Seminar zur Kryptographie (SS 2009)

PRIMES is in P
Der AKS-Primzahltest

Florian Weingarten

Betreuer:
Dipl.-Inform. Alexander Skopalik
Lehrstuhl für Informatik 1
Algorithmen und Komplexität
RWTH Aachen

Aachen/Bonn, 5. August 2009

Inhalt

Überblick

Definition

Eine natürliche Zahl $n > 1$ nennt man *prim*, wenn sie keine nichttrivialen Teiler hat, d.h. die einzigen natürlichen Zahlen, die n teilen, sind 1 und n selbst. Anderenfalls nennt man n *zusammengesetzt*.

Definition

Eine natürliche Zahl $n > 1$ nennt man *prim*, wenn sie keine nichttrivialen Teiler hat, d.h. die einzigen natürlichen Zahlen, die n teilen, sind 1 und n selbst. Anderenfalls nennt man n *zusammengesetzt*.

Fundamentalsatz der Arithmetik

Jede natürliche Zahl $n > 1$ lässt sich **eindeutig** (bis auf die Reihenfolge der Faktoren) in ein Produkt von Primzahlen faktorisieren.

Definition

Eine natürliche Zahl $n > 1$ nennt man *prim*, wenn sie keine nichttrivialen Teiler hat, d.h. die einzigen natürlichen Zahlen, die n teilen, sind 1 und n selbst. Anderenfalls nennt man n *zusammengesetzt*.

Fundamentalsatz der Arithmetik

Jede natürliche Zahl $n > 1$ lässt sich **eindeutig** (bis auf die Reihenfolge der Faktoren) in ein Produkt von Primzahlen faktorisieren.

Klar

Teilt eine Primzahl p ein Produkt $a \cdot b$, dann teilt p bereits a oder b.

Definition

Eine natürliche Zahl $n > 1$ nennt man *prim*, wenn sie keine nichttrivialen Teiler hat, d.h. die einzigen natürlichen Zahlen, die n teilen, sind 1 und n selbst. Anderenfalls nennt man n *zusammengesetzt*.

Fundamentalsatz der Arithmetik

Jede natürliche Zahl $n > 1$ lässt sich **eindeutig** (bis auf die Reihenfolge der Faktoren) in ein Produkt von Primzahlen faktorisieren.

Klar

Teilt eine Primzahl p ein Produkt $a \cdot b$, dann teilt p bereits a oder b.

Primzahlproblem

Gegeben $n > 1$. Ist n prim oder zusammengesetzt?
(Äquivalent dazu: Gegeben $n > 1$. Gibt es einen nichttrivialen Teiler von n?)

Definition

Eine natürliche Zahl $n > 1$ nennt man *prim*, wenn sie keine nichttrivialen Teiler hat, d.h. die einzigen natürlichen Zahlen, die n teilen, sind 1 und n selbst. Anderenfalls nennt man n *zusammengesetzt*.

Fundamentalsatz der Arithmetik

Jede natürliche Zahl $n > 1$ lässt sich **eindeutig** (bis auf die Reihenfolge der Faktoren) in ein Produkt von Primzahlen faktorisieren.

Klar

Teilt eine Primzahl p ein Produkt $a \cdot b$, dann teilt p bereits a oder b.

Primzahlproblem

Gegeben $n > 1$. Ist n prim oder zusammengesetzt?
(Äquivalent dazu: Gegeben $n > 1$. Gibt es einen nichttrivialen Teiler von n?)

Faktorisierungsproblem

Gegeben $n > 1$. Bestimme einen nichttrivialen Teiler von n (falls einer existiert).

Frage

Gibt es einen Primzahltest, der gleichzeitig

- effizient ist (polynomiell **in der Eingabelänge**),
- deterministisch ist,
- unabhängig von unbewiesenen Sätzen ist und
- für alle Primzahlen funktioniert?

Frage

Gibt es einen Primzahltest, der gleichzeitig

- effizient ist (polynomiell **in der Eingabelänge**),
- deterministisch ist,
- unabhängig von unbewiesenen Sätzen ist und
- für alle Primzahlen funktioniert?

Komplexitätstheorie

Ist das Primzahltestproblem $\textsc{Primes}$ in der Klasse P?

Frage

Gibt es einen Primzahltest, der gleichzeitig

- effizient ist (polynomiell **in der Eingabelänge**),
- deterministisch ist,
- unabhängig von unbewiesenen Sätzen ist und
- für alle Primzahlen funktioniert?

Komplexitätstheorie

Ist das Primzahltestproblem $\textsc{Primes}$ in der Klasse P?

Antwort (seit 2002)

Ja!

Frage

Gibt es einen Primzahltest, der gleichzeitig

- effizient ist (polynomiell **in der Eingabelänge**),
- deterministisch ist,
- unabhängig von unbewiesenen Sätzen ist und
- für alle Primzahlen funktioniert?

Komplexitätstheorie

Ist das Primzahltestproblem $\textsc{Primes}$ in der Klasse P?

Antwort (seit 2002)

Ja!

Noch unbeantwortet

Ist das Faktorisierungsproblem $\textsc{Factoring}$ in P?

