

# Probabilistic Primality Testing

Daniel Stüwe and Manuel Eberl

September 13, 2023

## Abstract

The most efficient known primality tests are *probabilistic* in the sense that they use randomness and may, with some probability, mistakenly classify a composite number as prime – but never a prime number as composite. Examples of this are the Miller–Rabin test, the Solovay–Strassen test, and (in most cases) Fermat’s test.

This entry defines these three tests and proves their correctness. It also develops some of the number-theoretic foundations, such as Carmichael numbers and the Jacobi symbol with an efficient executable algorithm to compute it.

## Contents

<b>1</b>	<b>Additional Facts about the Legendre Symbol</b>	<b>3</b>
<b>2</b>	<b>Auxiliary Material</b>	<b>5</b>
<b>3</b>	<b>The Jacobi Symbol</b>	<b>11</b>
<b>4</b>	<b>Residue Rings of Natural Numbers</b>	<b>15</b>
4.1	The multiplicative group of residues modulo $n$ . . . . .	15
4.2	The ring of residues modulo $n$ . . . . .	16
4.3	The ring of residues modulo a prime . . . . .	17
4.4	$-1$ in residue rings . . . . .	18
<b>5</b>	<b>Additional Material on Quadratic Residues</b>	<b>18</b>
<b>6</b>	<b>Euler Witnesses</b>	<b>19</b>
<b>7</b>	<b>Carmichael Numbers</b>	<b>22</b>
<b>8</b>	<b>Fermat Witnesses</b>	<b>24</b>
<b>9</b>	<b>A Generic View on Probabilistic Prime Tests</b>	<b>28</b>
<b>10</b>	<b>Fermat's Test</b>	<b>29</b>
<b>11</b>	<b>The Miller–Rabin Test</b>	<b>30</b>
<b>12</b>	<b>The Solovay–Strassen Test</b>	<b>31</b>

# 1 Additional Facts about the Legendre Symbol

**theory** *Legendre-Symbol*

**imports**

*HOL-Number-Theory.Number-Theory*

**begin**

**lemma** *basic-cong[simp]*:

**fixes**  $p :: int$

**assumes**  $2 < p$

**shows**  $[-1 \neq 1] (mod\ p)$

$[1 \neq -1] (mod\ p)$

$[0 \neq 1] (mod\ p)$

$[1 \neq 0] (mod\ p)$

$[0 \neq -1] (mod\ p)$

$[-1 \neq 0] (mod\ p)$

*<proof>*

**lemma** *[simp]*:  $0 < n \implies (a\ mod\ 2) ^ n = a\ mod\ 2$  **for**  $n :: nat$  **and**  $a :: int$

*<proof>*

**lemma** *Legendre-in-cong-eq*:

**fixes**  $p :: int$

**assumes**  $p > 2$  **and**  $b \in \{-1, 0, 1\}$

**shows**  $[Legendre\ a\ m = b] (mod\ p) \longleftrightarrow Legendre\ a\ m = b$

*<proof>*

**lemma** *Legendre-p-eq-2[simp]*:  $Legendre\ a\ 2 = a\ mod\ 2$

*<proof>*

**lemma** *Legendre-p-eq-1[simp]*:  $Legendre\ a\ 1 = 0$  *<proof>*

**lemma** *euler-criterion-int*:

**assumes** *prime*  $p$  **and**  $2 < p$

**shows**  $[Legendre\ a\ p = a^{((nat\ p-1)\ div\ 2)}] (mod\ p)$

*<proof>*

**lemma** *QuadRes-neg[simp]*:  $QuadRes\ (-p)\ a = QuadRes\ p\ a$  *<proof>*

**lemma** *Legendre-neg[simp]*:  $Legendre\ a\ (-p) = Legendre\ a\ p$  *<proof>*

**lemma** *Legendre-mult[simp]*:

**assumes** *prime*  $p$

**shows**  $Legendre\ (a*b)\ p = Legendre\ a\ p * Legendre\ b\ p$

*<proof>*

**lemma** *QuadRes-mod[simp]*:  $p\ dvd\ n \implies QuadRes\ p\ (a\ mod\ n) = QuadRes\ p\ a$

*<proof>*

**lemma** *Legendre-mod*[simp]:  $p \text{ dvd } n \implies \text{Legendre } (a \text{ mod } n) p = \text{Legendre } a p$   
(proof)

**lemma** *two-cong-0-iff*:  $[2 = 0] \text{ (mod } p) \longleftrightarrow p = 1 \vee p = 2$  **for**  $p :: \text{nat}$   
(proof)

**lemma** *two-cong-0-iff-nat*:  $[2 = 0] \text{ (mod int } p) \longleftrightarrow p = 1 \vee p = 2$   
(proof)

**lemma** *two-cong-0-iff-int*:  $p > 0 \implies [2 = 0] \text{ (mod } p) \longleftrightarrow p = 1 \vee p = 2$  **for**  $p :: \text{int}$   
(proof)

**lemma** *QuadRes-2-2* [simp, intro]: *QuadRes 2 2*  
(proof)

**lemma** *Suc-mod-eq*[simp]:  $[\text{Suc } a = \text{Suc } b] \text{ (mod } 2) = [a = b] \text{ (mod } 2)$   
(proof)

**lemma** *div-cancel-aux*:  $c \text{ dvd } a \implies (d + a * b) \text{ div } c = (d \text{ div } c) + a \text{ div } c * b$   
**for**  $a b c :: \text{nat}$   
(proof)

**corollary** *div-cancel-Suc*:  $c \text{ dvd } a \implies 1 < c \implies \text{Suc } (a * b) \text{ div } c = a \text{ div } c * b$   
(proof)

**lemma** *cong-aux-eq-1*:  $\text{odd } p \implies [(p - 1) \text{ div } 2 - p \text{ div } 4 = (p^2 - 1) \text{ div } 8]$   
 $\text{(mod } 2)$  **for**  $p :: \text{nat}$   
(proof)

**lemma** *cong-2-pow*[intro]:  $(-1 :: \text{int})^a = (-1)^b$  **if**  $[a = b] \text{ (mod } 2)$  **for**  $a b :: \text{nat}$   
(proof)

**lemma** *card-Int*:  $\text{card } (A \cap B) = \text{card } A - \text{card } (A - B)$  **if** *finite*  $A$   
(proof)

Proofs are inspired by [3].

