Mechanisation of the AKS Algorithm

Hing-Lun Chan

College of Engineering and Computer Science
Australian National University

PhD Yearly Review 2014

Outline

Mechanisation of AKS Algorithm – Road Map

- Foundation Work:
  - Build Monoid theory in HOL4.($\surd$)
  - Build Group theory from Monoid theory.($\surd$)
  - Build Ring theory using Group and Monoid.($\surd$)
  - Build Field theory using Ring and Group.($\surd$)
  - Build Polynomial theory using Field and Ring.($\surd$)
- Apply to AKS (to be amended):
  - Code in HOL4: *AKS n* that returns true or false upon input n.
  - Prove in HOL4: *AKS n* returns true iff n is prime.
  - Prove in HOL4: number of steps of *AKS n* is bound by $O(\log^k n)$.
  -

Mechanisation of AKS Algorithm – Road Map

- Foundation Work:
  - Build Monoid theory in HOL4.($\surd$)
  - Build Group theory from Monoid theory.($\surd$)
  - Build Ring theory using Group and Monoid.($\surd$)
  - Build Field theory using Ring and Group.($\surd$)
  - Build Polynomial theory using Field and Ring.($\surd$)
- Apply to AKS (now):
  - Define: a number n to be *AKS n* if n satisfies certain properties.
  - Prove (the AKS main theorem): *AKS n* is true iff n is prime.
  - Code: an algorithm to verify that a given number n is *AKS n*.
  - Prove: number of steps of the algorithm is bounded by $O(\log^k n)$.

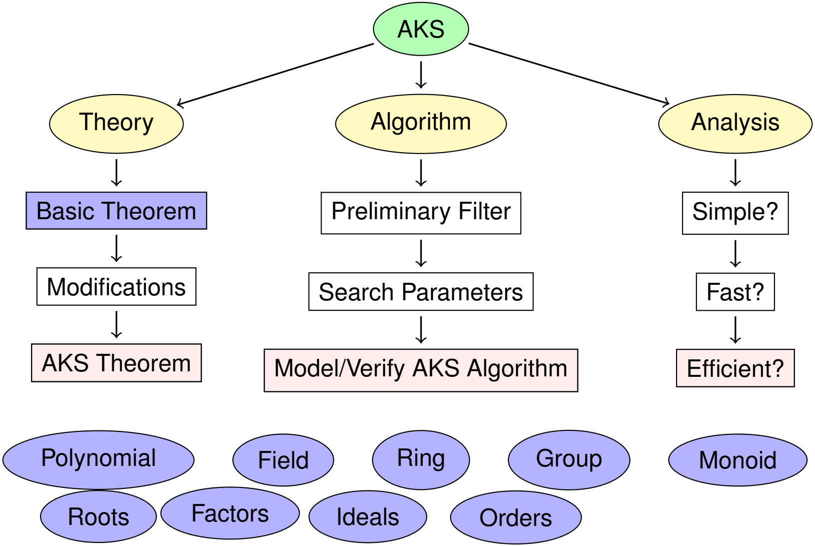
Work Summary — from Elementary

- What Have I Done?
  - Consolidation of basics: Monoid, Group, Ring, Field.
  - Expanded theory of polynomials over Ring and Field.
  - Developed theory of orders for elements in Monoid and Group.
  - Developed polynomial divisions with Ring and Field coefficients.

Work Summary — to Advanced

- What Have I Done?
  - Developed theory of ring ideals and quotient rings.
  - Proved that quotient ring by a maximal ideal is a Field.
  - Developed theory of irreducible elements in a Ring.
  - Developed theory of Principal Ideal Rings (PIR).
  - Proved in PIR, the principal ideal of an irreducible element is maximal.
  - Proved that Euclidean Rings are Principal Ideal Rings.
  - Proved that division property in a polynomial ring makes it Euclidean.

Current Work



Example 1

How to prove: $1 + 1 = 2$

Example 1

How to prove: $1 + 1 = 2$

By Calculator:

- $\boxed{1}$ $\boxed{+}$ $\boxed{1}$ $\boxed{=}$, see display.

