Faster factorization into coprimes

D. J. Bernstein

Finding multiplicative relations

Define $q = 1000003$;
$u_1 = x^2 + 690277x + 961618$;
$u_2 = x^2 + 532806x + 661735$;
$u_3 = x^2 + 874868x + 419951$;
$u_4 = x^3 + 548974x^2 + 43298x +$
899386; $u_5 = x^7 + 587463x^6 +$
$66890x^5 + 25045x^4 + 886824x^3 +$
$439217x^2 + 28014x + 191136$.

Does $u_1^{1952681} u_2^{1513335} u_3^{634643}$
equal $u_4^{1708632} u_5^{439346}$ in $\mathbf{F}_q[x]$?

Which $(a, b, c, d, e) \in \mathbf{Z}^5$ have
$u_1^a u_2^b u_3^c u_4^d u_5^e = 1$ in $\mathbf{F}_q(x)$?

Factor into primes of $\mathbf{F}_q[x]$:

$u_1 = p_1 p_2$ where $p_1 = x - 325894$ and $p_2 = x - 983835$;

$u_2 = p_1 p_3$ where $p_3 = x - 141303$;

$u_3 = p_2 p_3$; $u_4 = p_1 p_2 p_3$;

$u_5 = p_1^4 p_2^2 p_3$.

Now $u_1^a u_2^b u_3^c u_4^d u_5^e = p_1^{a+b+d+4e} p_2^{a+c+d+2e} p_3^{b+c+d+e}$; and p_1, p_2, p_3 are distinct primes.

$u_1^a u_2^b u_3^c u_4^d u_5^e = 1 \Leftrightarrow$

$p_1^{a+b+d+4e} \ldots = 1 \Leftrightarrow$

$(a + b + d + 4e, \ldots) = 0 \Leftrightarrow$

$(a, b, c, d, e) \in$

$(1, 1, 1, -2, 0)\mathbf{Z} + (3, 2, 0, -1, -1)\mathbf{Z}$.

Primality was overkill here.

All we needed was **coprimality**:

$\gcd\{p_1, p_2\} = 1$;

$\gcd\{p_1, p_3\} = 1$;

$\gcd\{p_2, p_3\} = 1$.

Many applications of
factorization into primes
can instead use any
factorization into coprimes.

In particular, this application:

Coprimes p_1, p_2, p_3 have
$p_1^{a_1} p_2^{a_2} p_3^{a_3} = 1$ iff $(a_1, a_2, a_3) = 0$.

Finding multiplicative relations
is the critical bottleneck in
modern "index-calculus" methods
to compute "discrete logarithms"
by "combining congruences":
e.g., the "function-field sieve."

In simplest methods, can "sieve"
to efficiently find prime factors.

Advanced methods can't sieve.
Standard backup: qth powering
to factor into primes.
(For integers: ECM etc.)
As before, this is overkill;
adequate to factor into coprimes.

How fast is factorization?

Obvious algorithm to factor monic $u_1, u_2, \ldots \in k[x]$ into coprimes:
divide out gcd's until coprime.
Algorithm is algebraic over k;
$O(n^3)$ ops for n input coeffs.

1990 Bach Driscoll Shallit:
avoid pointless gcd's; $O(n^2)$.

1995 Bernstein: $n^{1+o(1)}$;
more precisely, $n(\lg n)^{O(1)}$.

2004 Bernstein: $n(\lg n)^{4+o(1)}$;
$O(n(\lg n)^4(\lg \lg n)^2)$.

The **natural coprime base** $\mathrm{cb}\, S$
for a set S
is the unique set P such that

- P can be obtained from
  $S \cup \{1\}$ via product,
  exact quotient, gcd;
- S can be obtained from
  $P \dot\cup \{1\}$ via product; and
- P is coprime: $\gcd\{a, b\} = 1$
  for all distinct $a, b \in P$.

All of the above algorithms
compute $\mathrm{cb}\{u_1, u_2, \ldots\}$.

(Can replace $k[x]$ by $\mathbf{Z}$.)

Factor into primes? Abandon cb?

1995 Kaltofen Shoup for $k = \mathbf{F}_q$:
factor into primes using
$O(n^{1.815} \lg q)$ operations in k.
For large n, faster than
1990 Bach Driscoll Shallit,
but slower than 1995 Bernstein.

Slight improvements later,
but every known method to
factor into primes is slower than
factoring into coprimes.
Look for applications that
(1) factor into primes but
(2) can use arbitrary coprimes.

Some algorithms for
factorization into primes
use, as a subroutine,
factorization into coprimes.

e.g. Given squarefree $g \in \mathbf{F}_2[x]$:
Find basis $h_1, h_2, \ldots$
for $\left\{h \in \mathbf{F}_2[x] : (gh)' = h^2\right\}$
as a vector space over $\mathbf{F}_2$.
Then cb$\{g, h_1, h_2, \ldots\}$ contains
all prime divisors of g.