**theorem** *supplement2-Legendre*:

**fixes**  $p :: \text{int}$   
**assumes**  $p > 2$  *prime*  $p$   
**shows**  $\text{Legendre } 2 p = (-1)^{((\text{nat } p)^2 - 1) \text{ div } 8}$   
(proof)

**theorem** *supplement1-Legendre*:

*prime*  $p \implies 2 < p \implies \text{Legendre } (-1) p = (-1)^{(p-1) \text{ div } 2}$   
(proof)

**lemma** *QuadRes-1-right* [intro, simp]: *QuadRes p 1*

*<proof>*

**lemma** *Legendre-1-left* [simp]:  $\text{prime } p \implies \text{Legendre } 1 \ p = 1$   
*<proof>*

**lemma** *cong-eq-0-not-coprime*:  $\text{prime } p \implies [a = 0] \ (\text{mod } p) \implies \neg \text{coprime } a \ p$  **for**  
 $a \ p :: \text{int}$   
*<proof>*

**lemma** *not-coprime-cong-eq-0*:  $\text{prime } p \implies \neg \text{coprime } a \ p \implies [a = 0] \ (\text{mod } p)$   
**for**  $a \ p :: \text{int}$   
*<proof>*

**lemma** *prime-cong-eq-0-iff*:  $\text{prime } p \implies [a = 0] \ (\text{mod } p) \longleftrightarrow \neg \text{coprime } a \ p$  **for**  
 $a \ p :: \text{int}$   
*<proof>*

**lemma** *Legendre-eq-0-iff* [simp]:  $\text{prime } p \implies \text{Legendre } a \ p = 0 \longleftrightarrow \neg \text{coprime } a \ p$   
*<proof>*

**lemma** *Legendre-prod-mset* [simp]:  $\text{prime } p \implies \text{Legendre } (\text{prod-mset } M) \ p =$   
 $(\prod_{q \in \#M. \text{Legendre } q \ p)$   
*<proof>*

**lemma** *Legendre-0-eq-0* [simp]:  $\text{Legendre } 0 \ p = 0$  *<proof>*

**lemma** *Legendre-values*:  $\text{Legendre } p \ q \in \{1, -1, 0\}$   
*<proof>*

**end**

## 2 Auxiliary Material

**theory** *Algebraic-Auxiliaries*

**imports**

*HOL-Algebra.Algebra*

*HOL-Computational-Algebra.Squarefree*

*HOL-Number-Theory.Number-Theory*

**begin**

**hide-const** (**open**) *Divisibility.prime*

**lemma** *sum-of-bool-eq-card*:

**assumes** *finite S*

**shows**  $(\sum a \in S. \text{of-bool } (P \ a)) = \text{real } (\text{card } \{a \in S . P \ a \})$

*<proof>*

**lemma** *mod-natE*:

**fixes**  $a \ n \ b :: \text{nat}$

**assumes**  $a \bmod n = b$   
**shows**  $\exists l. a = n * l + b$   
 $\langle proof \rangle$

**lemma** (**in group**) *r-coset-is-image*:  $H \#> a = (\lambda x. x \otimes a) \text{ ` } H$   
 $\langle proof \rangle$

**lemma** (**in group**) *FactGroup-order*:  
**assumes** *subgroup H G finite H*  
**shows**  $order\ G = order\ (G\ Mod\ H) * card\ H$   
 $\langle proof \rangle$

**corollary** (**in group**) *FactGroup-order-div*:  
**assumes** *subgroup H G finite H*  
**shows**  $order\ (G\ Mod\ H) = order\ G\ div\ card\ H$   
 $\langle proof \rangle$

**lemma** *group-hom-imp-group-hom-image*:  
**assumes** *group-hom G G h*  
**shows**  $group-hom\ G\ (G\ (\backslash carrier := h \text{ ` } carrier\ G))\ h$   
 $\langle proof \rangle$

**theorem** *homomorphism-thm*:  
**assumes** *group-hom G G h*  
**shows**  $G\ Mod\ kernel\ G\ (G\ (\backslash carrier := h \text{ ` } carrier\ G))\ h \cong G\ (\backslash carrier := h \text{ ` } carrier\ G)$   
 $\langle proof \rangle$

**lemma** *is-iso-imp-same-card*:  
**assumes**  $H \cong G$   
**shows**  $order\ H = order\ G$   
 $\langle proof \rangle$

**corollary** *homomorphism-thm-order*:  
**assumes** *group-hom G G h*  
**shows**  $order\ (G\ (\backslash carrier := h \text{ ` } carrier\ G)) * card\ (kernel\ G\ (G\ (\backslash carrier := h \text{ ` } carrier\ G))\ h) = order\ G$   
 $\langle proof \rangle$

**lemma** (**in group-hom**) *kernel-subset*:  $kernel\ G\ H\ h \subseteq carrier\ G$   
 $\langle proof \rangle$

**lemma** (**in group**) *proper-subgroup-imp-bound-on-card*:  
**assumes**  $H \subset carrier\ G$  *subgroup H G finite (carrier G)*  
**shows**  $card\ H \leq order\ G\ div\ 2$   
 $\langle proof \rangle$

**lemma** *cong-exp-trans[trans]*:  
 $[a \wedge b = c] (mod\ n) \implies [a = d] (mod\ n) \implies [d \wedge b = c] (mod\ n)$

$$[c = a \wedge b] \text{ (mod } n) \implies [a = d] \text{ (mod } n) \implies [c = d \wedge b] \text{ (mod } n)$$

*<proof>*

**lemma** *cong-exp-mod*[simp]:

$$[(a \text{ mod } n) \wedge b = c] \text{ (mod } n) \longleftrightarrow [a \wedge b = c] \text{ (mod } n)$$

$$[c = (a \text{ mod } n) \wedge b] \text{ (mod } n) \longleftrightarrow [c = a \wedge b] \text{ (mod } n)$$

