with coefficients in $\{0,1,\ldots,P-1\}$

Evaluate all at $x \in \mathbb{Z}$ in $\mathcal{O}(kP^k)$ with Horner's rule

Evaluate at $y = x^k \in \mathbb{Z}$ $\sum_{i=0}^{\lceil d/k \rceil} q_i Y^i$ in $\mathcal{O}(d/k)$ with Horner's rule

In total $\mathcal{O}(kP^k + d/k)$

For $k := \log_p d - 2\log_p \log_p d$ in $\mathcal{O}(d/\log_p d)$.

Throwing in integer division

For $N \in \mathbb{N}$ and $Z > \max_{0 \leq x \leq N} p(x)$ $\quad Y := \sum_{x=0}^{N} p(x) Z^x$
$P(x) = (Y \text{ div } Z^x) \bmod Z$
can be calculated in $\mathcal{O}(\log x)$ over $\{+,-,\times,\text{div}\}$.

Y

P(N)	…	P(x+1)	P(x)	P(x-1)	….	P(2)	P(1)	P(0)
Z^N		Z^{x+1}	Z^x	Z^{x-1}		Z^2	Z	

Y div Z^x

P(N)	…	P(x+1)	P(x)
Z^{N-x}	Z^2	Z	

Ymod Z

P(x)

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

A polynomial p(x) can be evaluated on a finite domain D in ℕ over {+,-,×c, div} in constant time independent of p and D.

$$P(x) = \sum_{i=0}^{d} p_i x^i \; ; \; P = \sum_{i=0}^{d} |p_i|;$$

$$Z > \max\{N^d P, (N^d+1)N\}$$

$$g(x) = Z^{d+1} \text{ div } (Z - x)$$

$$h(x) = P(Z)\, g(x)$$

$$a(x) = h(x) \text{ div } Z^d$$

$$b(x) = a(x) \bmod Z = P(x)$$

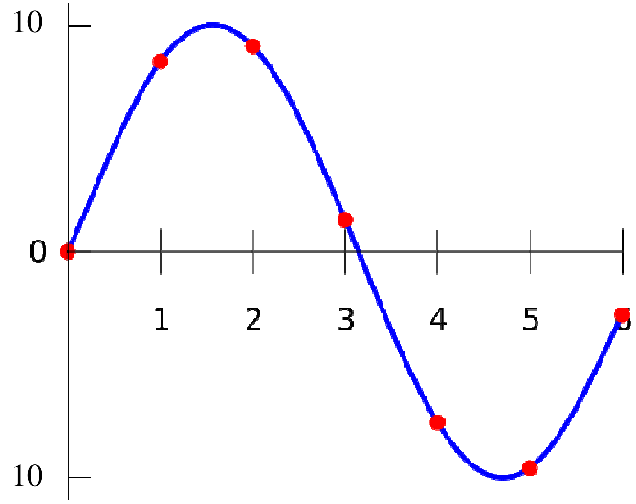
$$\left\lfloor \frac{Z^{d+1}}{Z-x} \right\rfloor = \left\lfloor Z^d \sum_{i=0}^{\infty} (x/Z)^i \right\rfloor = \left\lfloor \sum_{i=0}^{\infty} Z^{(d-i)} x^i \right\rfloor$$

$$\sum_{j=0}^{d} p_j Z^j \; \sum_{i=0}^{d} Z^{(d-i)} x^i = \sum_{i=0}^{d} \left(\sum_{j=0}^{d} p_j Z^{(d-i+j)}\right) x^i$$

$$\left\lfloor \sum_{j=0}^{d} \left(\sum_{i=0}^{d} p_j Z^{(-i+j)}\right) x^i \right\rfloor = \sum_{j=0}^{d} \left(\sum_{0 \le i \le j}^{d} p_j Z^{(-i+j)}\right) x^i$$

$$\sum_{i=0}^{d} p_i x^i$$

Every finite integer sequence $y_0,y_1,\ldots,y_N$ is computable over $\{+,-,\times_c,\text{div}\}$ in constant time independent of the length of the sequence.



Proof:
Interpolation polynomial $P \in \mathbb{Q}[X]$ of degree $\leq N+1$ with $P(n)=y_n$, $n\in\{0,\ldots,N\}$
Take $M \in \mathbb{N}$ such that $M\cdot P \in \mathbb{Z}[X]$,
calculate $M\cdot P(n)$ div M in $\mathcal{O}(1)$.

