Mechanisation of the AKS Algorithm

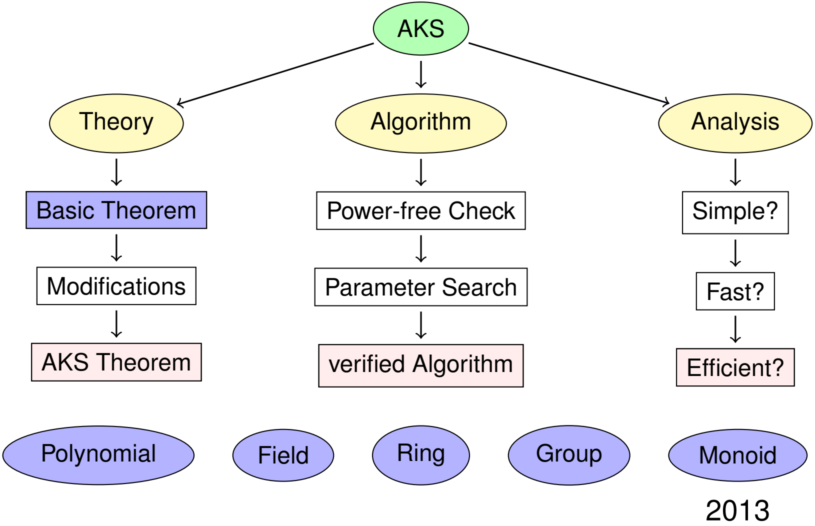
Hing-Lun Chan

College of Engineering and Computer Science
Australian National University

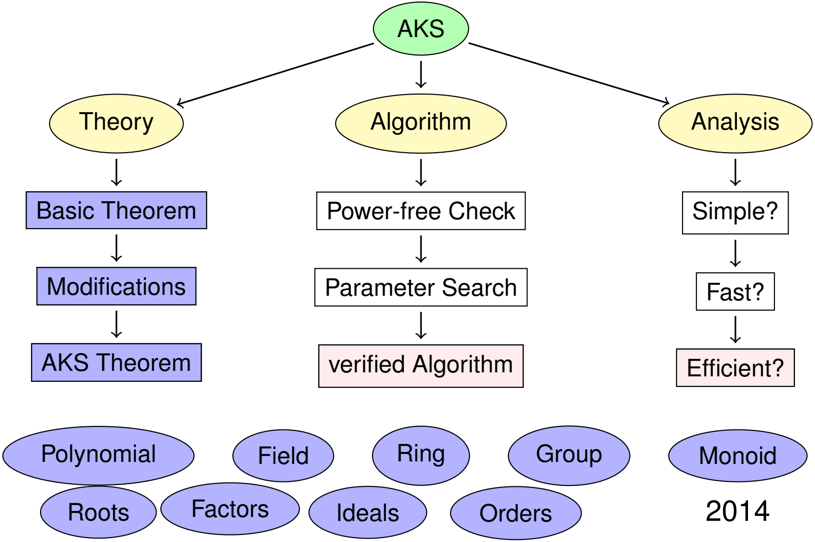
PhD Review 2016

Outline

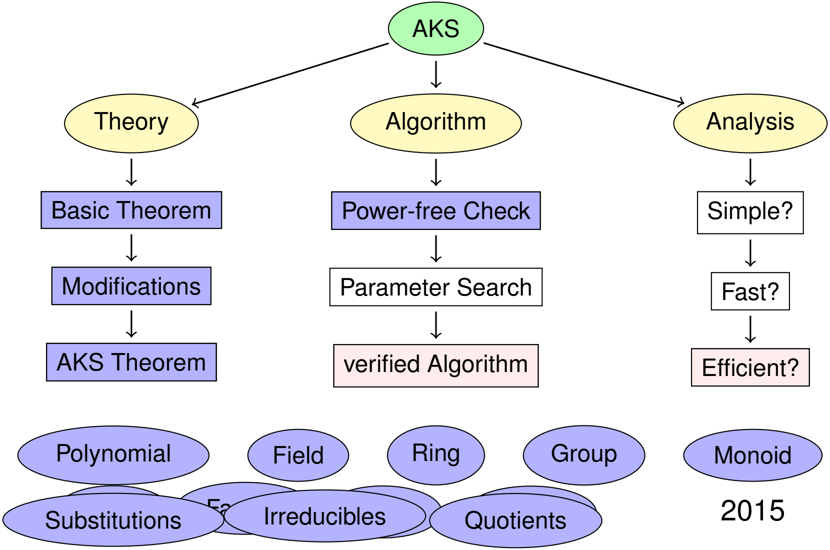
Mechanisation of AKS Algorithm – Road Map