Überblick

Motivation

„The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic. (. . .) Further, the dignity of the science itself seems to require that every possible means be explored for the solution of a problem so elegant and so celebrated.“

– *Carl Friedrich Gauss*

Motivation

„The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic. (. . .) Further, the dignity of the science itself seems to require that every possible means be explored for the solution of a problem so elegant and so celebrated.“

– *Carl Friedrich Gauss*

- Sehr einfache Fragestellung lange unbeantwortet
- Theoretische Motivation
- Komplexitätstheorie
- Kryptographie

Komplexitätstheorie

P NP

Komplexitätstheorie

$$\mathrm{P} \quad \subseteq \quad \mathrm{ZPP} \qquad\qquad\qquad\qquad \mathrm{NP}$$

- ZPP: „Immer korrekt, aber nur im Durchschnitt polynomiell“

Komplexitätstheorie

$$\mathrm{P} \subseteq \mathrm{ZPP} \subseteq \mathrm{RP} \subseteq \mathrm{NP}$$

- ZPP: „Immer korrekt, aber nur im Durchschnitt polynomiell“
- RP: „Polynomiell; *JA* ist immer korrekt, *NEIN* ist meistens korrekt“

Komplexitätstheorie

$$\begin{array}{ccccccc} \mathrm{P} & \subseteq & \mathrm{ZPP} & \subseteq & \mathrm{RP} & \subseteq & \mathrm{NP} \\ & & & \subseteq & \textsc{Co}\text{-}\mathrm{RP} & & \end{array}$$

- ZPP: „Immer korrekt, aber nur im Durchschnitt polynomiell"
- RP: „Polynomiell; *JA* ist immer korrekt, *NEIN* ist meistens korrekt"
- $\mathrm{RP} \cap \textsc{Co}\text{-}\mathrm{RP} = \mathrm{ZPP}$

Komplexitätstheorie

$$\begin{array}{ccccccc} \mathrm{P} & \subseteq & \mathrm{ZPP} & \subseteq & \mathrm{RP} & \subseteq & \mathrm{NP} \\ & & & \subseteq & \textsc{Co}\text{-}\mathrm{RP} & \subseteq & \mathrm{BPP} \end{array}$$

- ZPP: „Immer korrekt, aber nur im Durchschnitt polynomiell“
- RP: „Polynomiell; *JA* ist immer korrekt, *NEIN* ist meistens korrekt“
- $\mathrm{RP} \cap \textsc{Co}\text{-}\mathrm{RP} = \mathrm{ZPP}$
- BPP: „Polynomiell, aber nicht immer korrekt“

Komplexitätstheorie

$$\begin{array}{ccccccc} P & \subseteq & ZPP & \subseteq & RP & \subseteq & NP \\ & & & \subseteq & \text{Co-RP} & \subseteq & BPP \\ & & & & & \subseteq & \text{Co-NP} \end{array}$$

- ZPP: „Immer korrekt, aber nur im Durchschnitt polynomiell“
- RP: „Polynomiell; *JA* ist immer korrekt, *NEIN* ist meistens korrekt“
- $\mathrm{RP} \cap \text{Co-RP} = \mathrm{ZPP}$
- BPP: „Polynomiell, aber nicht immer korrekt“

Komplexitätstheorie

$$\begin{array}{ccccccc} \mathrm{P} & \subseteq & \mathrm{ZPP} & \subseteq & \mathrm{RP} & \subseteq & \mathrm{NP} \\ & & & \subseteq & \textsc{Co}\text{-}\mathrm{RP} & \subseteq & \mathrm{BPP} \\ & & & & & \subseteq & \textsc{Co}\text{-}\mathrm{NP} \end{array}$$

- ZPP: „Immer korrekt, aber nur im Durchschnitt polynomiell“
- RP: „Polynomiell; *JA* ist immer korrekt, *NEIN* ist meistens korrekt“
- $\mathrm{RP} \cap \textsc{Co}\text{-}\mathrm{RP} = \mathrm{ZPP}$
- BPP: „Polynomiell, aber nicht immer korrekt“

- Klar: $\textsc{Primes} \in \textsc{Co}\text{-}\mathrm{NP}$ (227923 ist nicht prim. Beweis: teilbar durch 719)

Komplexitätstheorie

$$
\begin{array}{ccccccc}
\mathrm{P} & \subseteq & \mathrm{ZPP} & \subseteq & \mathrm{RP} & \subseteq & \mathrm{NP} \\
 & & & \subseteq & \textsc{Co}\text{-}\mathrm{RP} & \subseteq & \mathrm{BPP} \\
 & & & & & \subseteq & \textsc{Co}\text{-}\mathrm{NP}
\end{array}
$$

- ZPP: „Immer korrekt, aber nur im Durchschnitt polynomiell“
- RP: „Polynomiell; *JA* ist immer korrekt, *NEIN* ist meistens korrekt“
- $\mathrm{RP} \cap \textsc{Co}\text{-}\mathrm{RP} = \mathrm{ZPP}$
- BPP: „Polynomiell, aber nicht immer korrekt“

- Klar: $\textsc{Primes} \in \textsc{Co}\text{-}\mathrm{NP}$ (227923 ist nicht prim. Beweis: teilbar durch 719)
- 1975, Pratt: $\textsc{Primes} \in \mathrm{NP}$ (227947 ist prim. Beweis?)