Every finite language $L \subset \mathbb{Z}$ is decidable over $\{+,-,\times_c, \text{div}\}$ within constant time independent of L.

Proof:
Let $L\subseteq \{0,1,\ldots, N\}$, decide the characteristic sequence $y_0,y_1,\ldots,y_N$ of L with $y_n :=1$ for $n \in L$ and $y_n :=0$ for $n \notin L$ in $\mathcal{O}(1)$.

A polynomial $P \in \mathbb{Z}[x_1,\ldots,x_n]$ can be evaluated on a finite domain D in $\mathbb{N}^n$ over $\{+,-,\times,\mathrm{div}\}$ in time $\mathcal{O}(n)$ independent of p and D.

Proof:

$$(Z^{d^2} \operatorname{div} (Z^d - x_2)) \cdot (Z^d \operatorname{div} (Z - x_1)) = \sum_{i_1,i_2=0}^{d} Z^{d^2-1-(di_2+i_1)} \cdot x_2^{i_2} \cdot x_1^{i_1}$$

Inductively using $\mathcal{O}(n)$ operations from $\{+;-;\times;\ \mathrm{div}\}$

$$\sum_{i_1,\ldots i_n=0}^{d} Z^{d^n-1-(d^{n-1}i_n+\ldots+di_2+i_1)} \cdot x_n^{i_n} \cdot \ldots \cdot x_2^{i_2} \cdot x_1^{i_1}$$

Multiply with the constant $P(Z, Z^d, Z^{d^2}, \ldots, Z^{d^{n-1}})$.

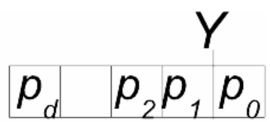
Extract the term corresponding to Z^{d^n-1}

A polynomial $P \in \mathbb{Z}[X]$ over $\mathbb{N}$ can be evaluated using $\mathcal{O}(\log d)$ operations over $\{+,-,\times,\text{div},\&\}$.

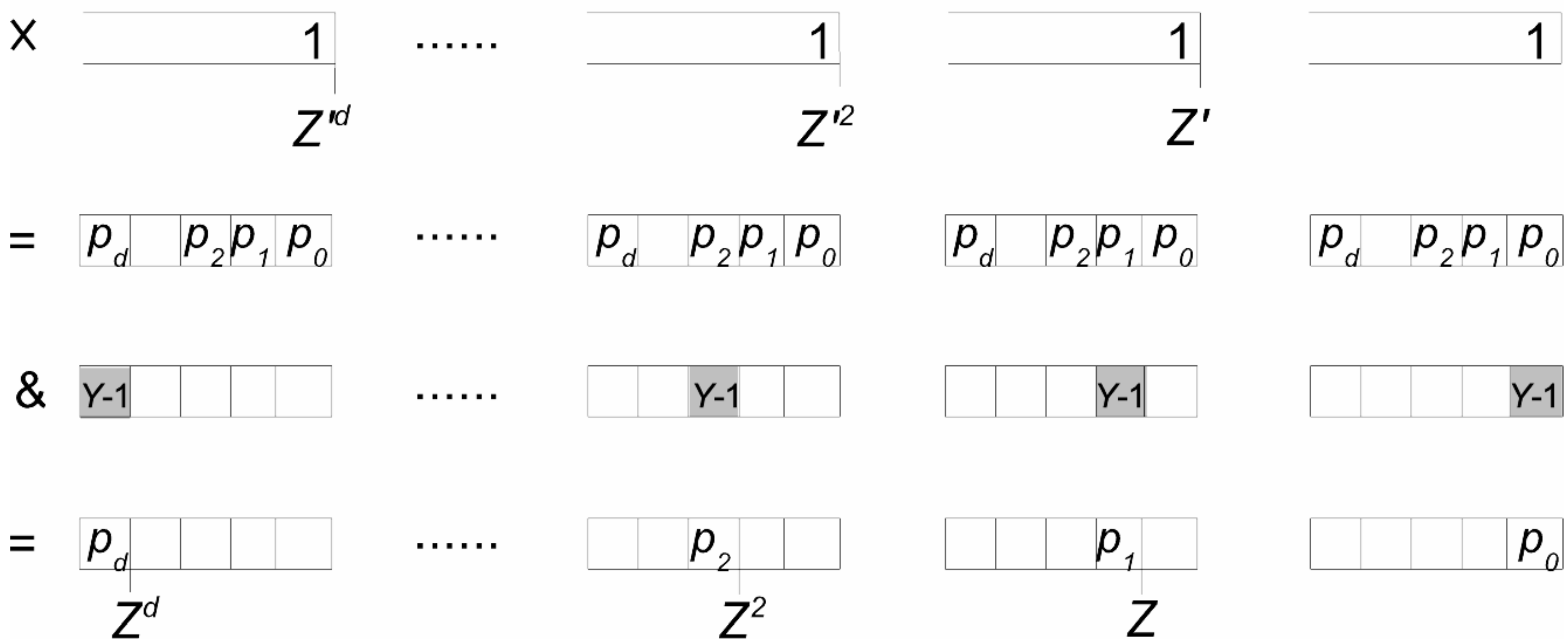
$P(Y)$, $\quad Z' := x^{d+2} = 2^{(d+2)\cdot\log x}$