(1993 Niederreiter, 1994 Göttfert)

More examples, applications of factoring into coprimes: see 1890 Stieltjes; 1974 Collins; 1985 Kaltofen; 1985 Della Dora DiCrescenzo Duval; 1986 Bach Miller Shallit; 1986 von zur Gathen; 1986 Lüneburg; 1989 Pohst Zassenhaus; 1990 Teitelbaum; 1990 Smedley; 1993 Bach Driscoll Shallit; 1994 Ge; 1994 Buchmann Lenstra; 1996 Bernstein; 1997 Silverman; 1998 Cohen Diaz y Diaz Olivier; 1998 Storjohann; ...

cr.yp.to/coprimes.html

How to compute cb

$n(\lg n)^{1+o(1)}$: $a, b \mapsto ab$
if a, b together have n coeffs.

$n(\lg n)^{1+o(1)}$: $a, b \mapsto \lfloor a/b \rfloor$.

$n(\lg n)^{1+o(1)}$: $a, b \mapsto a \bmod b$.

$n(\lg n)^{2+o(1)}$: $a, b \mapsto \gcd\{a, b\}$.

$n(\lg n)^{1+o(1)} + m(\lg m)^{2+o(1)}$:
$a, b \mapsto \gcd\{a, b\}$
if b has m coeffs.

$n(\lg n)^{2+o(1)}$: $a,b \mapsto \gcd\{a, b^\infty\}$.

See cr.yp.to/papers.html#multapps for a fast-mult survey.

$n(\lg n)^{2+o(1)}$: $a, b \mapsto \mathrm{cb}\{a, b\}$
by the following algorithm.

Permute (a, b) so $\deg a \geq \deg b$.
Compute successively
$a_0 = a$; $g_0 = \gcd\{a_0, b\}$;
$a_1 = a_0/g_0$; $g_1 = \gcd\{a_1, g_0^2\}$;
$a_2 = a_1/g_1$; $g_2 = \gcd\{a_2, g_1^2\}$;
etc. Stop when $g_k = 1$.

Compute
$x_0 = g_0/\gcd\{g_0, g_1^\infty\}$;
$x_1 = g_1/\gcd\{g_1, g_2^\infty\}$;
etc.

Compute $b \bmod g_1; b \bmod g_2; \ldots$
using a remainder tree.

Compute

$\gcd\{b, g_1\} = \gcd\{b \bmod g_1, g_1\}$;

$\gcd\{b, g_2\} = \gcd\{b \bmod g_2, g_2\}$;

etc.

Compute $y_0 = \gcd\{b, x_0^\infty\}$;

$y_1 = \gcd\{g_0, x_1^\infty\}$;

$y_2 = \gcd\{\gcd\{b, g_1\}, x_2^\infty\}$;

$y_3 = \gcd\{\gcd\{b, g_2\}, x_3^\infty\}$;

$y_4 = \gcd\{\gcd\{b, g_3\}, x_4^\infty\}$;

etc.

Then $\mathrm{cb}\{a, b\}$ is disjoint union of

$\mathrm{cb}\{x_0, y_0/x_0\}$,

$\mathrm{cb}\{x_1, y_1\}, \mathrm{cb}\{x_2, y_2\}, \ldots,$

$\{a_k\} - \{1\}$, $\{b/\gcd\{b, a^\infty\}\} - \{1\}$.

What about $\operatorname{cb} S$ for $\#S \geq 3$?

$n(\lg n)^{2+o(1)}$ if $\lg \#P \in (\lg n)^{o(1)}$:
multiset S, coprime set P
$\mapsto \gcd\{s, p^\infty\}$
for each $s \in S$, each $p \in P$.

$n(\lg n)^{2+o(1)}$:
a, coprime set $Q \mapsto \operatorname{cb}(\{a\} \cup Q)$.

More complicated than the case
$Q = \{b\}$ but same basic ideas.

$n(\lg n)^{3+o(1)}$:
coprime set P, coprime set Q
$\mapsto \operatorname{cb}(P \cup Q)$.

Idea of $\operatorname{cb}(P \cup Q)$ algorithm:

Replace Q with $\operatorname{cb}(\{a\} \cup Q)$

for each $a \in P$ successively.

But that's too slow if $\#P$ is large,

so first replace P with P' having

$\#P' \in O(\lg n)$ and $\operatorname{cb} P' = \operatorname{cb} P$.

e.g. $p_0 p_1 p_4 p_5 p_8 p_9 \cdots \in P'$.

$n(\lg n)^{4+o(1)}$: $S \mapsto \operatorname{cb} S$.

$n(\lg n)^{3+o(1)}$: Factor S over $\operatorname{cb} S$.

cr.yp.to/papers.html#dcba

cr.yp.to/papers.html#dcba2

cr.yp.to/talks.html

#2004.07.07