Komplexitätstheorie

$$\begin{array}{ccccccc} \mathrm{P} & \subseteq & \mathrm{ZPP} & \subseteq & \mathrm{RP} & \subseteq & \mathrm{NP} \\ & & & \subseteq & \textsc{Co}\text{-}\mathrm{RP} & \subseteq & \mathrm{BPP} \\ & & & & & \subseteq & \textsc{Co}\text{-}\mathrm{NP} \end{array}$$

- ZPP: „Immer korrekt, aber nur im Durchschnitt polynomiell“
- RP: „Polynomiell; *JA* ist immer korrekt, *NEIN* ist meistens korrekt“
- $\mathrm{RP} \cap \textsc{Co}\text{-}\mathrm{RP} = \mathrm{ZPP}$
- BPP: „Polynomiell, aber nicht immer korrekt“

- Klar: $\textsc{Primes} \in \textsc{Co}\text{-}\mathrm{NP}$ (227923 ist nicht prim. Beweis: teilbar durch 719)
- 1975, Pratt: $\textsc{Primes} \in \mathrm{NP}$ (227947 ist prim. Beweis?)
- 1977, Solovay, Strassen: $\textsc{Primes} \in \textsc{Co}\text{-}\mathrm{RP} \subseteq \mathrm{BPP}$

Komplexitätstheorie

$$\begin{array}{ccccccc} \mathrm{P} & \subseteq & \mathrm{ZPP} & \subseteq & \mathrm{RP} & \subseteq & \mathrm{NP} \\ & & & \subseteq & \textsc{Co}\text{-}\mathrm{RP} & \subseteq & \mathrm{BPP} \\ & & & & & \subseteq & \textsc{Co}\text{-}\mathrm{NP} \end{array}$$

- ZPP: „Immer korrekt, aber nur im Durchschnitt polynomiell“
- RP: „Polynomiell; *JA* ist immer korrekt, *NEIN* ist meistens korrekt“
- $\mathrm{RP} \cap \textsc{Co}\text{-}\mathrm{RP} = \mathrm{ZPP}$
- BPP: „Polynomiell, aber nicht immer korrekt“

- Klar: $\textsc{Primes} \in \textsc{Co}\text{-}\mathrm{NP}$ (227923 ist nicht prim. Beweis: teilbar durch 719)
- 1975, Pratt: $\textsc{Primes} \in \mathrm{NP}$ (227947 ist prim. Beweis?)
- 1977, Solovay, Strassen: $\textsc{Primes} \in \textsc{Co}\text{-}\mathrm{RP} \subseteq \mathrm{BPP}$
- 1992, Adleman, Huang: $\textsc{Primes} \in \mathrm{RP}$ (also $\textsc{Primes} \in \mathrm{ZPP}$)

Komplexitätstheorie

$$\begin{array}{ccccccc} \mathrm{P} & \subseteq & \mathrm{ZPP} & \subseteq & \mathrm{RP} & \subseteq & \mathrm{NP} \\ & & & \subseteq & \textsc{Co}\text{-}\mathrm{RP} & \subseteq & \mathrm{BPP} \\ & & & & & \subseteq & \textsc{Co}\text{-}\mathrm{NP} \end{array}$$

- ZPP: „Immer korrekt, aber nur im Durchschnitt polynomiell“
- RP: „Polynomiell; *JA* ist immer korrekt, *NEIN* ist meistens korrekt“
- $\mathrm{RP} \cap \textsc{Co}\text{-}\mathrm{RP} = \mathrm{ZPP}$
- BPP: „Polynomiell, aber nicht immer korrekt“

- Klar: $\textsc{Primes} \in \textsc{Co}\text{-}\mathrm{NP}$ (227923 ist nicht prim. Beweis: teilbar durch 719)
- 1975, Pratt: $\textsc{Primes} \in \mathrm{NP}$ (227947 ist prim. Beweis?)
- 1977, Solovay, Strassen: $\textsc{Primes} \in \textsc{Co}\text{-}\mathrm{RP} \subseteq \mathrm{BPP}$
- 1992, Adleman, Huang: $\textsc{Primes} \in \mathrm{RP}$ (also $\textsc{Primes} \in \mathrm{ZPP}$)
- 2002, Agrawal, Kayal, Saxena: $\textsc{Primes} \in \mathrm{P}$

Überblick

Kryptographische Verfahren

- Primzahlen werden in sehr vielen kryptographischen Verfahren benötigt.
- „Sind diese Verfahren jetzt gebrochen?!“
- Nein!
- Aus effizienten Primzahltests folgt keineswegs effiziente Faktorisierung.

Kryptographische Verfahren

- Primzahlen werden in sehr vielen kryptographischen Verfahren benötigt.
- „Sind diese Verfahren jetzt gebrochen?!“
- Nein!
- Aus effizienten Primzahltests folgt keineswegs effiziente Faktorisierung.

Auswirkungen für die Praxis

- Eigentlich überhaupt keine :-)
- AKS-Algorithmus deutlich langsamer als z.B. Miller-Rabin.

Überblick

Definition

Kleinstes $k \in \mathbb{N}$ mit $a^k \equiv 1 \pmod{n}$ heißt *Ordnung von a modulo n* (kurz $\mathrm{ord}_n(a)$).

Definition

Kleinstes $k \in \mathbb{N}$ mit $a^k \equiv 1 \pmod{n}$ heißt *Ordnung von* a *modulo* n (kurz $\text{ord}_n(a)$).

Binomischer Lehrsatz

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} y^{n-i} x^i$$

Definition

Kleinstes $k \in \mathbb{N}$ mit $a^k \equiv 1 \pmod{n}$ heißt *Ordnung von a modulo n* (kurz $\mathrm{ord}_n(a)$).

Binomischer Lehrsatz

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} y^{n-i} x^i$$

Kleiner Fermat

$$a^p \equiv a \pmod{p}$$

Definition

Kleinstes $k \in \mathbb{N}$ mit $a^k \equiv 1 \pmod{n}$ heißt *Ordnung von a modulo n* (kurz $\text{ord}_n(a)$).