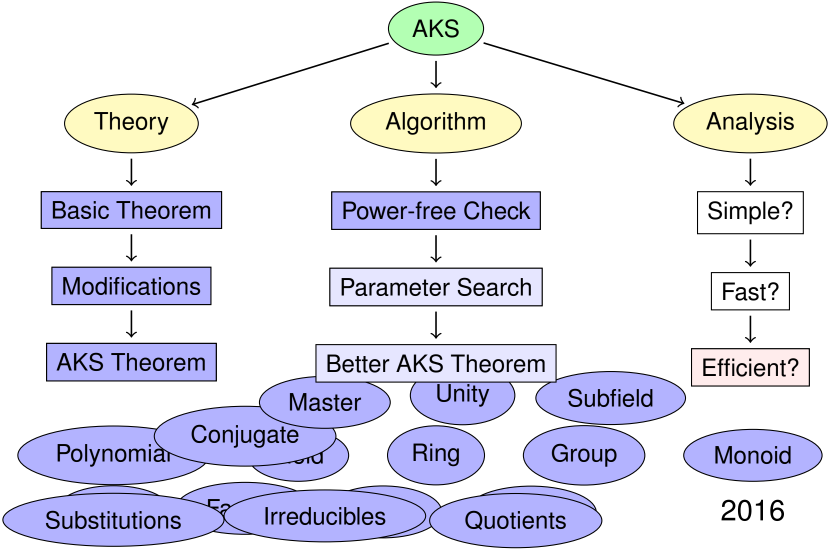
Mechanisation of AKS Algorithm – Road Map



Mechanisation of AKS Algorithm – Road Map



Mechanisation of AKS Algorithm – Road Map



Back in 2014

Completed:

- Basic libraries for Group, Ring, Field.
- Basic libraries for Polynomials.
- Show that Polynomials modulo an irreducible form a finite field.

To Do:

- Understand the role of parameters for the AKS algorithm.
- Formulate the notion of introspective relation in AKS proof.
- Formulate the sets and maps critical for the AKS proof.
- How the Pigeonhole Principle will play in AKS proof?

Back in 2015

Completed:

- Put AKS in script, with one parameter k, derived from input n.
- Correct definition of introspective relation after some false starts.
- AKS parameter k imposes injective maps between sets.
- Key: if AKS is false, an injective map between finite sets would violate the Pigeonhole Principle.
- Part 1: a prime $k \Rightarrow$ AKS Main Theorem, and such k exists.

To Do:

- Why 2nd version of AKS proof does not require k to be prime?
- Remove the prime requirement on k for the formalized AKS proof.
- Show that parameter k can be found within steps of $O(\log n)$.

Now in 2016

Revised:

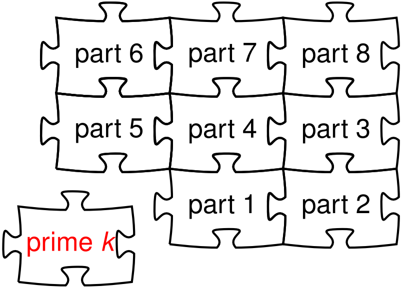
- Need: to gather information about Cyclotomic factors.
- Need: to establish the existence of Finite Fields.
- Key: obtain a count of monic irreducibles for a given degree.
- Todo: Reformalute AKS proof with a simple parameter k.

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 $0 < n$.

