Mechanisation of the AKS Algorithm

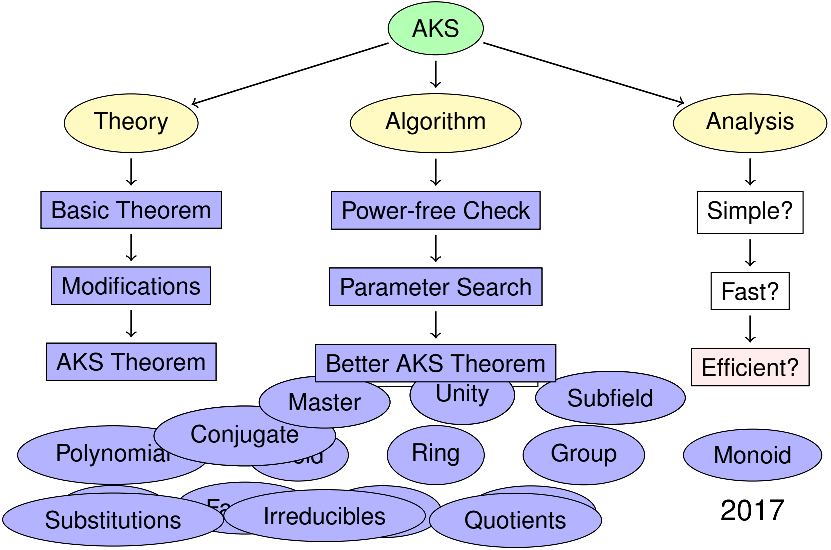
Hing-Lun Chan

College of Engineering and Computer Science
Australian National University

PhD Monitoring 2017

Outline

Mechanisation of AKS Algorithm – Road Map



Back in 2016

Revised:

- Need: to gather information about Cyclotomic factors.
- Need: to establish the existence of Finite Fields.
- Key: obtain a count of monic irreducibles for a given degree.
- Todo: Reformalute AKS proof with a simple parameter k.

Achieved:

- Search for simple k is $O(log\ n)$ if an LCM bound is true.
- A short joint paper to ITP2016 for a cute proof of this result: $2^n \leq LCM\ \{1; 2; 3; \ldots; (n+1)\}$.
- An implicit formula for the monic irreducibles count.
- A finite field exists with cardinality p^n, for prime p and $0 < n$.

Now in 2017

Achieved:

- Worked out a theory of Cyclotomic factors.
- Removed the prime requirement on AKS parameter.
- Reformulated the AKS proof with a simple parameter k.
- Submitted a paper on AKS mechanisation work to JAR.
- Submitted a paper on Finite Field classification to JAR.
- Submitted an extended version of the consecutive LCM bound to JAR.

To Do:

- Investigate a computational model to analyse algorithm.
- Working on: the use of separation logic in computational analysis.
- Working on: include a clock to count the number of steps in code execution.

The Theorem

Theorem

The AKS Primality Test.

$\vdash$ prime n $\iff$ AKS n

The Algorithm

The algorithm starts with a power free test,
then performs a parameter search (AKS_param):

$$
\begin{array}{l}
\text{AKS } n \iff \\
\quad 1 < n \wedge \text{power\_free } n \wedge \\
\quad \textbf{case } \text{AKS\_param } n \textbf{ of} \\
\quad\quad \text{nice } j \Rightarrow j = n \\
\quad \mid \ \text{good } k \Rightarrow \text{poly\_checks } n\ k\ (\sqrt{\varphi(k)} \times \lceil \log n \rceil) \\
\quad \mid \ \text{bad} \Rightarrow \text{F}
\end{array}
$$

The Algorithm

The algorithm starts with a power free test,
then performs a parameter search (AKS_param):

$$
\begin{array}{l}
\text{AKS}\ \ n \iff \\
\quad 1 < n \land \text{power\_free}\ \ n \land \\
\quad \textbf{case}\ \ \text{AKS\_param}\ \ n\ \ \textbf{of} \\
\qquad \ \ \text{nice}\ \ j \Rightarrow j = n \\
\quad \mid \ \text{good}\ \ k \Rightarrow \text{poly\_checks}\ \ n\ \ k\ \ (\sqrt{\varphi(k)} \times \lceil \log n \rceil) \\
\quad \mid \ \text{bad} \Rightarrow \text{F}
\end{array}
$$

Of the 3 result from the search:

- a nice j takes a single check for TRUE or FALSE,
- a good k needs further polynomial checks, and
- bad never happens.

The Pseudo Code

Input: integer $n > 1$.

1. **Power Free Test**
   For each $j = 2$ to $\lceil \log n \rceil$:
   - If (integer j-th root of n)$^j = n$, $\texttt{COMPOSITE}$.
2. **Parameter Search**
   For each $k = 2$ to $2 + \frac{\lceil \log n \rceil^5}{2}$:
   - If $k \mid n$, then if $k = n$ $\texttt{PRIME}$ else $\texttt{COMPOSITE}$.
   - If $k \geq \lceil \log n \rceil^2 \wedge \text{order}_k(n) \geq \lceil \log n \rceil^2$, go to Step 3.
3. **Identity Checks**
   For each $c = 1$ to $\sqrt{\varphi(k)} \times \lceil \log n \rceil$:
   - if $(\boldsymbol{X} + \boldsymbol{c})^n \not\equiv (\boldsymbol{X}^n + \boldsymbol{c}) \pmod{n,\ \boldsymbol{X}^k - 1}$, $\texttt{COMPOSITE}$.
4. return $\texttt{PRIME}$.