*<proof>*

**lemma** *cong-mult-mod*[simp]:

$$[(a \text{ mod } n) * b = c] \text{ (mod } n) \longleftrightarrow [a * b = c] \text{ (mod } n)$$

$$[a * (b \text{ mod } n) = c] \text{ (mod } n) \longleftrightarrow [a * b = c] \text{ (mod } n)$$

*<proof>*

**lemma** *cong-add-mod*[simp]:

$$[(a \text{ mod } n) + b = c] \text{ (mod } n) \longleftrightarrow [a + b = c] \text{ (mod } n)$$

$$[a + (b \text{ mod } n) = c] \text{ (mod } n) \longleftrightarrow [a + b = c] \text{ (mod } n)$$

$$[\sum_{i \in A} f i \text{ mod } n = c] \text{ (mod } n) \longleftrightarrow [\sum_{i \in A} f i = c] \text{ (mod } n)$$

*<proof>*

**lemma** *cong-add-trans*[trans]:

$$[a = b + x] \text{ (mod } n) \implies [x = y] \text{ (mod } n) \implies [a = b + y] \text{ (mod } n)$$

$$[a = x + b] \text{ (mod } n) \implies [x = y] \text{ (mod } n) \implies [a = y + b] \text{ (mod } n)$$

$$[b + x = a] \text{ (mod } n) \implies [x = y] \text{ (mod } n) \implies [b + y = a] \text{ (mod } n)$$

$$[x + b = a] \text{ (mod } n) \implies [x = y] \text{ (mod } n) \implies [y + b = a] \text{ (mod } n)$$

*<proof>*

**lemma** *cong-mult-trans*[trans]:

$$[a = b * x] \text{ (mod } n) \implies [x = y] \text{ (mod } n) \implies [a = b * y] \text{ (mod } n)$$

$$[a = x * b] \text{ (mod } n) \implies [x = y] \text{ (mod } n) \implies [a = y * b] \text{ (mod } n)$$

$$[b * x = a] \text{ (mod } n) \implies [x = y] \text{ (mod } n) \implies [b * y = a] \text{ (mod } n)$$

$$[x * b = a] \text{ (mod } n) \implies [x = y] \text{ (mod } n) \implies [y * b = a] \text{ (mod } n)$$

*<proof>*

**lemma** *cong-diff-trans*[trans]:

$$[a = b - x] \text{ (mod } n) \implies [x = y] \text{ (mod } n) \implies [a = b - y] \text{ (mod } n)$$

$$[a = x - b] \text{ (mod } n) \implies [x = y] \text{ (mod } n) \implies [a = y - b] \text{ (mod } n)$$

$$[b - x = a] \text{ (mod } n) \implies [x = y] \text{ (mod } n) \implies [b - y = a] \text{ (mod } n)$$

$$[x - b = a] \text{ (mod } n) \implies [x = y] \text{ (mod } n) \implies [y - b = a] \text{ (mod } n)$$

**for**  $a :: 'a :: \{\text{unique-euclidean-semiring, euclidean-ring-cancel}\}$

*<proof>*

**lemma** *eq-imp-eq-mod-int*:  $a = b \implies [a = b] \text{ (mod } m)$  **for**  $a b :: \text{int}$  *<proof>*

**lemma** *eq-imp-eq-mod-nat*:  $a = b \implies [a = b] \text{ (mod } m)$  **for**  $a b :: \text{nat}$  *<proof>*

**lemma** *cong-pow-I*:  $a = b \implies [x^a = x^b] \text{ (mod } n)$  *<proof>*

**lemma** *gre1I*:  $(n = 0 \implies \text{False}) \implies (1 :: \text{nat}) \leq n$

*<proof>*

**lemma** *gre1I-nat*:  $(n = 0 \implies \text{False}) \implies (\text{Suc } 0 :: \text{nat}) \leq n$   
*<proof>*

**lemma** *totient-less-not-prime*:  
**assumes**  $\neg \text{prime } n$   $1 < n$   
**shows**  $\text{totient } n < n - 1$   
*<proof>*

**lemma** *power2-diff-nat*:  $x \geq y \implies (x - y)^2 = x^2 + y^2 - 2 * x * y$  **for**  $x y :: \text{nat}$   
*<proof>*

**lemma** *square-inequality*:  $1 < n \implies (n + n) \leq (n * n)$  **for**  $n :: \text{nat}$   
*<proof>*

**lemma** *square-one-cong-one*:  
**assumes**  $[x = 1](\text{mod } n)$   
**shows**  $[x^2 = 1](\text{mod } n)$   
*<proof>*

**lemma** *cong-square-alt-int*:  
 $\text{prime } p \implies [a * a = 1](\text{mod } p) \implies [a = 1](\text{mod } p) \vee [a = p - 1](\text{mod } p)$   
**for**  $a p :: 'a :: \{\text{normalization-semidom, linordered-idom, unique-euclidean-ring}\}$   
*<proof>*

**lemma** *cong-square-alt*:  
 $\text{prime } p \implies [a * a = 1](\text{mod } p) \implies [a = 1](\text{mod } p) \vee [a = p - 1](\text{mod } p)$   
**for**  $a p :: \text{nat}$   
*<proof>*

**lemma** *square-minus-one-cong-one*:  
**fixes**  $n x :: \text{nat}$   
**assumes**  $1 < n$   $[x = n - 1](\text{mod } n)$   
**shows**  $[x^2 = 1](\text{mod } n)$   
*<proof>*

**lemma** *odd-prime-gt-2-int*:  
 $2 < p$  **if**  $\text{odd } p$   $\text{prime } p$  **for**  $p :: \text{int}$   
*<proof>*

**lemma** *odd-prime-gt-2-nat*:  
 $2 < p$  **if**  $\text{odd } p$   $\text{prime } p$  **for**  $p :: \text{nat}$   
*<proof>*

**lemma** *gt-one-imp-gt-one-power-if-coprime*:  
 $1 \leq x \implies 1 < n \implies \text{coprime } x n \implies 1 \leq x^{(n - 1)} \text{ mod } n$   
*<proof>*