Binomischer Lehrsatz

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} y^{n-i} x^i$$

Kleiner Fermat

$$a^p \equiv a \pmod{p}$$

Folgerung (Fermat-Test)

- *Falls $a^n \not\equiv a \pmod{n}$, dann kann n nicht prim sein.*
- *Problem: Falls $a^n \equiv a \pmod{n}$ doch gilt, dann kann n trotzdem zusammengesetzt sein (Carmichael Zahlen, eulersche Pseudoprimzahlen).*

Überblick

Satz

$a \in \mathbb{Z}$, $n \in \mathbb{N}$, $n > 1$, $\mathrm{ggT}(a, n) = 1$.

$$n \textit{ prim} \quad \Leftrightarrow \quad (x + a)^n = x^n + a \; (\textit{mod } n).$$

Satz

$a \in \mathbb{Z}$, $n \in \mathbb{N}$, $n > 1$, $\mathrm{ggT}(a, n) = 1$.

$$n \textit{ prim} \quad \Leftrightarrow \quad (x+a)^n = x^n + a \ (\textit{mod } n).$$

Beweis.

„$\Rightarrow$“: LA1 / Diskrete Strukturen („Frobenius“)

Satz

$a \in \mathbb{Z}$, $n \in \mathbb{N}$, $n > 1$, $\mathrm{ggT}(a, n) = 1$.

$$n \textit{ prim} \quad \Leftrightarrow \quad (x + a)^n = x^n + a \; (\textit{mod } n).$$

Beweis.

„$\Rightarrow$“: LA1 / Diskrete Strukturen („Frobenius“)
„$\Leftarrow$“:

- Koeffizient von x^i in $(x + a)^n$ ist $\binom{n}{i} a^{n-i}$.
- n zusammengesetzt mit $p | n$.

$$\binom{n}{p} = \frac{n \cdot (n-1) \cdot \ldots \cdot (n-(p-1))}{p \cdot (p-1) \cdot \ldots \cdot 1}$$

- p teilt n im Zähler und p im Nenner, sonst keinen Faktor.
- p^k größte Potenz, die n teilt, dann p^{k-1} größte Potenz, die $\binom{n}{p}$ teilt.
- Also: $\binom{n}{p}$ nicht durch n teilbar.
- $\mathrm{ggT}(a, n) = 1$, also a^{n-i} auch nicht durch n teilbar.

Beispiel

$n = 4$ ist zusammengesetzt:

$$(x+a)^4 = x^4 + 4a^1x^3 + 6a^2x^2 + 4a^3x^1 + a^4$$

$n = 5$ ist prim:

$$(x+a)^5 = x^5 + 5a^1x^4 + 10a^2x^3 + 10a^3x^2 + 5a^4x^1 + a^5$$

Beispiel

$n = 4$ ist zusammengesetzt:

$$(x+a)^4 = x^4 + 4a^1x^3 + 6a^2x^2 + 4a^3x^1 + a^4$$

$n = 5$ ist prim:

$$(x+a)^5 = x^5 + 5a^1x^4 + 10a^2x^3 + 10a^3x^2 + 5a^4x^1 + a^5$$

Beobachtungen

- Kann als Primzahltest benutzt werden.
- Ist aber nicht effizient ($O(n)$ Koeffizienten auswerten).
- **Idee:** Exponenten der Polynome reduzieren.
- **Konkret:** Rechne in $\mathbb{Z}_n[x]/\langle x^r - 1\rangle$ statt in $\mathbb{Z}_n[x]$.
- **Beispiel:** $x^{2000} = (x^3)^{666} \cdot x^2 = x^2$ in $\mathbb{Z}_n[x]/\langle x^3 - 1\rangle$.
- **Klar:** $(x+a)^p = x^p + a \pmod{x^r - 1, p}$.
- **Problem (wie bei Fermat):** Umkehrung gilt nichtmehr.
- **Aber:** Geschickte Wahl von r und „kleine" Schranke für a rettet die Idee.

Überblick

Der AKS-Primzahltest

Eingabe: $n \in \mathbb{N}$ mit $n > 1$.

1. Falls $n = a^b$ für $a, b \in \mathbb{N}$ und $a, b > 1$, gebe ZUSAMMENGESETZT aus.
2. Finde das kleinste r, so dass $\text{ord}_r(n) > \log^2 n$.
3. Falls $1 < \text{ggT}(a, n) < n$ für ein $a \leq r$, gebe ZUSAMMENGESETZT aus.
4. Falls $n \leq r$, gebe PRIM aus.
5. Für $a = 1$ bis $\lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$

   Falls $(x + a)^n \not\equiv x^n + a \pmod{x^r - 1, n}$, gebe ZUSAMMENGESETZT aus.
6. Gebe PRIM aus.

Der AKS-Primzahltest

Eingabe: $n \in \mathbb{N}$ mit $n > 1$.

1. Falls $n = a^b$ für $a, b \in \mathbb{N}$ und $a, b > 1$, gebe ZUSAMMENGESETZT aus.
2. Finde das kleinste r, so dass $\text{ord}_r(n) > \log^2 n$.
3. Falls $1 < \text{ggT}(a, n) < n$ für ein $a \leq r$, gebe ZUSAMMENGESETZT aus.
4. Falls $n \leq r$, gebe PRIM aus.
5. Für $a = 1$ bis $\lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$
   Falls $(x + a)^n \not\equiv x^n + a \pmod{x^r - 1, n}$, gebe ZUSAMMENGESETZT aus.
6. Gebe PRIM aus.

Satz

Ist n prim, so gibt der Algorithmus PRIM *aus.*

Der AKS-Primzahltest

Eingabe: $n \in \mathbb{N}$ mit $n > 1$.