Example 1

How to prove: $1 + 1 = 2$

By Calculator:

- $\boxed{1}\ \boxed{+}\ \boxed{1}\ \boxed{=}$, see display.

By Theorem Prover:

- $ONE : |- 1 = SUC\ 0$
- $TWO : |- 2 = SUC\ 1$
- $ADD : |- (\forall n.\ 0 + n = n) \wedge \forall m\, n.\ (SUC\ m) + n = SUC\ (m + n)$
- $1 + 1 = (SUC\ 0) + 1 = SUC\ (0 + 1) = SUC\ 1 = 2$

Example 1

How to prove: $1 + 1 = 2$

By Calculator:

- $\boxed{1}$ $\boxed{+}$ $\boxed{1}$ $\boxed{=}$, see display.

By Theorem Prover:

- $ONE : | - 1 = SUC\ 0$
- $TWO : | - 2 = SUC\ 1$
- $ADD : | - (\forall n.\ 0 + n = n) \wedge \forall m\ n.\ (SUC\ m) + n = SUC\ (m + n)$
- $1 + 1 = (SUC\ 0) + 1 = SUC\ (0 + 1) = SUC\ 1 = 2$

HOL session:

```
- g `1 + 1 = 2`;
- e (decide_tac);
> val it =
    Initial goal proved.
    |- 1 + 1 = 2 :  proof
```

Example 2

Make clear definitions: Just translate the traditional proof?

Example 2

Make clear definitions: Just translate the traditional proof?

Theorem: *There are infinitely many primes*.

Example 2

Make clear definitions: Just translate the traditional proof?

Theorem: *There are infinitely many primes*.

Euclid's proof:

- By contradiction. Let N be the largest prime.
- Consider $(\prod_{\texttt{prime}\ p}^{N} p) + 1$.
- Then conclude in one or two lines.

Example 2

Make clear definitions: Just translate the traditional proof?

Theorem: *There are infinitely many primes*.

Euclid's proof:

- By contradiction. Let N be the largest prime.
- Consider $\left(\prod_{\texttt{prime}\ p}^{N} p\right) + 1$.
- Then conclude in one or two lines.

HOL Tutorial's proof:

- Define: $\texttt{FACT}\ n = \prod_{m=1}^{n} m$.
- Prove properties of $\texttt{FACT}\ n$, *e.g.* $\forall n\, m.\ 1 \leq m \leq n \Rightarrow m \mid \texttt{FACT}\ n$.
- Let N be the largest prime. Consider $(\texttt{FACT}\ N) + 1$.
- Derive a contradiction by properties of factorial and divisibility.

Example 2

Make clear definitions: Just translate the traditional proof?

Theorem: *There are infinitely many primes*.

Euclid's proof:

- By contradiction. Let N be the largest prime.
- Consider $(\prod_{\texttt{prime}\ p}^{N} p) + 1$.
- Then conclude in one or two lines.

HOL Tutorial's proof:

- Define: $\texttt{FACT}\ n = \prod_{m=1}^{n} m$.
- Prove properties of $\texttt{FACT}\ n$, *e.g.* $\forall n\, m.\ 1 \leq m \leq n \Rightarrow m \mid \texttt{FACT}\ n$.
- Let N be the largest prime. Consider $(\texttt{FACT}\ N) + 1$.
- Derive a contradiction by properties of factorial and divisibility.

HOL builds up a tool: the `Factorial` library.

Example 3

Develop useful tools: To what level?

Example 3

Develop useful tools: To what level?

Theorem: *Let H be a subgroup of finite group G. Then* $|H| \mid |G|$.

Example 3

Develop useful tools: To what level?

Theorem: *Let H be a subgroup of finite group G. Then* $|H| \mid |G|$.

Basic:

- Consider the cosets of subgroup H.
- Show that each coset has the same size as H.
- Show that these cosets form a partition of G.
- Conclude by appealing to equal-size partition.