$W := \sum_{i=0}^{d} Z'^{i} = Z'^{d+1} \text{div}(Z'-1) \qquad = (2^{(d+2)(d+1)})^{\cdot\log x} \text{div}(Z'-1)$

$V := \sum_{i=0}^{d} Z^{i} = \sum_{i=0}^{d} (Z'Y)^{i} = (Z'Y)^{d+1} \text{div}(Z'Y-1)$



$P(Z) = ((P(Y)\cdot W) \& ((Y-1)\cdot V)$



HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Univariate case

Polynomials over $\mathbb{N}$ can be evaluated using $\mathcal{O}(\log \log|x|)$ operations over $\{+,-,\times_c,\mathrm{div},\&\}$.

Polynomials over $\mathbb{N}$ can be evaluated using $\mathcal{O}(\min\{\log \log|x|,\log d\})$ operations over $\{+,-,\times,\mathrm{div},\&\}$.

Polynomials over $\mathbb{N}$ can be evaluated using $\mathcal{O}(\sqrt{\min\{\log \log|x|,\log d\}})$ operations over $\{+,-,\times,\mathrm{div},\&\}$ if in addition some arbitrary integer $y>|x|^{d^2}$ is given.

Multivariate case

Polynomials $P \in \mathbb{Z}[x_1,\ldots,x_n]$ of maximum degree less than d can be evaluated over $\mathbb{Z}^n$ using $\mathcal{O}(n \cdot \min\{\log d, \log\log \max_i|x_i|\})$ operations over $\{+,-,\times,\mathrm{div},\&,\leq\}$.

Polynomials $P \in \mathbb{Z}[x_1,\ldots,x_n]$ of maximum degree less than d can be evaluated over $\mathbb{Z}^n$ using $\mathcal{O}(n \cdot \sqrt{\min\{\log d, \log\log \max_i|x_i|\}})$ operations over $\{+,-,\times_c,\mathrm{div},\&,\leq\}$ if in addition some arbitrary integer $y > (\max_i|x_i|)^{dn+1}$ is given.

Storing and extracting algebraic numbers

$Z_n := Y \cdot 2^n$ with $Y = 2^k > \sum_{i=0}^{d} |p_i|$

$P(Z_n) < Z_n^d \cdot \sum_{i=0}^{d} |p_i| \leq 2^{K+dn}, n \in \mathbb{N}, K := k(d+1)$

$$\rho_p := \sum_n P(Z_n) \cdot 2^{-n(K+dn)}$$

Let $p \in \mathbb{N}[X]$ be of degree $<d$ and suppose that $\sum_n 2^{-dn^2}$ is algebraic of degree $<\delta$. Then $p(x)$ can be calculated over $\{+,-,\times,\mathrm{div}\}$ using $\mathcal{O}(\delta \cdot \log\log x)$ steps.

Lemma

For $\alpha \in \mathbb{R}$ algebraic of degree $<\delta$. Then, given $n \in \mathbb{N}$, one can calculate $u,v \in \mathbb{N}$ such that $|\alpha - u/v| \leq 2^{-n}$ using $\mathcal{O}(\delta \cdot \log n)$ operations over $\{+,-,\times\}$.

Matrix Multiplication

Theorem:

Given $A \in \mathbb{Z}^{n \times n}$ and $B \in \mathbb{Z}^{n \times n}$, one can compute $C := AB \in \mathbb{Z}^{n \times n}$ over {+,-,×,div} using $\mathcal{O}(n^2)$ steps.