**lemma** *residue-one-dvd*:  $a \text{ mod } n = 1 \implies n \text{ dvd } a - 1$  **for**  $a n :: \text{nat}$   
*<proof>*



**lemma** *coprimeI-power-mod*:

**fixes**  $x r n :: \text{nat}$

**assumes**  $x \wedge r \bmod n = 1 \ r \neq 0 \ n \neq 0$

**shows** *coprime*  $x \ n$

*<proof>*

**lemma** *prime-dvd-choose*:

**assumes**  $0 < k \ k < p \ \text{prime } p$

**shows**  $p \ \text{dvd } (p \ \text{choose } k)$

*<proof>*

**lemma** *cong-eq-0-I*:  $(\forall i \in A. [f \ i \ \text{mod } n = 0] \ (\text{mod } n)) \implies [\sum i \in A. f \ i = 0] \ (\text{mod } n)$

*<proof>*

**lemma** *power-mult-cong*:

**assumes**  $[x \wedge n = a] \ (\text{mod } m) \ [y \wedge n = b] \ (\text{mod } m)$

**shows**  $[(x * y) \wedge n = a * b] \ (\text{mod } m)$

*<proof>*

**lemma**

**fixes**  $n :: \text{nat}$

**assumes**  $n > 1$

**shows** *odd-pow-cong*:  $\text{odd } m \implies [(n - 1) \wedge m = n - 1] \ (\text{mod } n)$

**and** *even-pow-cong*:  $\text{even } m \implies [(n - 1) \wedge m = 1] \ (\text{mod } n)$

*<proof>*

**lemma** *cong-mult-uneq'*:

**fixes**  $a :: 'a :: \{\text{unique-euclidean-ring, ring-gcd}\}$

**assumes** *coprime*  $d \ a$

**shows**  $[b \neq c] \ (\text{mod } a) \implies [d = e] \ (\text{mod } a) \implies [b * d \neq c * e] \ (\text{mod } a)$

*<proof>*

**lemma** *p-coprime-right-nat*:  $\text{prime } p \implies \text{coprime } a \ p = (\neg p \ \text{dvd } a)$  **for**  $p \ a :: \text{nat}$

*<proof>*

**lemma** *squarefree-mult-imp-coprime*:

**assumes** *squarefree*  $(a * b :: 'a :: \text{semiring-gcd})$

**shows** *coprime*  $a \ b$

*<proof>*

**lemma** *prime-divisor-exists-strong*:

**fixes**  $m :: \text{int}$

**assumes**  $m > 1 \ \neg \text{prime } m$

**shows**  $\exists n \ k. m = n * k \wedge 1 < n \wedge n < m \wedge 1 < k \wedge k < m$

*<proof>*

**lemma** *prime-divisor-exists-strong-nat*:

**fixes**  $m :: \text{nat}$

**assumes**  $1 < m \neg\text{prime } m$

**shows**  $\exists p k. m = p * k \wedge 1 < p \wedge p < m \wedge 1 < k \wedge k < m \wedge \text{prime } p$

*<proof>*

**lemma** *prime-factorization-eqI*:

**assumes**  $\bigwedge p. p \in \# P \implies \text{prime } p \text{ prod-mset } P = n$

**shows**  $\text{prime-factorization } n = P$

*<proof>*

**lemma** *prime-factorization-prime-elem*:

**assumes**  $\text{prime-elem } p$

**shows**  $\text{prime-factorization } p = \{\# \text{normalize } p\}$

*<proof>*

**lemma** *size-prime-factorization-eq-Suc-0-iff [simp]*:

**fixes**  $n :: 'a :: \text{factorial-semiring-multiplicative}$

**shows**  $\text{size } (\text{prime-factorization } n) = \text{Suc } 0 \longleftrightarrow \text{prime-elem } n$

*<proof>*

**lemma** *squarefree-prime-elem [simp, intro]*:

**fixes**  $p :: 'a :: \text{algebraic-semidom}$

**assumes**  $\text{prime-elem } p$

**shows**  $\text{squarefree } p$

*<proof>*

**lemma** *squarefree-prime [simp, intro]*:  $\text{prime } p \implies \text{squarefree } p$

*<proof>*

**lemma** *not-squarefree-primelow*:

**assumes**  $\text{primelow } n$

**shows**  $\text{squarefree } n \longleftrightarrow \text{prime } n$

*<proof>*

**lemma** *prime-factorization-normalize [simp]*:

$\text{prime-factorization } (\text{normalize } n) = \text{prime-factorization } n$

*<proof>*

**lemma** *one-prime-factor-iff-primelow*:

**fixes**  $n :: 'a :: \text{factorial-semiring-multiplicative}$

**shows**  $\text{card } (\text{prime-factors } n) = \text{Suc } 0 \longleftrightarrow \text{primelow } (\text{normalize } n)$

*<proof>*

**lemma** *squarefree-imp-prod-prime-factors-eq*:  
**fixes**  $x :: 'a :: \text{factorial-semiring-multiplicative}$   
**assumes** *squarefree*  $x$   
**shows**  $\prod (\text{prime-factors } x) = \text{normalize } x$   
 $\langle \text{proof} \rangle$

**end**

### 3 The Jacobi Symbol

**theory** *Jacobi-Symbol*  
**imports**  
*Legendre-Symbol*  
*Algebraic-Auxiliaries*  
**begin**

The Jacobi symbol is a generalisation of the Legendre symbol to non-primes [4, 2]. It is defined as

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{k_1} \cdots \left(\frac{a}{p_l}\right)^{k_l}$$

where  $\left(\frac{a}{p}\right)$  denotes the Legendre symbol,  $a$  is an integer,  $n$  is an odd natural number and  $p_1^{k_1} \dots p_l^{k_l}$  is its prime factorisation.

