A Formalisation of Lehmer’s Primality Criterion

By Simon Wimmer and Lars Noschinski

August 16, 2018

Abstract

In 1927, Lehmer presented criterions for primality, based on the converse of Fermat’s little theorem [2]. This work formalizes the second criterion from Lehmer’s paper, a necessary and sufficient condition for primality.

As a side product we formalize some properties of Euler’s \( \varphi \)-function, the notion of the order of an element of a group, and the cyclicity of the multiplicative group of a finite field.

Contents

1 Introduction 1

2 Lehmer’s Theorem 1

1 Introduction

Section ?? provides some technical lemmas about polynomials. Section ?? to ?? formalize some basic number-theoretic and algebraic properties: Euler’s \( \varphi \)-function, the order of an element of a group and an upper bound of the number of roots of a polynomial. Section ?? combines these results to prove that the multiplicative group of a finite field is cyclic. Based on that, Section 2 formalizes an extended version of Lehmer’s Theorem, which gives us necessary and sufficient conditions to decide whether a number is prime.

theory Lehmer

imports

Main

HOL-Number-Theory.Residues

begin

2 Lehmer’s Theorem

In this section we prove Lehmer’s Theorem [2] and its converse. These two theorems characterize a necessary and complete criterion for primality. This
criterion is the basis of the Lucas-Lehmer primality test and the primality certificates of Pratt [3].

**Lemma** \(\text{mod-1-coprime-nat}:\)

\[\text{coprime } a \text{ b if } 0 < n \Rightarrow [a \not\sim n = 1] \mod b \text{ for } a \text{ b :: nat}\]

**Proof**
- From that \(\text{coprime-1-left}\) have \(\text{coprime } (a \not\sim n) \text{ b}\)
- Using \(\text{cong-imp-coprime-nat}\) \(\text{cong-sym}\) by blast
- With \(0 < n\) show \(\varnothing\)
- By simp

**QED**

This is a weak variant of Lehmer’s theorem: All numbers less then \(p - 1\) must be considered.

**Lemma** \(\text{lehmers-weak-theorem}:\)

- Assumes \(2 \leq p\)
- Assumes \(\min-cong1: \forall x. 0 < x \Rightarrow x < p - 1 \Rightarrow [a \not\sim x \neq 1] \mod p\)
- Assumes \(\text{cong1}: [a \not\sim (p - 1) = 1] \mod p\)
- Shows prime \(p\)

**Proof** (rule totient-imp-prime)
- From \(\langle 2 \leq p \rangle\) have \(\text{coprime } a \text{ p}\)
- By \(\langle \text{intro mod-1-coprime-nat[of } p - 1\rangle \text{ auto}\)
- Then have \(\text{totient } p \geq p - 1 \lor \text{totient } p = 0\)
- Using \(\min-cong1\) \(\langle \text{of } \text{totient } p\rangle\) by fastforce
- Moreover have \(\text{totient } p > 0\)
- Using \(\langle 2 \leq p \rangle\) by simp
- Moreover from \(\langle p \geq 2\rangle\) have \(\text{totient } p < p\) by \(\langle \text{intro totient-less} \rangle\) auto
- Ultimately show \(\text{totient } p = p - 1\) by presburger

**QED** (insert \(\langle p \geq 2\rangle, \text{auto}\))

**Lemma** \(\text{prime-factors-elem}:\)

- Fixes \(n :: \text{nat}\)
- Assumes \(1 < n\)
- Shows \(\exists p. p \in \text{prime-factors } n\)
- Using \(\text{assms}\) by \(\langle \text{cases prime } n\rangle\) (auto simp: prime-factors-dvd prime-factor-nat)

**Lemma** \(\text{cong-pow-1-nat}:\)

\[\text{[a \not\sim x = 1] \mod b}\] if \(\text{[a = 1] \mod b}\) for \(a \text{ b :: nat}\)

Using cong-pow \(\langle \text{of } a \text{ b } x\rangle\) that by simp

**Lemma** \(\text{cong-gcd-eq-1-nat}:\)

- Fixes \(a \text{ b :: nat}\)
- Assumes \(0 < m\) and \(\text{cong-props}: [a \not\sim m = 1] \mod b\)
- Shows \(\text{[a \not\sim gcd m n = 1] \mod b}\)