Proof:

To compute $c_{i,j} = \sum_{l=1}^{n} a_{i,l} \cdot b_{l,j}$, $i=1,\dots,n, j=1,\dots,n$

Let $Z > (\max_{i,l} a_{i,l}) \cdot (\max_{l,j} b_{lj}) \cdot n$

$\alpha := \sum_{i=1}^{n} \sum_{l=1}^{n} a_{i,l} \cdot Z^{(l-1)+2n^2(i-1)}$ $\quad \beta := \sum_{l=1}^{n} \sum_{j=1}^{n} b_{l,j} \cdot Z^{(n-l)+2n(j-1)}$

$a_{2n}\dots a_{22}a_{21}$			....			$a_{1n}\dots a_{12}a_{11}$

Z^{2n^2} $\quad Z^{2n(n-1)}$ $\quad Z^{2n}$ $\quad Z^{n}$ $\quad Z$

		$b_{1n}b_{2n}\dots b_{nn}$	....	$b_{12}b_{22}\dots b_{n2}$		$b_{11}b_{21}\dots b_{n1}$

$Z^{2n^2-(n-1)}$ $\quad Z^{2n(n-1)+(n-1)}$ $\quad Z^{3n-1}$ $\quad Z^{n-1}$

*	c_{21} * … *	* *	c_{1n} * … *	....	c_{12} * … *		c_{11} * … *

$\gamma := \alpha \cdot \beta$ $\quad c_{i,j}$ are at position $Z^{2n(j-1)+(n-1)+2n^2(i-1)}$

Determinant and Permanent

Fact Given $A \in \mathbb{N}^{n \times n}$ one can calculate

$\mathrm{perm}(A) = \sum_{\pi \in S_n} a_{1\pi(1)} \ldots a_{n\,\pi(n)}$ over {+,-,×,div} in $\mathcal{O}(n^2)$ steps.

Theorem Given $A \in \mathbb{Z}^{n \times n}$ one can calculate

$\det(A) = \sum_{\pi \in S_n} \mathrm{sgn}(\pi) a_{1\,\pi(1)} \ldots a_{n\,\pi(n)}$ over {+,-,×,div} in $\mathcal{O}(n^2)$ steps.

Proof (sketch)

$\det_+(A) = \sum_{\pi \in S_n,\ \mathrm{sgn}(\pi)=+} a_{1\,\pi(1)} \ldots a_{n\,\pi(n)}$ und $\det_-(A) = \sum_{\pi \in S_n,\ \mathrm{sgn}(\pi)=-} a_{1\,\pi(1)} \ldots a_{n\,\pi(n)}$

$\mathrm{perm}(A) =: \det_+(A) + \det_-(A)$; $\det_+ = (\mathrm{perm}+\det)/2$; $\det_- = (\mathrm{perm}-\det)/2$

$\det_+(A)$ and $\det_-(A)$ are polynomials in n^2 variables $x_{i-1+n(j-1)} := a_{ij}$ with coefficients 0,1 of maximum degree 1

$\det_+(Z', Z'^2, \ldots, Z'^{2^{n^2-1}})$ $\det_-(Z', Z'^2, \ldots, Z'^{2^{n^2-1}})$ with $Z' > (\max_i |x_i|)^{2^{n^2+1}}$

Given $A \in \mathbb{Z}^{n \times n}$ one can calculate $\det(A) = \sum_{\pi \in S_n} \operatorname{sgn}(\pi) a_{1\,\pi(1)} \cdots a_{n\,\pi(n)}$ over {+,-,×,div} in $\mathcal{O}(n^2)$ steps.

$\mathrm{Det}(Z'; Z'^2; Z'^{2^2}; \ldots; Z'^{2^{n^2-1}}) =$

$$\begin{vmatrix} Z' & Z'^2 & Z'^4 & Z'^8 \cdots & Z'^{2^{n-1}} \\ Z'^{2^n} & Z'^{2^{n+1}} & Z'^{2^{n+2}} & \cdots & Z'^{2^{2n-1}} \\ Z'^{2^{2n}} & Z'^{2^{2n+1}} & \ddots & & Z'^{2^{3n-1}} \\ Z'^{2^{3n}} & \ddots & & & Z'^{2^{4n-1}} \\ \vdots & & & & \vdots \\ Z'^{2^{(n-1)n}} & \cdots & & \cdots & Z'^{2^{n^2-1}} \end{vmatrix} =$$