AKS Jigsaw Puzzle

Parameter k is derived from input n



Input: a power-free number n

1: Coprime checks for $\{1; \ldots; k\}$

2: From k derive another limit ℓ

3: Polynomial checks for $\{1; \ldots; \ell\}$

in $\pmod{n, X^k - 1}$

4: Choose p, a prime divisor of n

5: Shift to $\pmod{p, X^k - 1}$

6: Construct introspective sets

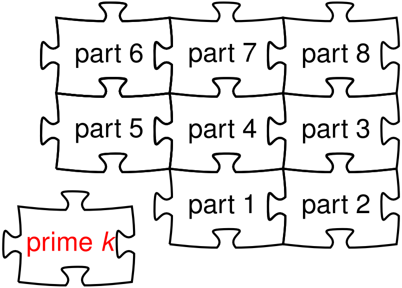
7: Craft finite sets and 1-to-1 maps

8: Conclude: n must be a power of p

or Pigeonhole principle is violated!

AKS Jigsaw Puzzle

Parameter k is derived from input n



Input: a power-free number n

1: Coprime checks for $\{1; \ldots; k\}$

2: From k derive another limit ℓ

3: Polynomial checks for $\{1; \ldots; \ell\}$

in $\pmod{n, X^k - 1}$

4: Choose p, a prime divisor of n

5: Shift to $\pmod{p, X^k - 1}$

6: Construct introspective sets

7: Craft finite sets and 1-to-1 maps

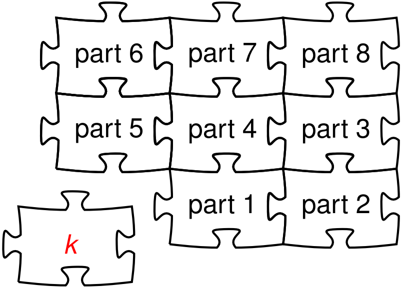
8: Conclude: n must be a power of p

or Pigeonhole principle is violated!

A prime k makes a key proof step easy.

AKS Jigsaw Puzzle – improved

Parameter k is derived from input n



Input: a power-free number n

1: Coprime checks for $\{1; \ldots; k\}$

2: From k derive another limit ℓ

3: Polynomial checks for $\{1; \ldots; \ell\}$

in $\pmod{n, X^k - 1}$

4: Choose p, a prime divisor of n

5: Shift to $\pmod{p, X^k - 1}$

6: Construct introspective sets

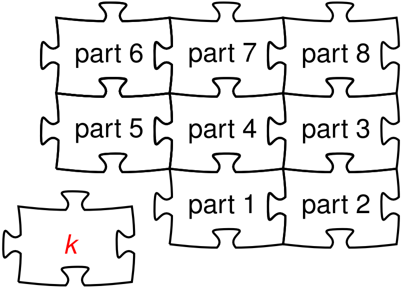
7: Craft finite sets and 1-to-1 maps

8: Conclude: n must be a power of p

or Pigeonhole principle is violated!

AKS Jigsaw Puzzle – improved

Parameter k is derived from input n



Input: a power-free number n

1: Coprime checks for $\{1; \ldots ; k\}$

2: From k derive another limit ℓ

3: Polynomial checks for $\{1; \ldots ; \ell\}$

in $\pmod{n, X^k - 1}$

4: Choose p, a prime divisor of n

5: Shift to $\pmod{p, X^k - 1}$

6: Construct introspective sets

7: Craft finite sets and 1-to-1 maps

8: Conclude: n must be a power of p

or Pigeonhole principle is violated!

Just k will simplify its search bound.

Possible Timeline

Thesis plan:

April, 2015:	AKS Main Theorem (with suitable prime k)
June, 2016:	AKS Main Theorem (with suitable k)
June, 2017:	Complexity/Efficiency of AKS algorithm
December, 2017:	Thesis written (hopefully!)

Possible Timeline

Thesis plan:

April, 2015:	AKS Main Theorem (with suitable prime k)
June, 2016:	AKS Main Theorem (with suitable k)
June, 2017:	Complexity/Efficiency of AKS algorithm
December, 2017:	Thesis written (hopefully!)

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:	(submitted) Bounding LCMs with Triangles
??	(planned) Mechanisation of AKS Algorithm Part 2.