1. Falls $n = a^b$ für $a, b \in \mathbb{N}$ und $a, b > 1$, gebe ZUSAMMENGESETZT aus.
2. Finde das kleinste r, so dass $\text{ord}_r(n) > \log^2 n$.
3. Falls $1 < \text{ggT}(a, n) < n$ für ein $a \le r$, gebe ZUSAMMENGESETZT aus.
4. Falls $n \le r$, gebe PRIM aus.
5. Für $a = 1$ bis $\lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$
   Falls $(x + a)^n \not\equiv x^n + a \pmod{x^r - 1, n}$, gebe ZUSAMMENGESETZT aus.
6. Gebe PRIM aus.

Satz

Ist n prim, so gibt der Algorithmus PRIM *aus.*

Beweis.

1 und 3 können nicht ZUSAMMENGESETZT ausgeben. Nach Satz auch 5 nicht. □

Der AKS-Primzahltest

Eingabe: $n \in \mathbb{N}$ mit $n > 1$.

1. Falls $n = a^b$ für $a, b \in \mathbb{N}$ und $a, b > 1$, gebe ZUSAMMENGESETZT aus.
2. Finde das kleinste r, so dass $\mathrm{ord}_r(n) > \log^2 n$.
3. Falls $1 < \mathrm{ggT}(a, n) < n$ für ein $a \leq r$, gebe ZUSAMMENGESETZT aus.
4. Falls $n \leq r$, gebe PRIM aus.
5. Für $a = 1$ bis $\lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$
   Falls $(x + a)^n \not\equiv x^n + a \pmod{x^r - 1, n}$, gebe ZUSAMMENGESETZT aus.
6. Gebe PRIM aus.

Ziel

Gibt der Algorithmus PRIM aus, so ist n tatsächlich prim.

Der AKS-Primzahltest

Eingabe: $n \in \mathbb{N}$ mit $n > 1$.

1. Falls $n = a^b$ für $a, b \in \mathbb{N}$ und $a, b > 1$, gebe $\textsc{Zusammengesetzt}$ aus.
2. Finde das kleinste r, so dass $\text{ord}_r(n) > \log^2 n$.
3. Falls $1 < \text{ggT}(a, n) < n$ für ein $a \leq r$, gebe $\textsc{Zusammengesetzt}$ aus.
4. Falls $n \leq r$, gebe $\textsc{Prim}$ aus.
5. Für $a = 1$ bis $\lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$
   Falls $(x + a)^n \not\equiv x^n + a \pmod{x^r - 1, n}$, gebe $\textsc{Zusammengesetzt}$ aus.
6. Gebe $\textsc{Prim}$ aus.

Ziel

Gibt der Algorithmus $\textsc{Prim}$ aus, so ist n tatsächlich prim.

Klar

Wenn $\textsc{Prim}$ in 4 ausgegeben wird, so ist n prim.

Überblick

Beobachtung

- $(x+a)^n \equiv x^n + a \pmod{x^r - 1, n}$ (wegen Schritt 5).
- Daraus folgt: $(x+a)^n \equiv x^n + a \pmod{x^r - 1, p}$ (denn p teilt n).
- Da p prim ist, gilt auch: $(x+a)^p \equiv x^p + a \pmod{x^r - 1, p}$.
- Da n durch p teilbar auch: $(x+a)^{\frac{n}{p}} \equiv x^{\frac{n}{p}} + a \pmod{x^r - 1, p}$.
- n und $\frac{n}{p}$ „verhalten sich wie Primzahlen in dieser Gleichung“.

Beweisüberblick

- Charakterisierung solcher Zahlen.
- Struktur solcher Zahlen?
- Definition einer bestimmten Untergruppe von $(\mathbb{Z}_n[x]/\langle h(x)\rangle)^*$.
- Abschätzung der Gruppenordnung nach unten.
- Abschätzung der Gruppenordnung nach oben, falls n keine p-Potenz ist.
- Widerspruch!

Überblick

Introspektive Zahlen

Definition

Gilt $f(x)^m \equiv f(x^m) \pmod{x^r - 1, p}$, so heißt m *introspektiv* für $f(x)$.

Introspektive Zahlen

Definition

Gilt $f(x)^m \equiv f(x^m) \pmod{x^r - 1, p}$, so heißt m *introspektiv* für $f(x)$.

Satz (ohne Beweis)

1. *m introspektiv für $f(x)$ und $g(x)$, dann auch für $f(x) \cdot g(x)$.*
2. *m, m' introspektiv für $f(x)$, dann $m \cdot m'$ auch*

Überblick

Notation: $\ell := \lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$.

Notation: $\ell := \lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$.

Definition

- $I := \{(\frac{n}{p})^i p^j \mid i, j \geq 0\}$.
- $P := \{\prod_{a=0}^{\ell} (x+a)^{e_a} \mid e_a \geq 0\}$.

Notation: $\ell := \lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$.

Definition

- $I := \{ (\frac{n}{p})^i p^j \mid i, j \geq 0 \}$.
- $P := \{ \prod_{a=0}^{\ell} (x+a)^{e_a} \mid e_a \geq 0 \}$.

Beobachtung

Jedes $m \in I$ ist introspektiv für **jedes** $f(x) \in P$.

Notation: $\ell := \lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$.

Definition

- $I := \{(\frac{n}{p})^i p^j \mid i, j \geq 0\}$.
- $P := \{\prod_{a=0}^{\ell} (x+a)^{e_a} \mid e_a \geq 0\}$.