Vandermonde matrix

$$= \begin{vmatrix} Z' & Z'^2 & Z'^4 & Z'^8 & \cdots & Z'^{2^{n-1}} \\ Z'^{2^n} & \left(Z'^{2^n}\right)^2 & \left(Z'^{2^n}\right)^4 & \left(Z'^{2^n}\right)^8 & & \left(Z'^{2^n}\right)^{2^{n-1}} \\ Z'^{2^{2n}} & \left(Z'^{2^{2n}}\right)^2 & \left(Z'^{2^{2n}}\right)^4 & \left(Z'^{2^{2n}}\right)^8 & & \left(Z'^{2^{2n}}\right)^{2^{n-1}} \\ Z'^{2^{3n}} & \left(Z'^{2^{3n}}\right)^2 & \ddots & & & \left(Z'^{2^{3n}}\right)^{2^{n-1}} \\ \vdots & & & & & \vdots \\ Z'^{2^{(n-1)n}} & \left(Z'^{2^{(n-1)n}}\right)^2 & \cdots & & \cdots & \left(Z'^{2^{(n-1)n}}\right)^{2^{n-1}} \end{vmatrix} =$$

$$= Z' \cdot Z'^{2^n} \cdot Z'^{2^{2n}} \cdots Z'^{2^{(n-1)n}} \cdot \prod_{1 \le i < j \le n} \left(Z'^{2^{(j-1)n}} - Z'^{2^{(i-1)n}} \right)$$

Matrix Powering

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Theorem

Given $k \in \mathbb{N}$, $A, B \in \mathbb{N}^{d \times d}$, $r := d^{2^k - 1}(\max_{ij} a_{ij})$ such that for all C with $0 \le c_{ij} < r$ $\gcd(B-C) > r$, one can compute A^{2^k} using $\mathcal{O}(d^2\sqrt{k})$ operations over $\{+,-,\times,\text{div},\gcd\}$.

Corollary

Given $k \in \mathbb{N}$, $A, B \in \mathbb{N}^{d \times d}$, $r := d^{2^k - 1}(\max_{ij} a_{ij})$ such that for all C with $0 \le c_{ij} < r$ $\gcd(B-C) > r$, one can compute $A'^{2^{k'}}$ using $\mathcal{O}(d^2\sqrt{k'})$ operations over $\{+,-,\times,\text{div},\gcd\}$ for any $0 \le k' \le k$; $0 \le a'_{ij} \le a_{ij}$.

Outlook

Is there an upper bound for $\{+;-;\times;\text{div}\}$ to compute a polynomial less than the degree?

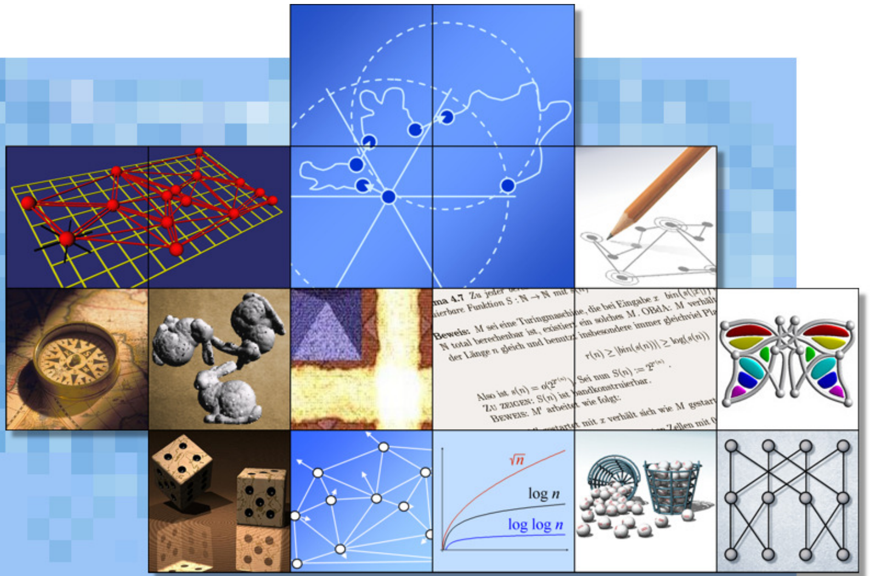
Are there further algorithms in $\mathbb{Z}^n$ which can be accelerated by using integer division or other non-arithmetic primitives?

It is known that an algorithm over $\{+;-;\times;\text{div}\}$ can be simulated in polynomial time over $\{+;-;\times;\&\}$. Is this also true vice versa ? Since polynomially steps over $\{+;-;\times;\text{div}\}$ cover NP and over $\{+;-;\times;\&\}$ PSPACE

$(NP \overset{?}{=} PSPACE)$

(A.Schönhage, On the Power of Random Access Machines, Automata,Languages and Programming, 6th Colloquium 1979)

Thanks for the attention !