**Proof**
- Obtain \(c\ d\) where \(\text{gcd: } m \ast c = n \ast d + \text{gcd } m\ n\) using bezout-nat \(\langle \text{of } m\ n\rangle\) \(\langle 0 < m\rangle\)
- By auto
- Have cong-m: \(\text{[a \not\sim (m \ast c) = 1] \mod b}\) and cong-n: \(\text{[a \not\sim (n \ast d) = 1] \mod b}\)
- Using cong-props by \(\langle \text{simp-all only: cong-pow-1-nat power-mult}\rangle\)
have \([I \ast a \cdot \text{gcd} \; m \; n = a \cdot (n \ast d) \ast a \cdot \text{gcd} \; m \; n]\) (mod b) 
  by (rule cong-scalar-right, rule cong-sym) (fact cong-n)

also have \([a \cdot (n \ast d) \ast a \cdot \text{gcd} \; m \; n = a \cdot (m \ast c)]\) (mod b) 
  using gcd by (simp add: power-add)

also have \([a \cdot (m \ast c) = 1]\) (mod b) using cong-m by simp

finally show \([a \cdot \text{gcd} \; m \; n = 1]\) (mod b) by simp

qed

lemma One-leq-div:
1 < b div a if a dvd b a < b for a b :: nat
  using that by (metis dvd-div-mult-self mult_leftNeutral mult_less_cancel2)

theorem lehmers-theorem:
assumes 2 ≤ p
assumes pf-notcong1: \(\forall x.\; x \in \text{prime-factors} (p - 1) \implies [a \cdot ((p - 1) \div x) \neq 1]\) (mod p)
assumes cong1: \([a \cdot (p - 1) = 1]\) (mod p)
shows prime p
proof cases
  assume [a = 1] (mod p) with ⟨2 ≤ p⟩ pf-notcong1 show ?thesis 
    by (metis cong-pow-1-nat less_diff_conv linorder_neqE_nat linorder_not_less prime-factors-elem two_is_prime_nat)

next
  assume A-notcong-1: [a ≠ 1] (mod p)
  { fix x assume 0 < x x < p - 1
    have [a ∗ x ≠ 1] (mod p)
      proof
        assume [a ∗ x = 1] (mod p)
        then have gcd-cong-1: \([a \cdot \text{gcd} \; x \; (p - 1) = 1]\) (mod p) 
          by (rule cong-gcd-eq-1-nat[OF ⟨0 < x⟩ cong1])

        have \(p - 1 = \text{gcd} \; x \; (p - 1)\)
          proof (rule ccontr)
            assume ¬?thesis
            then have gcd-p1: \(\text{gcd} \; x \; (p - 1) \div (p - 1) \cdot \text{gcd} \; x \; (p - 1) < p - 1\)
              using gcd-le2_nat[of p - 1 x] ⟨2 ≤ p⟩ by (simp, linarith)

            define c where \(c = (p - 1) \div (\text{gcd} \; x \; (p - 1))\)
            then have p-1-eq: \(p - 1 = \text{gcd} \; x \; (p - 1) \ast c\) unfolding c_def using gcd-p1 
              by (metis dvd-mult_div-cancel)

            from gcd-p1 have \(1 < c\) unfolding c_def by (rule One-leq-div)
            then obtain q where q-pf: \(q \in \text{prime-factors} \; c\) 
              using prime-factors-elem by auto
            then have q dvd c by auto

            have \(q \in \text{prime-factors} \; (p - 1)\) using q-pf ⟨1 < c⟩ ⟨2 ≤ p⟩ 
              by (subst p-1-eq) (simp add: prime-factors-product)
moreover
have \([a \cdot ((p - 1) \div q) = 1] \pmod{p}\)
  by (subst p-1-eq, subst ded-div-mult-self[OF \(q\ \text{ded}\ c,\ symmetric\)]
  (simp del: One-nat-def add: power-mult gcd-cong-1 cong-pow-1-nat)
ultimately
show False using pf-notcong1 by metis
qed
then show False using \(\langle x < p - 1\rangle\)
  by (metis \(0 < x\) gcd-le1-nat gr-implies-not0 linorder-not-less)
qed

The converse of Lehmer’s theorem is also true.

lemma converse-lehmer-weak:
  assumes prime-p: prime p
  shows \(\exists a. (a^{(p - 1)} = 1] \pmod{p} \land (\forall x . 0 < x \rightarrow x \leq p - 2 \rightarrow [a^x \neq 1] \pmod{p})\)
    \& a > 0 \& a < p
proof
  have p \geq 2 by (rule prime-ge-2-nat[OF prime-p])
  obtain a where a: a \in \(1 .. p - 1\) \& \(1 .. p - 1\) = \(\{a^i \mod{p} | i . i \in UNIV\}\)
    using residue-prime-mult-group-has-gen[OF prime-p] by blast
  { fix x::nat assume x:0 < x \& x \leq p - 2 \& [a^x = 1] \pmod{p}
    have \(a^i \mod{p} | i . i \in UNIV\} = \(\{a^i \mod{p} | i . i \in UNIV\}\) by blast
    { fix y assume y:y \in \(a^i \mod{p} | i . i \in UNIV\}\)
      then obtain i where i:y = a^i mod p by auto
      define q r where q = i \& div x and r = i \& mod x
      have i = q*x + r by (simp add: r-def q-def)
      hence y-q-r:y = (((a \cdot (q*x)) \mod{p}) \& ((a ^ r) \mod{p})) \mod{p}
        by (simp add: i power-add mod-mult-eq)
      have a ^ (q*x) \mod{p} = (a ^ x \mod{p}) \& q \mod{p}
        by (simp add: power-mod mult.commute power-mult[ symmetric])
      then have y-r:y = a ^ r \mod{p} using \(\langle p \geq 2\rangle\) x
        by (simp add: cong-def y-q-r)
      have y \in \(\{a ^ i \mod{p} | i . 0 < i \& i \leq x\}\) by blast
      proof (cases)
        assume r = 0
        then have y = a ^ x \mod{p} using \(\langle p \geq 2\rangle\) x
          by (simp add: cong-def y-r)
        thus ?thesis using x by blast
      next
        assume r \neq 0
      qed
  }
with lehmers-weak-theorem[OF \(2 \leq p\) - cong1] show \(?thesis\) by metis
qed
thus \( \text{thesis using } x \text{ by (auto simp add: y-r r-def)} \)