Beobachtung

Jedes $m \in I$ ist introspektiv für **jedes** $f(x) \in P$.

Wiederholung Algebra

- $\mathbb{Z}_p$ ist ein Körper.
- $\mathbb{Z}_p[x]/\langle h(x) \rangle$ ist genau dann ein Körper, wenn $h(x)$ irreduzibel ist.
- Polynome über Körpern haben maximal soviele Nullstellen wie ihr Grad angibt (i.A. falsch! $x^2 - 1 \in \mathbb{Z}_8[x]$ hat 4 Nullstellen)
- $x^r - 1$ ist nicht irreduzibel (Nullstellen sind genau die r-ten Einheitswurzeln).
- $x^r - 1$ hat über $\mathbb{Z}_p$ aber irreduzible Faktoren von Grad > 1.

Notation: $\ell := \lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$.

Definition

- $I := \{(\frac{n}{p})^i p^j \mid i, j \geq 0\}$.
- $P := \{\prod_{a=0}^{\ell} (x+a)^{e_a} \mid e_a \geq 0\}$.

Sei $h(x)$ ein irreduzibler Faktor von Grad > 1 von $x^r - 1$ und $\mathbb{F} := \mathbb{Z}_p[x]/\langle h(x) \rangle$.

Notation: $\ell := \lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$.

Definition

- $I := \{(\frac{n}{p})^i p^j \mid i, j \geq 0\}$.
- $P := \{\prod_{a=0}^{\ell} (x+a)^{e_a} \mid e_a \geq 0\}$.

Sei $h(x)$ ein irreduzibler Faktor von Grad > 1 von $x^r - 1$ und $\mathbb{F} := \mathbb{Z}_p[x]/\langle h(x) \rangle$.

Definition

- $G :=$ „I modulo r“, $t := |G|$
- $\mathcal{G} :=$ „P modulo $h(x)$ und p“.

Notation: $\ell := \lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$.

Definition

- $I := \{(\frac{n}{p})^i p^j \mid i, j \geq 0\}$.
- $P := \{\prod_{a=0}^{\ell} (x+a)^{e_a} \mid e_a \geq 0\}$.

Sei $h(x)$ ein irreduzibler Faktor von Grad > 1 von $x^r - 1$ und $\mathbb{F} := \mathbb{Z}_p[x]/\langle h(x) \rangle$.

Definition

- $G :=$ „I modulo r“, $t := |G|$
- $\mathcal{G} :=$ „P modulo $h(x)$ und p“.

Beobachtung

- G ist Untergruppe von $\mathbb{Z}_r^*$ ($\mathrm{ggT}(m, r) = 1$ für alle $m \in I$ nach Schritt 3).
- $\mathcal{G}$ ist Untergruppe von $\mathbb{F}^*$ und wird von $x, x+1, ..., x+\ell$ erzeugt.

Überblick

Abschätzung nach unten

Satz

$$|\mathcal{G}| \geq \binom{t+\ell}{t-1}$$

Abschätzung nach unten

Satz

$$|\mathcal{G}| \geq \binom{t+\ell}{t-1}$$

Beweisskizze.

- Je zwei Polynome von Grad $< t$ bilden verschiedene Restklassen in $\mathcal{G}$.
- Restklassen von $x, x+1, x+2, ..., x+\ell$ alle verschieden und ungleich 0.
- Anzahl der Polynome von Grad kleiner t ist Anzahl der Möglichkeiten $t-1$ Elemente aus der $\ell+2$ elementigen Menge $\{1, x, x+1, ..., x+\ell\}$ zu wählen (mit Wiederholung, ohne Reihenfolge):

$$\binom{(\ell+2)+(t-1)-1}{t-1} = \binom{\ell+t}{t-1}.$$

□

Abschätzung nach oben

Satz

Ist n keine Potenz von p, so gilt:

$$|\mathcal{G}| \leq n^{\sqrt{t}}.$$

Abschätzung nach oben

Satz

Ist n keine Potenz von p, so gilt:

$$|\mathcal{G}| \leq n^{\sqrt{t}}.$$

Beweisskizze.

- $\hat{I} := \{(\frac{n}{p})^i p^j \mid 0 \leq i, j \leq \lfloor\sqrt{t}\rfloor\}$.
- Ist n keine Potenz von p, so gilt $|\hat{I}| = (\lfloor\sqrt{t}\rfloor + 1)^2 > t$.
- Also muss es $m_1, m_2 \in \hat{I}$ geben mit $m_1 \equiv m_2 \pmod{r}$.
- Dann gilt aber $f(x)^{m_1} = f(x)^{m_2} \pmod{h(x), p}$.
- $f(x) \in \mathbb{F}$ ist Nullstelle von $Q(y) := y^{m_1} - y^{m_2} \in \mathbb{F}[y]$.
- $|\mathcal{G}| \leq$ #Nullstellen $\leq m_1 \leq n^{\sqrt{t}}$.

□

n muss eine p-Potenz sein

Satz (ohne Beweis)

$$|\mathcal{G}| \geq \binom{t+\ell}{t-1} \geq 2^{\lfloor \sqrt{t} \log(n) \rfloor + 1} > n^{\sqrt{t}}$$

Folgerung

- Ist n **keine** p-Potenz, so gilt:

$$n^{\sqrt{t}} < |\mathcal{G}| \leq n^{\sqrt{t}}.$$

- Also $n = p^k$ für ein $k \in \mathbb{N}$.
- Wäre $k \geq 2$, so hätte Schritt 1 des Algorithmus $\textsc{Zusammengesetzt}$ ausgegeben.
- Also $k = 1$.