Heinz Nixdorf Institute
& Computer Science Institute
University of Paderborn
Fürstenallee 11
33102 Paderborn, Germany

Fax: +49 (0) 52 51/62 64 82
http://www.upb.de/cs/ag-madh

Is the number $x \in \mathbb{N}$ even ?

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

It can be decided in $\mathcal{O}$**(log log x)** steps over {+,-,×, ≤ }.

It can be decided in **4** steps over {+,-,×,**div**, ≤ } .

n mod 2 = n - 2·(n div 2) = 0

It can be decided in **3** steps over {+,-,**&**, ≤ }.

n	1	1	0	0	1	1	0	0	1	1
& (n-1)	1	1	0	0	1	1	0	0	1	0
=	0	0	0	0	0	0	0	0	0	1

Is the number n $\in \mathbb{N}$ even ?

It can be decided in $\mathcal{O}$**(log n)** steps over $\{+,-,\times,\leq\}$.

It can be decided in **4** steps over $\{+,-,\times,$**div**$,\leq\}$.

n mod 2 = n - 2·(n div 2) = 0

It can be decided in **3** steps over $\{+,-,$**&**$,\leq\}$.

n	1	1	0	0	1	1	1	0	0	0
& (n-1)	1	1	0	0	1	1	0	1	1	1
=	0	0	0	0	0	0	0	1	1	1

$1 \leq n \,\&\, (n-1)$

A polynomial $P \in \mathbb{Z}[x_1,\ldots,x_n]$ can be evaluated on a finite domain D in $\mathbb{Z}^n$ over {+,-, ×,div,≤} in time $\mathcal{O}(n)$ independent of p and D.

Proof:

2^n separate algorithms for each polynomial $P(\pm x_1,\ldots, \pm x_n)$.

For $x \in \mathbb{Z}^n$ determine in $\mathcal{O}(n)$ which polynomial to evaluate at $(|x_1|,\ldots, |x_n|)$.

Separate each polynomial in the difference of 2 polynomials with positive coefficients.

$$(Z^{d^2} \operatorname{div} (Z^d - x_2)) \cdot (Z^d \operatorname{div} (Z - x_1)) = \sum_{i_1,i_2=0}^{d} Z^{d^2-1-(di_2+i_1)} \cdot x_2^{i_2} \cdot x_1^{i_1}$$

Inductively using $\mathcal{O}(n)$ operations from {+;-;×; div}

$$\sum_{i_1,\ldots i_n=0}^{d} Z^{d^n-1-(d^{n-1}i_n+\ldots+di_2+i_1)} \cdot x_n^{i_n} \cdot \ldots \cdot x_2^{i_2} \cdot x_1^{i_1}$$

Multiply with the constant $P(Z,Z^d, Z^{d^2},\ldots, Z^{d^{n-1}})$.

Extract the term corresponding to Z^{d^n-1}

Locally Lower-Bounding the GCD

Lemma

For all $d,r,s \in \mathbb{N}$ there exist $x_1,\ldots,x_d \in \mathbb{N}$ such that, for all $u_1,\ldots,u_d \in \{0,1,\ldots,s-1\}$, it holds $\gcd(x_1+u_1,\ldots,x_d+u_d) \geq r$.

Proof

$p_{\bar{u}} > r$ pairwise coprime, $\bar{u} \in \{0,1,\ldots,s-1\}^d$

$u_{i,j} := \prod_{\bar{u}:\, u_i = j} p_{\bar{u}},\ i=1,\ldots,d;\ j=0,1,\ldots,s-1$

$u_{i,0}, u_{i,1},\ldots, u_{i,s-1}$ pairwise coprime

Chinese Remainder Theorem: $\exists\, x_i \in \mathbb{N}$ with $u_{i,j} \mid x_i + j,\ j=1,\ldots,s-1$

$p_{\bar{u}} \mid x_i + u_i,\ i=1,\ldots,d$

$\Rightarrow p_{\bar{u}} \mid \gcd(x_1+u_1,\ldots,x_d+u_d)$

$\Rightarrow \gcd(x_1+u_1,\ldots,x_d+u_d) \geq p_{\bar{u}} \geq r$

Chinese Remainder

Given $a_1,\dots,a_n \in \mathbb{N}$ and pairwise coprime $m_1,\dots,m_n \in \mathbb{N}$ one can calculate $x \in \mathbb{N}$ with $x \equiv a_i \bmod m_i$, $i=1,\dots,n$ in $\mathcal{O}(\log n \cdot \sum_i \log m_i)$ over $\{+,-,\times,\mathrm{div}\}$ and in $\mathcal{O}(n)$ steps over $\{+,-,\times,\mathrm{div},\mathrm{gcdex}\}$