\[ \text{qed} \]

\[ \text{thus } \{ a \mod p \mid \exists i. i \in \text{UNIV} \} \subseteq \{ a \mod p \mid 0 < i < x \} \text{ by auto} \]

\[ \text{qed} \]

\[ \text{note } X = \text{this} \]

have \( p - 1 = \text{card } \{1 \ldots p - 1\} \) by auto

also have \( \{1 \ldots p - 1\} = \{ a \mod p \mid i. 1 \leq i \land i \leq x \} \) using \( X \) a by auto

also have \( \ldots = (\lambda \ i. \ a \mod p) \cdot \{1..x\} \) by auto

also have \( \text{card } \ldots \leq p - 2 \)

using \( \text{Finite-Set.card-image-le[of } \{1..x\} \lambda \ i. \ a \mod p \text{]} x \) by auto

finally have \( \text{False} \) using \( \langle 2 \leq p \rangle \) by arith

hence \( \forall x . \ 0 < x \rightarrow x \leq p - 2 \rightarrow |a^x \neq 1| \) (mod p) by auto

\[ \text{note a-is-gen = this} \]

\{ assume \( a > 1 \)

have \( \neg p \mid a \)

proof (rule ccontr)

assume \( \neg p \mid a \)

hence \( p \mid a \) by auto

have \( p \leq a \) using dvd-nat-bounds[OF \( \langle p \mid a \rangle \)] a by simp

thus \( \text{False} \) using \( \langle a > 1 \rangle \) a by force

\[ \text{qed} \]

hence \( \text{coPrime a p} \)

using \( \text{prime-imp-coPrime-nat [OF prime-p]} \) by (simp add: ac-simps)

then have \( [a^{\text{totient } p} = 1] \) (mod p)

by (rule euler-theorem)

also from \( \text{prime-p} \) have \( \text{totient } p = p - 1 \)

by (rule totient-prime)

finally have \( [a^{\text{(p - 1)}} = 1] \) (mod p).

\[ \text{hence } [a^{\text{(p - 1)}} = 1] \) (mod p) using a by fastforce

thus \( \text{thesis using a-is-gen a by auto} \)

\[ \text{qed} \]

\[ \text{theorem converse-lehmer:} \]

\[ \text{assumes prime-p:prime(p)} \]

\[ \text{shows } \exists a . [a^{\text{(p - 1)}} = 1] \) (mod p) \land

\( (\forall q. q \in \text{prime-factors } (p - 1) \rightarrow [a^{\text{(p - 1)} \mod q} = 1] \) (mod p)) \land a > 0 \land a < p \]

\[ \text{proof} - \]

have \( p \geq 2 \) by (rule prime-ge-2-nat[OF prime-p])

obtain \( a \) where \( a^{\text{(p - 1)}} = 1 \) (mod p) \land (\forall x . 0 < x \rightarrow x \leq p - 2 \rightarrow [a^x \neq 1] \) (mod p)

\( \land a > 0 \land a < p \)

using converse-lehmer-weak[OF prime-p] by blast

\{ fix q assume q:q \in \text{prime-factors } (p - 1) \]

\[ \text{5} \]
hence \( 0 < q \land q \leq p - 1 \) using \((p \geq 2)\) by \((auto \ simp \ add: \ dvd-nat-bounds \ prime-factors-gt-0-nat)\)

hence \((p - 1) \div q \geq 1\) using \(\text{div-le-mono}[of \ q \ p - 1 \ q]\ \text{div-self}[of \ q]\) by simp

have \(q \geq 2\) using \(q\) by \((auto \ intro: \ prime-ge-2-nat)\)

hence \((p - 1) \div q < p - 1\) using \((p \geq 2)\) by simp

hence \([a^((p - 1) \div q) \neq 1] \ (mod \ p)\) using \(a\) \((p - 1) \div q \geq 1\) by \((auto \ simp \ add: \ Suc-diff-Suc \ less-eq-Suc-le)\)

} thus \(?thesis\) using \(a\) by \(auto\)
qed

end

References