There is, however, a fairly natural generalisation to all non-zero integers for  $n$ . It is less clear what a good choice for  $n = 0$  is; Mathematica and Maxima adopt the convention that  $\left(\frac{\pm 1}{0}\right) = 1$  and  $\left(\frac{a}{0}\right) = 0$  otherwise. However, we chose the slightly different convention  $\left(\frac{a}{0}\right) = 0$  for *all*  $a$  because then the Jacobi symbol is completely multiplicative in both arguments without any restrictions.

**definition** *Jacobi* ::  $\text{int} \Rightarrow \text{int} \Rightarrow \text{int}$  **where**  
*Jacobi*  $a$   $n = (\text{if } n = 0 \text{ then } 0 \text{ else}$   
 $(\prod_{p \in \# \text{prime-factorization } n. \text{Legendre } a } p))$

**lemma** *Jacobi-0-right* [*simp*]: *Jacobi*  $a$   $0 = 0$   
 $\langle \text{proof} \rangle$

**lemma** *Jacobi-mult-left* [*simp*]: *Jacobi*  $(a * b)$   $n = \text{Jacobi } a$   $n * \text{Jacobi } b$   $n$   
 $\langle \text{proof} \rangle$

**lemma** *Jacobi-mult-right* [*simp*]: *Jacobi*  $a$   $(n * m) = \text{Jacobi } a$   $n * \text{Jacobi } a$   $m$   
 $\langle \text{proof} \rangle$

**lemma** *prime-p-Jacobi-eq-Legendre*[*intro!*]: *prime*  $p \implies \text{Jacobi } a$   $p = \text{Legendre } a$   $p$   
 $\langle \text{proof} \rangle$

**lemma** *Jacobi-mod* [*simp*]:  $Jacobi (a \bmod m) n = Jacobi a n$  **if**  $n \text{ dvd } m$   
 ⟨*proof*⟩

**lemma** *Jacobi-mod-cong*:  $[a = b] \text{ (mod } n) \implies Jacobi a n = Jacobi b n$   
 ⟨*proof*⟩

**lemma** *Jacobi-1-eq-1* [*simp*]:  $p \neq 0 \implies Jacobi 1 p = 1$   
 ⟨*proof*⟩

**lemma** *Jacobi-eq-0-not-coprime*:  
**assumes**  $n \neq 0 \neg coprime a n$   
**shows**  $Jacobi a n = 0$   
 ⟨*proof*⟩

**lemma** *Jacobi-p-eq-2* [*simp*]:  $n > 0 \implies Jacobi a (2^n) = a \bmod 2$   
 ⟨*proof*⟩

**lemma** *Jacobi-prod-mset* [*simp*]:  $n \neq 0 \implies Jacobi (\text{prod-mset } M) n = (\prod_{q \in \#M} Jacobi q n)$   
 ⟨*proof*⟩

**lemma** *non-trivial-coprime-neg*:  
 $1 < a \implies 1 < b \implies coprime a b \implies a \neq b$  **for**  $a b :: int$  ⟨*proof*⟩

**lemma** *odd-odd-even*:  
**fixes**  $a b :: int$   
**assumes**  $odd a odd b$   
**shows**  $even ((a*b-1) \text{ div } 2) = even ((a-1) \text{ div } 2 + (b-1) \text{ div } 2)$   
 ⟨*proof*⟩

**lemma** *prime-nonprime-wlog* [*case-names primes nonprime sym*]:  
**assumes**  $\bigwedge p q. prime p \implies prime q \implies P p q$   
**assumes**  $\bigwedge p q. \neg prime p \implies P p q$   
**assumes**  $\bigwedge p q. P p q \implies P q p$   
**shows**  $P p q$   
 ⟨*proof*⟩

**lemma** *Quadratic-Reciprocity-Jacobi*:  
**fixes**  $p q :: int$   
**assumes**  $coprime p q$   
**and**  $2 < p 2 < q$   
**and**  $odd p odd q$   
**shows**  $Jacobi p q * Jacobi q p = (-1)^{\wedge (nat ((p-1) \text{ div } 2 * ((q-1) \text{ div } 2))})$   
 ⟨*proof*⟩

**lemma** *Jacobi-values*:  $Jacobi p q \in \{1, -1, 0\}$   
 ⟨*proof*⟩

**lemma** *Quadratic-Reciprocity-Jacobi'*:

**fixes**  $p\ q :: int$   
**assumes** *coprime*  $p\ q$   
**and**  $2 < p\ 2 < q$   
**and** *odd*  $p\ odd\ q$   
**shows**  $Jacobi\ q\ p = (if\ p\ mod\ 4 = 3 \wedge q\ mod\ 4 = 3\ then\ -1\ else\ 1) * Jacobi\ p\ q$   
*<proof>*

**lemma** *dvd-odd-square: 8 dvd a<sup>2</sup> - 1 if odd a for a :: int*  
*<proof>*

**lemma** *odd-odd-even'*:

**fixes**  $a\ b :: int$   
**assumes** *odd*  $a\ odd\ b$   
**shows**  $even\ (((a * b)^2 - 1)\ div\ 8) \longleftrightarrow even\ (((a^2 - 1)\ div\ 8) + ((b^2 - 1)\ div\ 8))$   
*<proof>*

**lemma** *odd-odd-even-nat'*:

**fixes**  $a\ b :: nat$   
**assumes** *odd*  $a\ odd\ b$   
**shows**  $even\ (((a * b)^2 - 1)\ div\ 8) \longleftrightarrow even\ (((a^2 - 1)\ div\ 8) + ((b^2 - 1)\ div\ 8))$   
*<proof>*

**lemma** *supplement2-Jacobi: odd p  $\implies p > 1 \implies Jacobi\ 2\ p = (-1)^{((nat\ p)^2 - 1)\ div\ 8}$*   
*<proof>*

**lemma** *mod-nat-wlog [consumes 1, case-names modulo]*:

**fixes**  $P :: nat \Rightarrow bool$   
**assumes**  $b > 0$   
**assumes**  $\bigwedge k. k \in \{0..<b\} \implies n\ mod\ b = k \implies P\ n$   
**shows**  $P\ n$   
*<proof>*