Proof

$N := m_1 \cdot \ldots \cdot m_n$, $1=\gcd(m_i,N/m_i)=s_i m_i + t_i N/m_i$ $e_i = t_i N/m_i$, $i=1,\dots,n$,

$e_i \equiv 1 \bmod m_i$ and $e_i \equiv 0 \bmod m_j$ for $i \neq j$

$x := \sum_i e_i \cdot a_i$

$\gcd(m_i,N/m_i)$ within $\mathcal{O}(\log N) := \mathcal{O}(\sum_i \log m_i)$ for $i=1,\dots,n$

i.e. $\mathcal{O}(n \cdot \sum_i \log m_i)$ over $\{+,-,\times,\mathrm{div}\}$ and $\mathcal{O}(n)$ over $\{+,-,\times,\mathrm{div},\mathrm{gcdex}\}$

congruences y_j with $y \equiv a_{2j} \bmod m_{2j}$ and $y \equiv a_{2j+1} \bmod m_{2j+1}$, $j=1,\dots,n/2$

$n/4$ quadruples $x \equiv y_{2j} \bmod m_{4j} \cdot m_{4j+1}$ and $x \equiv y_{2j+1} \bmod m_{4j+2} \cdot m_{4j+3}$,

$n/2^k$ k-tuples of congruences with disjoint k-tuples of $m_1,\dots,m_n$

$\mathcal{O}(\sum_i \log m_i)$ for $k=1,\dots,\mathcal{O}(\log n)$ i.e. $\mathcal{O}(\log n \cdot \sum_i \log m_i)$

Lemma

For all d,r,s ∈ ℕ there exist $x_1,\ldots,x_d \in \mathbb{N}$ such that, for all $u_1,\ldots,u_d \in \{0,1,\ldots,s-1\}$, it holds $\gcd(x_1+u_1,\ldots,x_d+u_d) \geq r$.

a) $x_1,\ldots,x_d$ can be chosen between 0 and $\mathcal{O}(r \cdot S)^{\mathcal{O}(S)}$ with $S := s^d$.

b) Over {+,-,×,div,gcdex} $x_1,\ldots,x_d$ can be constructed in $\mathcal{O}(S)$.

Proof

a) k_r-th prime p_{k_r} $\mathcal{O}(k \cdot \log k)$ and $\pi(n) \leq \mathcal{O}(n/\log n)$ primes less n

$k_r \leq \mathcal{O}(r/\log r)$ $N := p_{k_r} \cdot \ldots \cdot p_{k_r+S}$

$(r+l)\#/r\#$ with $r+l = p_{k_r+S} = r + (S \cdot \log S)$

$\pi(r+l) - \pi(r) \leq 2\,\pi(l)$

i.e.at most $\mathcal{O}(l/\log l) = \mathcal{O}(S)$ primes between r and r+l

$\Rightarrow (r+l)\#/r\# \leq (r+l)^{\mathcal{O}(l/\log l)} \leq (r \cdot l)^{\mathcal{O}(l/\log l)}$ for $l = \mathcal{O}(S \cdot \log S)$

b) $p_1 := r$, $p_2 := r+1$, $p_3 := p_1 \cdot p_2 + 1, \ldots, p_{i+1} := p_1 \cdot \ldots \cdot p_i + 1$

Constructing Primes Using Integer Division

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Mill's constant θ≈1,3067… yields a sequence of primes $p_n := \lfloor \theta^{3^n} \rfloor$ with $p_{n+1} > p_n^3$.

If θ is rational, one can obtain $p_n := \lfloor \theta^{3^n} \rfloor > 3^n =: N$ over {+,-, ×,div} in $\mathcal{O}(n) = \mathcal{O}(\log N)$.

$$(\theta+\varepsilon)^N = \theta^N + \underbrace{N\cdot\varepsilon\cdot\theta^{N-1} + \sum_{k=2}^{N}\binom{N}{k}\cdot\varepsilon^k\cdot\theta^{N-k}}_{<1}$$

If θ is algebraic, a rational approximation θ' of θ up to error $\varepsilon \approx 2^{-N}/N$ in time $\mathcal{O}(\log N)$ suffices.