Is 91 a prime?

Is 91 a prime?

Trial division (known since antiquity)

- not divisible by $2, 3, 5$.
- but divisible by 7, so `COMPOSITE`.

Is 91 a prime?

Trial division (known since antiquity)

- not divisible by $2, 3, 5$.
- but divisible by 7, so `COMPOSITE`.

Fermat's method (around 1640)

- $91 = 100 - 9 = 10^2 - 3^2$, must be `COMPOSITE`.

Is 91 a prime?

Trial division (known since antiquity)

- not divisible by $2, 3, 5$.
- but divisible by 7, so COMPOSITE.

Fermat's method (around 1640)

- $91 = 100 - 9 = 10^2 - 3^2$, must be COMPOSITE.
- by $x^2 - y^2 = (x - y)(x + y)$, $91 = (10 - 3)(10 + 3) = 7 \times 13$.

Is 91 a prime?

Trial division (known since antiquity)

- not divisible by $2, 3, 5$.
- but divisible by 7, so COMPOSITE.

Fermat's method (around 1640)

- $91 = 100 - 9 = 10^2 - 3^2$, must be COMPOSITE.
- by $x^2 - y^2 = (x - y)(x + y)$, $91 = (10 - 3)(10 + 3) = 7 \times 13$.

AKS method (August 2002)

- Search: found `nice` 7 that divides 91, so COMPOSITE.
- Even if this is missed, polynomial check gives:
  - $(x + 1)^{91} \not\equiv x^{91} + 1 \pmod{91, x^{37} - 1}$.
  - LHS: $(x + 1)^{91} \equiv 13x^{35} + \cdots + x^{17} + \cdots + 1 \pmod{91, x^{37} - 1}$
  - RHS: $x^{91} + 1 \equiv x^{17} + 1 \pmod{91, x^{37} - 1}$
- so COMPOSITE.

Is 97 a prime?

Is 97 a prime?

Trial division (known since antiquity)

- not divisible by $2, 3, 5, 7$.
- since $\sqrt{97} \approx 9.85$, so $\texttt{PRIME}$.

Is 97 a prime?

Trial division (known since antiquity)

- not divisible by $2, 3, 5, 7$.
- since $\sqrt{97} \approx 9.85$, so $\texttt{PRIME}$.

Fermat's method (around 1640)

- note $10^2 = 100$ is nearest to 97, try $97 = 10^2 - y^2$, fail.
- fail $97 = 11^2 - y^2 = \cdots = 48^2 - y^2$ where $48 \approx \frac{97}{2}$, so $\texttt{PRIME}$.

Is 97 a prime?

Trial division (known since antiquity)

- not divisible by $2, 3, 5, 7$.
- since $\sqrt{97} \approx 9.85$, so $\texttt{PRIME}$.

Fermat's method (around 1640)

- note $10^2 = 100$ is nearest to 97, try $97 = 10^2 - y^2$, fail.
- fail $97 = 11^2 - y^2 = \cdots = 48^2 - y^2$ where $48 \approx \frac{97}{2}$, so $\texttt{PRIME}$.

AKS method (August 2002)

- Search: found $\texttt{good}$ 59.
- Polynomial checks:
  - $(x + 1)^{97} \equiv x^{97} + 1 \pmod{97, x^{59} - 1}$ $\texttt{ok}$,
  - $(x + 2)^{97} \equiv x^{97} + 2 \pmod{97, x^{59} - 1}$ $\texttt{ok}$, $\cdots$, up to
  - $(x + 48)^{97} \equiv x^{97} + 48 \pmod{97, x^{59} - 1}$, all $\texttt{ok}$.
- so $\texttt{PRIME}$.

Possible Timeline

Thesis plan:

April, 2015:	AKS Main Theorem (with suitable prime k)
June, 2016:	AKS Main Theorem (with suitable k)
October, 2016:	AKS Computational Steps Identification
December, 2017:	Computational Model for AKS algorithm
September, 2018:	Thesis written (hopefully!)

Publications

Publications:

- CPP2012: A String of Pearls: Proofs of Fermat's Little Theorem
- JFR2013: Extended version for Journal of Formalized Reasoning
- ITP2015: Mechanisation of AKS Algorithm Part 1: Main Theorem
- ITP2016: Bounding LCMs with Triangles (a simple lower bound)
- JAR2017: (accepted) Bounding LCMs with Triangles (both lower and upper bounds)
- JAR2017? (rejected) Mechanisation of AKS Algorithm (revised and improved)
- JAR2017: (subject to revision) Classification of Finite Fields with Applications