**lemma** *mod-int-wlog [consumes 1, case-names modulo]*:

**fixes**  $P :: int \Rightarrow bool$   
**assumes**  $b > 0$   
**assumes**  $\bigwedge k. 0 \leq k \implies k < b \implies n\ mod\ b = k \implies P\ n$   
**shows**  $P\ n$   
*<proof>*

**lemma** *supplement2-Jacobi'*:

**assumes** *odd*  $p$  **and**  $p > 1$   
**shows**  $Jacobi\ 2\ p = (if\ p\ mod\ 8 = 1 \vee p\ mod\ 8 = 7\ then\ 1\ else\ -1)$   
*<proof>*

**theorem** *supplement1-Jacobi*:

$odd\ p \implies 1 < p \implies Jacobi\ (-1)\ p = (-1)^{\wedge(nat\ ((p - 1)\ div\ 2))}$   
*<proof>*

**theorem** *supplement1-Jacobi'*:

$odd\ n \implies 1 < n \implies Jacobi\ (-1)\ n = (if\ n\ mod\ 4 = 1\ then\ 1\ else\ -1)$   
*<proof>*

**lemma** *Jacobi-0-eq-0*:  $\neg is-unit\ n \implies Jacobi\ 0\ n = 0$

*<proof>*

**lemma** *is-unit-Jacobi-aux*:  $is-unit\ x \implies Jacobi\ a\ x = 1$

*<proof>*

**lemma** *is-unit-Jacobi[simp]*:  $Jacobi\ a\ 1 = 1\ Jacobi\ a\ (-1) = 1$

*<proof>*

**lemma** *Jacobi-neg-right [simp]*:

$Jacobi\ a\ (-n) = Jacobi\ a\ n$

*<proof>*

**lemma** *Jacobi-neg-left*:

**assumes**  $odd\ n\ 1 < n$

**shows**  $Jacobi\ (-a)\ n = (if\ n\ mod\ 4 = 1\ then\ 1\ else\ -1) * Jacobi\ a\ n$

*<proof>*

**function** *jacobi-code* ::  $int \Rightarrow int \Rightarrow int$  **where**

*jacobi-code*  $a\ n = ($

$\quad if\ n = 0\ then\ 0$

$\quad else\ if\ n = 1\ then\ 1$

$\quad else\ if\ a = 1\ then\ 1$

$\quad else\ if\ n < 0\ then\ jacobi-code\ a\ (-n)$

$\quad else\ if\ even\ n\ then\ if\ even\ a\ then\ 0\ else\ jacobi-code\ a\ (n\ div\ 2)$

$\quad else\ if\ a < 0\ then\ (if\ n\ mod\ 4 = 1\ then\ 1\ else\ -1) * jacobi-code\ (-a)\ n$

$\quad else\ if\ a = 0\ then\ 0$

$\quad else\ if\ a \geq n\ then\ jacobi-code\ (a\ mod\ n)\ n$

$\quad else\ if\ even\ a\ \quad then\ (if\ n\ mod\ 8 \in \{1, 7\}\ then\ 1\ else\ -1) * jacobi-code\ (a\ div\ 2)\ n$

$\quad else\ if\ coprime\ a\ n\ then\ (if\ n\ mod\ 4 = 3 \wedge a\ mod\ 4 = 3\ then\ -1\ else\ 1) * jacobi-code\ n\ a$

$\quad else\ 0)$

*<proof>*

**termination**

*<proof>*

**lemmas** [*simp del*] = *jacobi-code.simps*

**lemma** *Jacobi-code [code]*:  $Jacobi\ a\ n = jacobi-code\ a\ n$

*<proof>*

**lemma** *Jacobi-eq-0-imp-not-coprime:*

**assumes**  $p \neq 0$   $p \neq 1$

**shows**  $Jacobi\ n\ p = 0 \implies \neg coprime\ n\ p$

*<proof>*

**lemma** *Jacobi-eq-0-iff-not-coprime:*

**assumes**  $p \neq 0$   $p \neq 1$

**shows**  $Jacobi\ n\ p = 0 \iff \neg coprime\ n\ p$

*<proof>*

**end**

## 4 Residue Rings of Natural Numbers

**theory** *Residues-Nat*

**imports** *Algebraic-Auxiliaries*

**begin**

### 4.1 The multiplicative group of residues modulo $n$

**definition** *Residues-Mult* ::  $'a :: \{linordered-semidom, euclidean-semiring\} \Rightarrow 'a$   
*monoid* **where**

*Residues-Mult*  $p =$

$(\langle carrier = \{x \in \{1..p\} . coprime\ x\ p\}, monoid.mult = \lambda x\ y. x * y \bmod p, one = 1 \rangle)$

**locale** *residues-mult-nat* =

**fixes**  $n :: nat$  **and**  $G$

**assumes** *n-gt-1*:  $n > 1$

**defines**  $G \equiv Residues-Mult\ n$

**begin**

**lemma** *carrier-eq* [*simp*]: *carrier*  $G = totatives\ n$

**and** *mult-eq* [*simp*]:  $(x \otimes_G y) = (x * y) \bmod n$

**and** *one-eq* [*simp*]:  $\mathbf{1}_G = 1$

*<proof>*

**lemma** *mult-eq'*:  $(\otimes_G) = (\lambda x\ y. (x * y) \bmod n)$

*<proof>*

**sublocale** *group*  $G$

*<proof>*

**sublocale** *comm-group*

*<proof>*

**lemma** *nat-pow-eq* [*simp*]:  $x [\frown]_G (k :: nat) = (x \wedge k) \bmod n$

*<proof>*

**lemma** *nat-pow-eq'*:  $([\wedge]_G) = (\lambda x k. (x \wedge k) \bmod n)$   
*<proof>*

**lemma** *order-eq*:  $\text{order } G = \text{totient } n$   
*<proof>*

**lemma** *order-less*:  $\neg \text{prime } n \implies \text{order } G < n - 1$   
*<proof>*

**lemma** *ord-residue-mult-group*:  
  **assumes**  $a \in \text{totatives } n$   
  **shows**  $\text{local.ord } a = \text{Pocklington.ord } n a$   
*<proof>*