Matrix Powering

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Theorem

Given $k \in \mathbb{N}$, $A, B \in \mathbb{N}^{d \times d}$, $r := d^{2^k - 1}(\max_{ij} a_{ij})$ such that for all C with $0 \leq c_{ij} < r$ $\gcd(B-C) > r$, one can compute A^{2^k} using $\mathcal{O}(d^2\sqrt{k})$ operations over $\{+,-,\times,\text{div},\gcd\}$.

Definition

For $X, C \in \mathbb{Z}^{d \times d}$ let $\gcd(C) := \gcd(c_{ij}; 1 \leq i,j \leq d)$

$X \text{ rem } C := (x_{ij} \text{ rem } \gcd(C))$

$X \equiv Y \pmod{C}$, if $\gcd(C) \mid x_{ij} - y_{ij}$ for each entry from $X - Y$.

Lemma

a) If $X \equiv Y \pmod{C}$, then $S \cdot X \cdot T \equiv S \cdot Y \cdot T \pmod{C}$.

b) For each $n \in \mathbb{N}$ it holds $X^n \equiv Y^n \pmod{X-Y}$.

c) $X \text{ rem } C \equiv X \pmod{C}$.

d) If $0 \leq x_{ij} < \gcd(C)$ then $X \text{ rem } C = X$

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Theorem

Given $k \in \mathbb{N}$, $A, B \in \mathbb{N}^{d \times d}$ such that for all C with $0 \leq c_{ij} < d^{2^k-1} (\max_{ij} a_{ij}) =: r$ $\gcd(B-C)>r$, one can compute A^{2^k} using $\mathcal{O}(d^2\sqrt{k})$ operations over $\{+,-,\times,\mathrm{div},\gcd\}$.

Definition

For $X,C \in \mathbb{Z}^{d \times d}$ let $\gcd(C):=\gcd(c_{ij};1 \leq i,j \leq d)$

$X \text{ rem } C := (x_{ij} \text{ rem } \gcd(C))$

$X \equiv Y \pmod C$, if $\gcd(C) \mid x_{ij} - y_{ij}$ for each entry from $X - Y$.

Lemma

a) If $X \equiv Y \pmod C$, then $S \cdot X \cdot T \equiv S \cdot Y \cdot T \pmod C$.

b) For each $n \in \mathbb{N}$ it holds $X^n \equiv Y^n \pmod{X-Y}$.

c) $X \text{ rem } C \equiv X \pmod C$.

d) If $0 \leq x_{ij} < \gcd(C)$ then $X \text{ rem } C = X$

Proof (sketch)

$k:=l^2$, $X:=A^{2^{l(j-1)}}$, $Y:=B^{2^l}$, $C:=Y-X$

$A^{2^{lj}} = (A^{2^{l(j-1)}})^{2^l} = B^{2^l} \text{ rem } (B - A^{2^{l(j-1)}})$, *

$k=l^2$ B^{2^l} in $\mathcal{O}(d^2 \cdot l)$

Inductively for $j=1,\ldots,l$ compute $A^{2^{lj}}$ from $A^{2^{l(j-1)}}$ according to equation *

$\gcd(B - A^{2^{l(j-1)}})$ is computed with the binary gcd in $\mathcal{O}(d^2)$ steps.

Every finite integer sequence S={ $\bar{u}_0$, $\bar{u}_1$,…, $\bar{u}_N$} in $\mathbb{Z}^d$ is computable over {+,-, $\times_c$,div} in $\mathcal{O}$(d) independent of the length of the sequence.

Proof:

τ: $\mathbb{N}^d \rightarrow \mathbb{N}$, $(x_1,\ldots,x_d) \rightarrow x_1 + x_2T_2 + \ldots + T_d x_d$ mit $T_i \in \mathbb{N}$,
such that $\tau \mid S$ is bijectiv. $\tau(S) \subset \mathbb{N}$ is finite
$\Rightarrow$ $\tau(S)$ is computable over $\{+,-,*_c,\text{div}\}$ in constant time.

Every finite language $L \subset \mathbb{Z}^d$ is decidable over {+,-,$\times_c$,div} within $\mathcal{O}$(d) independent of L.

Proof:

Let $\tau(L) \subseteq \{0,1,\ldots, N\}$,such that $\tau \mid L$ is bijective, decide the characteristic sequence $y_0,y_1,\ldots,y_N$ of $\tau(L)$ with $y_n := 1$ for $n \in \tau(L)$ and $y_n := 0$ for $n \notin \tau(L)$ in $\mathcal{O}(1)$.