G

H	aH
bH	cH

Example 3

Develop useful tools: To what level?

Theorem: *Let H be a subgroup of finite group G. Then $|H| \mid |G|$.*

Basic:

- Consider the cosets of subgroup H.
- Show that each coset has the same size as H.
- Show that these cosets form a partition of G.
- Conclude by appealing to equal-size partition.

G

H	aH
bH	cH

Advanced:

- Define G/H, the quotient group of G by subgroup H.
- Prove that for finite group G, $|G/H| = \dfrac{|G|}{|H|}$.
- Original theorem is a corollary of this general theorem.

Example 3

Develop useful tools: To what level?

Theorem: *Let H be a subgroup of finite group G. Then $|H| \mid |G|$.*

Basic:

- Consider the cosets of subgroup H.
- Show that each coset has the same size as H.
- Show that these cosets form a partition of G.
- Conclude by appealing to equal-size partition.

G

H	aH
bH	cH

Advanced:

- Define G/H, the quotient group of G by subgroup H.
- Prove that for finite group G, $|G/H| = \dfrac{|G|}{|H|}$.
- Original theorem is a corollary of this general theorem.

High-level libraries have more general applications.

Formalization Lessons

How to formalize:

Formalization Lessons

How to formalize:

- Denote concepts by precise definitions.
- Build up a hierarchy of concepts, low-level to high-level.
- Organise related concepts into libraries.
- Find the best strategy to use built-up libraries.

Formalization Lessons

How to formalize:

- Denote concepts by precise definitions.
- Build up a hierarchy of concepts, low-level to high-level.
- Organise related concepts into libraries.
- Find the best strategy to use built-up libraries.

How to program (OOP style):

- Design the objects.
- Develop a hierarchy of objects.
- Put features into libraries.
- Adopt a modular approach and make use of libraries.

Experience with HOL

Working on HOL proofs is similar to crafting a good program:

Experience with HOL

Working on HOL proofs is similar to crafting a good program:

- *A String of Pearls: Proofs of Fermat's Little Theorem* (CPP 2012).
- One proof uses Orbit-Stabilizer Theorem in Group theory.
- Library of algebraic structures form the basis of my current work.

Experience with HOL

Working on HOL proofs is similar to crafting a good program:

- *A String of Pearls: Proofs of Fermat's Little Theorem* (CPP 2012).
- One proof uses Orbit-Stabilizer Theorem in Group theory.
- Library of algebraic structures form the basis of my current work.

HOL Theorem Prover provides a useful type system:

- For example, we can denote a generic type by α.
- From sets with elements of type α, we can define various algebraic structures: $\alpha\,\texttt{group}$, $\alpha\,\texttt{ring}$, $\alpha\,\texttt{field}$.

Experience with HOL

Working on HOL proofs is similar to crafting a good program:

- *A String of Pearls: Proofs of Fermat's Little Theorem* (CPP 2012).
- One proof uses Orbit-Stabilizer Theorem in Group theory.
- Library of algebraic structures form the basis of my current work.

HOL Theorem Prover provides a useful type system:

- For example, we can denote a generic type by α.
- From sets with elements of type α, we can define various algebraic structures: $\alpha\,\texttt{group}$, $\alpha\,\texttt{ring}$, $\alpha\,\texttt{field}$.
- Polynomials over an $\alpha\,\texttt{field}$ are simply $\alpha\,\texttt{poly} = \alpha\,\texttt{list}$.
- These polynomials themselves form a Ring, an $\alpha\,\texttt{poly}\,\texttt{ring}$.
- Theorems proved for $\alpha\,\texttt{ring}$ can be lifted to $\alpha\,\texttt{poly}\,\texttt{ring}$.

Possible Timeline

Revised thesis plan:

June, 2015:	AKS Theorem
June, 2016:	Model/Verify AKS Algorithm
June, 2017:	Complexity/Efficiency
December, 2017:	Thesis written (hopefully!)

My official start-date was 7 April 2012

Switched to part-time to extend original deadline (7 March 2016).