**end**

## 4.2 The ring of residues modulo $n$

**definition** *Residues-nat* ::  $\text{nat} \Rightarrow \text{nat ring}$  **where**

*Residues-nat*  $m = (\text{carrier} = \{0..<m\}, \text{monoid.mult} = \lambda x y. (x * y) \bmod m, \text{one} = 1,$   
 $\text{ring.zero} = 0, \text{add} = \lambda x y. (x + y) \bmod m)$

**locale** *residues-nat* =  
  **fixes**  $n :: \text{nat}$  **and**  $R$   
  **assumes** *n-gt-1*:  $n > 1$   
  **defines**  $R \equiv \text{Residues-nat } n$   
**begin**

**lemma** *carrier-eq* [*simp*]:  $\text{carrier } R = \{0..<n\}$   
  **and** *mult-eq* [*simp*]:  $x \otimes_R y = (x * y) \bmod n$   
  **and** *add-eq* [*simp*]:  $x \oplus_R y = (x + y) \bmod n$   
  **and** *one-eq* [*simp*]:  $\mathbf{1}_R = 1$   
  **and** *zero-eq* [*simp*]:  $\mathbf{0}_R = 0$   
*<proof>*

**lemma** *mult-eq'*:  $(\otimes_R) = (\lambda x y. (x * y) \bmod n)$   
  **and** *add-eq'*:  $(\oplus_R) = (\lambda x y. (x + y) \bmod n)$   
*<proof>*

**sublocale** *abelian-group*  $R$   
*<proof>*

**sublocale** *comm-monoid*  $R$   
*<proof>*

**sublocale** *cring*  $R$



*<proof>*

**lemma** *Units-eq*:  $Units\ R = totatives\ n$   
*<proof>*

**sublocale** *units: residues-mult-nat n units-of R*  
*<proof>*

**lemma** *nat-pow-eq* [*simp*]:  $x \ [\frown]_R\ (k :: nat) = (x \wedge k) \bmod n$   
*<proof>*

**lemma** *nat-pow-eq'*:  $([\frown]_R) = (\lambda x\ k. (x \wedge k) \bmod n)$   
*<proof>*

**end**

### 4.3 The ring of residues modulo a prime

**locale** *residues-nat-prime* =  
  **fixes**  $p :: nat$  **and**  $R$   
  **assumes** *prime-p*: *prime p*  
  **defines**  $R \equiv Residues\text{-}nat\ p$   
**begin**

**sublocale** *residues-nat p R*  
*<proof>*

**lemma** *carrier-eq'* [*simp*]:  $totatives\ p = \{0 <..<p\}$   
*<proof>*

**lemma** *order-eq*:  $order\ (units\text{-}of\ R) = p - 1$   
*<proof>*

**lemma** *order-eq'* [*simp*]:  $totient\ p = p - 1$   
*<proof>*

**sublocale** *field R*  
*<proof>*

**lemma** *residues-prime-cyclic*:  $\exists x \in \{0 <..<p\}. \{0 <..<p\} = \{y. \exists i. y = x \wedge i \bmod p\}$   
*<proof>*

**lemma** *residues-prime-cyclic'*:  $\exists x \in \{0 <..<p\}. units.ord\ x = p - 1$   
*<proof>*

**end**

## 4.4 $-1$ in residue rings

**lemma** *minus-one-cong-solve-weak*:

```
fixes n x :: nat
assumes 1 < n x ∈ totatives n y ∈ totatives n
  and [x = n - 1] (mod n) [x * y = 1] (mod n)
shows y = n - 1
⟨proof⟩
```

**lemma** *coprime-imp-mod-not-zero*:

```
fixes n x :: nat
assumes 1 < n coprime x n
shows 0 < x mod n
⟨proof⟩
```

**lemma** *minus-one-cong-solve*:

```
fixes n x :: nat
assumes 1 < n
  and eq: [x = n - 1] (mod n) [x * y = 1] (mod n)
  and coprime: coprime x n coprime y n
shows [y = n - 1](mod n)
⟨proof⟩
```

**corollary** *square-minus-one-cong-one'*:

```
fixes n x :: nat
assumes 1 < n
shows [(n - 1) * (n - 1) = 1](mod n)
⟨proof⟩
```

end

## 5 Additional Material on Quadratic Residues

**theory** *QuadRes*

**imports**

*Jacobi-Symbol*

*Algebraic-Auxiliaries*

**begin**

Proofs are inspired by [5].

**lemma** *inj-on-QuadRes*:

```
fixes p :: int
assumes prime p
shows inj-on (λx. x2 mod p) {0..(p-1) div 2}
⟨proof⟩
```

**lemma** *QuadRes-set-prime*:

```
assumes prime p and odd p
shows {x . QuadRes p x ∧ x ∈ {0..<p}} = {x2 mod p | x . x ∈ {0..(p-1) div
```

2}}  
<proof>

**corollary** *QuadRes-iff*:

**assumes** *prime p and odd p*  
**shows**  $(\text{QuadRes } p \ x \wedge x \in \{0..<p\}) \longleftrightarrow (\exists a \in \{0..(p-1) \text{ div } 2\}. a^2 \text{ mod } p = x)$   
<proof>

**corollary** *card-QuadRes-set-prime*:

**fixes**  $p :: \text{int}$   
**assumes** *prime p and odd p*  
**shows**  $\text{card } \{x. \text{QuadRes } p \ x \wedge x \in \{0..<p\}\} = \text{nat } (p+1) \text{ div } 2$   
<proof>

**corollary** *card-not-QuadRes-set-prime*:

**fixes**  $p :: \text{int}$   
**assumes** *prime p and odd p*  
**shows**  $\text{card } \{x. \neg \text{QuadRes } p \ x \wedge x \in \{0..<p\}\} = \text{nat } (p-1) \text{ div } 2$   
<proof>

**lemma** *not-QuadRes-ex-if-prime*:

**assumes** *prime p and odd p*  
**shows**  $\exists x. \neg \text{QuadRes } p \ x$   
<proof>

**lemma** *not-QuadRes-ex*:

$1 < p \implies \text{odd } p \implies \exists x. \neg \text{QuadRes } p \ x$   
<proof>

**end**

## 6 Euler Witnesses

**theory** *Euler-Witness*

**imports**

*Jacobi-Symbol*

*Residues-Nat*

*QuadRes*

**begin**

Proofs are inspired by [13, 8, 15, 11].