□

Überblick

Terminiert der Algorithmus?

Zusammenfassung

Wir haben gezeigt:

$$n \text{ prim} \;\Leftrightarrow\; \text{Algorithmus gibt } \textsc{Prim} \text{ aus.}$$

Terminiert der Algorithmus?

Zusammenfassung

Wir haben gezeigt:

$$n \text{ prim} \Leftrightarrow \text{Algorithmus gibt } \textsc{Prim} \text{ aus.}$$

Fragen

- Gibt er immer etwas aus?
- Schritt 2: Finde das kleinste r, so dass $\mathrm{ord}_r(n) > \log^2 n$.
- Gibt es so ein r überhaupt immer?

Terminiert der Algorithmus?

Zusammenfassung

Wir haben gezeigt:

$$n \text{ prim } \Leftrightarrow \text{ Algorithmus gibt } \textsc{Prim} \text{ aus.}$$

Fragen

- Gibt er immer etwas aus?
- Schritt 2: Finde das kleinste r, so dass $\mathrm{ord}_r(n) > \log^2 n$.
- Gibt es so ein r überhaupt immer?

Antwort

Ja! So ein r lässt sich **immer** finden (und es lässt sich „schnell“ finden).

Terminiert der Algorithmus?

Zusammenfassung

Wir haben gezeigt:

$$n \text{ prim} \;\Leftrightarrow\; \text{Algorithmus gibt } \textsc{Prim} \text{ aus.}$$

Fragen

- Gibt er immer etwas aus?
- Schritt 2: Finde das kleinste r, so dass $\text{ord}_r(n) > \log^2 n$.
- Gibt es so ein r überhaupt immer?

Antwort

Ja! So ein r lässt sich **immer** finden (und es lässt sich „schnell“ finden).

Satz (ohne Beweis)

Für jedes $n > 2$ existiert $r \leq \lceil \log^5(n) \rceil$ mit $\text{ord}_r(n) > \log^2(n)$.

Überblick

Laufzeitabschätzung

Satz

Der AKS-Algorithmus kann mit einer Laufzeit von

$$\tilde{O}(\log^{\frac{21}{2}}(n)) = O(\log^{\frac{21}{2}+\epsilon}(n))$$

implementiert werden.

Laufzeitabschätzung des AKS-Algorithmus

Eingabe: $n \in \mathbb{N}$ mit $n > 1$.

1. Falls $n = a^b$ für $a, b \in \mathbb{N}$ und $a, b > 1$, gebe $\textsc{Zusammengesetzt}$ aus.
2. Finde das kleinste r, so dass $\text{ord}_r(n) > \log^2 n$.
3. Falls $1 < \text{ggT}(a, n) < n$ für ein $a \leq r$, gebe $\textsc{Zusammengesetzt}$ aus.
4. Falls $n \leq r$, gebe $\textsc{Prim}$ aus.
5. Für $a = 1$ bis $\lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$
   Falls $(x + a)^n \not\equiv x^n + a \pmod{x^r - 1, n}$, gebe $\textsc{Zusammengesetzt}$ aus.
6. Gebe $\textsc{Prim}$ aus.

Laufzeitabschätzung des AKS-Algorithmus

Eingabe: $n \in \mathbb{N}$ mit $n > 1$.

1. Falls $n = a^b$ für $a, b \in \mathbb{N}$ und $a, b > 1$, gebe ZUSAMMENGESETZT aus.
2. Finde das kleinste r, so dass $\text{ord}_r(n) > \log^2 n$.
3. Falls $1 < \text{ggT}(a, n) < n$ für ein $a \leq r$, gebe ZUSAMMENGESETZT aus.
4. Falls $n \leq r$, gebe PRIM aus.
5. Für $a = 1$ bis $\lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$

   Falls $(x + a)^n \not\equiv x^n + a$ (mod $x^r - 1, n$), gebe ZUSAMMENGESETZT aus.
6. Gebe PRIM aus.

$\tilde{O}(\log^3(n))$

Laufzeitabschätzung des AKS-Algorithmus

Eingabe: $n \in \mathbb{N}$ mit $n > 1$.

1. Falls $n = a^b$ für $a, b \in \mathbb{N}$ und $a, b > 1$, gebe ZUSAMMENGESETZT aus.
2. Finde das kleinste r, so dass $\mathrm{ord}_r(n) > \log^2 n$.
3. Falls $1 < \mathrm{ggT}(a, n) < n$ für ein $a \leq r$, gebe ZUSAMMENGESETZT aus.
4. Falls $n \leq r$, gebe PRIM aus.
5. Für $a = 1$ bis $\lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$
   Falls $(x + a)^n \not\equiv x^n + a \pmod{x^r - 1, n}$, gebe ZUSAMMENGESETZT aus.
6. Gebe PRIM aus.

$$\tilde{O}(\log^3(n)) + \tilde{O}(\log^7(n))$$

- $n^i \stackrel{?}{=} 1$ für $1 \leq i \leq \log^2(n)$ testen.
- Jeweils $O(\log^2(n))$ Multiplikationen modulo r, jede einzelne in $\tilde{O}(\log(r))$.
- Spätestens bei $r = \lceil \log^5(n) \rceil$ klappts, also: $\tilde{O}(\log^2(n) \cdot \log(r)) = \tilde{O}(\log^{2+5}(n))$.

Laufzeitabschätzung des AKS-Algorithmus

Eingabe: $n \in \mathbb{N}$ mit $n > 1$.

1. Falls $n = a^b$ für $a, b \in \mathbb{N}$ und $a, b > 1$, gebe ZUSAMMENGESETZT aus.
2. Finde das kleinste r, so dass $\text{ord}_r(n) > \log^2 n$.
3. Falls $1 < \text{ggT}(a, n) < n$ für ein $a \leq r$, gebe ZUSAMMENGESETZT aus.
4. Falls $n \leq r$, gebe PRIM aus.
5. Für $a = 1$ bis $\lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$

   Falls $(x + a)^n \not\equiv x^n + a \pmod{x^r - 1, n}$, gebe ZUSAMMENGESETZT aus.
6. Gebe PRIM aus.

$$\tilde{O}(\log^3(n)) + \tilde{O}(\log^7(n)) + O(\log^6(n))$$

- r-mal den größten gemeinsamen Teiler finden (jeweils in $O(\log(n))$ mit Euklid).
- $r \leq \lceil \log^5(n) \rceil$ Iterationen.
- Also: $O(r \cdot \log(n)) = O(\log^{5+1}(n))$.

Laufzeitabschätzung des AKS-Algorithmus

Eingabe: $n \in \mathbb{N}$ mit $n > 1$.

1. Falls $n = a^b$ für $a, b \in \mathbb{N}$ und $a, b > 1$, gebe ZUSAMMENGESETZT aus.
2. Finde das kleinste r, so dass $\mathrm{ord}_r(n) > \log^2 n$.
3. Falls $1 < \mathrm{ggT}(a, n) < n$ für ein $a \leq r$, gebe ZUSAMMENGESETZT aus.
4. Falls $n \leq r$, gebe PRIM aus.
5. Für $a = 1$ bis $\lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$

   Falls $(x + a)^n \not\equiv x^n + a \pmod{x^r - 1, n}$, gebe ZUSAMMENGESETZT aus.
6. Gebe PRIM aus.

$$\tilde{O}(\log^3(n)) + \tilde{O}(\log^7(n)) + O(\log^6(n)) + O(\log(n))$$

Laufzeitabschätzung des AKS-Algorithmus

Eingabe: $n \in \mathbb{N}$ mit $n > 1$.

1. Falls $n = a^b$ für $a, b \in \mathbb{N}$ und $a, b > 1$, gebe ZUSAMMENGESETZT aus.
2. Finde das kleinste r, so dass $\mathrm{ord}_r(n) > \log^2 n$.
3. Falls $1 < \mathrm{ggT}(a, n) < n$ für ein $a \leq r$, gebe ZUSAMMENGESETZT aus.
4. Falls $n \leq r$, gebe PRIM aus.
5. Für $a = 1$ bis $\lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$
   Falls $(x + a)^n \not\equiv x^n + a \pmod{x^r - 1, n}$, gebe ZUSAMMENGESETZT aus.
6. Gebe PRIM aus.

$$\tilde{O}(\log^3(n)) + \tilde{O}(\log^7(n)) + O(\log^6(n)) + O(\log(n)) + \tilde{O}(\log^{\frac{21}{2}}(n))$$

- $\tilde{O}(\log(n))$ Polynommultiplikationen pro Iteration.
- $\tilde{O}(\log(n) \cdot r \cdot \log^2(n)) = \tilde{O}(r \log^3(n))$.
- Insgesamt: $\tilde{O}(\sqrt{\varphi(r)} \cdot r \cdot \log^3(n)) = \tilde{O}(\log(n)^{5 \cdot \frac{3}{2} + 3})$.

Laufzeitabschätzung des AKS-Algorithmus

Eingabe: $n \in \mathbb{N}$ mit $n > 1$.

1. Falls $n = a^b$ für $a, b \in \mathbb{N}$ und $a, b > 1$, gebe ZUSAMMENGESETZT aus.
2. Finde das kleinste r, so dass $\text{ord}_r(n) > \log^2 n$.
3. Falls $1 < \text{ggT}(a, n) < n$ für ein $a \leq r$, gebe ZUSAMMENGESETZT aus.
4. Falls $n \leq r$, gebe PRIM aus.
5. Für $a = 1$ bis $\lfloor \sqrt{\varphi(r)} \cdot \log(n) \rfloor$
   Falls $(x + a)^n \not\equiv x^n + a \pmod{x^r - 1, n}$, gebe ZUSAMMENGESETZT aus.
6. Gebe PRIM aus.

Gesamtlaufzeit

$$\tilde{O}(\log^{\frac{21}{2}}(n))$$

□

Überblick

Zusammenfassung

Was man behalten sollte

- Es lässt sich **effizient** testen, ob eine natürliche Zahl prim ist.
- Der Beweis ist „einfach“ und schön und kommt mit einem Semester Algebra aus.
- In der Praxis interessiert das niemanden, denn andere Tests sind (noch?) schneller.

Zusammenfassung

Was man behalten sollte

- Es lässt sich **effizient** testen, ob eine natürliche Zahl prim ist.
- Der Beweis ist „einfach“ und schön und kommt mit einem Semester Algebra aus.
- In der Praxis interessiert das niemanden, denn andere Tests sind (noch?) schneller.

Offene Frage

- Kann man effizient faktorisieren?

Zusammenfassung

Was man behalten sollte

- Es lässt sich **effizient** testen, ob eine natürliche Zahl prim ist.
- Der Beweis ist „einfach“ und schön und kommt mit einem Semester Algebra aus.
- In der Praxis interessiert das niemanden, denn andere Tests sind (noch?) schneller.

Offene Frage

- Kann man effizient faktorisieren?

Vielen Dank für die Aufmerksamkeit!