**definition** *euler-witness*  $a \ p \longleftrightarrow [\text{Jacobi } a \ p \neq a^{\wedge((p-1) \text{ div } 2)}] \text{ (mod } p)$

**abbreviation** *euler-liar*  $a \ p \equiv \neg \text{euler-witness } a \ p$

**lemma** *euler-witness-mod[simp]*:  $\text{euler-witness } (a \text{ mod } p) \ p = \text{euler-witness } a \ p$   
<proof>

**lemma** *euler-liar-mod*:  $\text{euler-liar } (a \text{ mod } p) \ p = \text{euler-liar } a \ p$  <proof>

**lemma** *euler-liar-cong*:

**assumes**  $[a = b] \pmod{p}$   
**shows**  $\text{euler-liar } a \ p = \text{euler-liar } b \ p$   
*<proof>*

**lemma** *euler-witnessI*:

$[x \wedge ((n - 1) \text{ div } 2) = a] \pmod{\text{int } n} \implies [\text{Jacobi } x \ (\text{int } n) \neq a] \pmod{\text{int } n}$   
 $\implies \text{euler-witness } x \ n$   
*<proof>*

**lemma** *euler-witnessI2*:

**fixes**  $a \ b :: \text{int}$  **and**  $k :: \text{nat}$   
**assumes**  $[a \neq b] \pmod{k}$   
**and**  $k \ \text{dvd } n$   
**and** *main*:  $\text{euler-liar } x \ n \implies [\text{Jacobi } x \ n = a] \pmod{k}$   
 $\text{euler-liar } x \ n \implies [x \wedge ((n - 1) \text{ div } 2) = b] \pmod{k}$   
**shows**  $\text{euler-witness } x \ n$   
*<proof>*

**lemma** *euler-witness-exists-weak*:

**assumes**  $\text{odd } n \ \neg \text{prime } n \ 2 < n$   
**shows**  $\exists a. \text{euler-witness } a \ n \wedge \text{coprime } a \ n$   
*<proof>*

**lemma** *euler-witness-exists*:

**assumes**  $\text{odd } n \ \neg \text{prime } n \ 2 < n$   
**shows**  $\exists a. \text{euler-witness } a \ n \wedge \text{coprime } a \ n \wedge 0 < a \wedge a < n$   
*<proof>*

**lemma** *euler-witness-exists-nat*:

**assumes**  $\text{odd } n \ \neg \text{prime } n \ 2 < n$   
**shows**  $\exists a. \text{euler-witness } (\text{int } a) \ n \wedge \text{coprime } a \ n \wedge 0 < a \wedge a < n$   
*<proof>*

**lemma** *euler-liar-1-p*[*intro, simp*]:  $p \neq 0 \implies \text{euler-liar } 1 \ p$

*<proof>*

**lemma** *euler-liar-mult*:

**shows**  $\text{euler-liar } y \ n \implies \text{euler-liar } x \ n \implies \text{euler-liar } (x*y) \ n$   
*<proof>*

**lemma** *euler-liar-mult'*:

**assumes**  $1 < n \ \text{coprime } y \ n$   
**shows**  $\text{euler-liar } y \ n \implies \text{euler-witness } x \ n \implies \text{euler-witness } (x*y) \ n$   
*<proof>*

**lemma** *prime-imp-euler-liar*:

**assumes**  $\text{prime } p \ 2 < p \ 0 < x \ x < p$

**shows** *euler-liar*  $x\ p$   
 $\langle$ *proof* $\rangle$

**locale** *euler-witness-context* =  
**fixes**  $p :: \text{nat}$   
**assumes**  $p\text{-gt-1}: p > 1$  **and**  $\text{odd-}p: \text{odd } p$   
**begin**

**definition**  $G$  **where**  $G = \text{Residues-Mult } p$

**sublocale** *residues-mult-nat*  $p\ G$   
 $\langle$ *proof* $\rangle$

**definition**  $H = \{x. \text{coprime } x\ p \wedge \text{euler-liar } (\text{int } x)\ p \wedge x \in \{1..<p\}\}$

**sublocale**  $H: \text{subgroup } H\ G$   
 $\langle$ *proof* $\rangle$

**lemma** *H-finite: finite*  $H$   
 $\langle$ *proof* $\rangle$

**lemma** *euler-witness-coset:*  
**assumes** *euler-witness*  $x\ p$   
**shows**  $y \in H \#>_G x \implies \text{euler-witness } y\ p$   
 $\langle$ *proof* $\rangle$

**lemma** *euler-liar-coset:*  
**assumes** *euler-liar*  $x\ p\ x \in \text{carrier } G$   
**shows**  $y \in H \#>_G x \implies \text{euler-liar } y\ p$   
 $\langle$ *proof* $\rangle$

**lemma** *in-cosets-aux:*  
**assumes** *euler-witness*  $x\ p\ x \in \text{carrier } G$   
**shows**  $H \#>_G x \in \text{rcosets}_G\ H$   
 $\langle$ *proof* $\rangle$

**lemma** *H-cosets-aux:*  
**assumes** *euler-witness*  $x\ p$   
**shows**  $(H \#>_G x) \cap H = \{\}$   
 $\langle$ *proof* $\rangle$

**lemma** *H-rcosets-aux:*  
**assumes** *euler-witness*  $x\ p\ x \in \text{carrier } G$   
**shows**  $\{H, H \#>_G x\} \subseteq \text{rcosets}_G\ H$   
 $\langle$ *proof* $\rangle$

**lemma** *H-not-eq-coset:*  
**assumes** *euler-witness*  $x\ p$   
**shows**  $H \neq H \#>_G x$

*<proof>*

**lemma** *finite-cosets-H*: *finite (rcosets<sub>G</sub> H)*  
*<proof>*

**lemma** *card-cosets-limit*:  
**assumes** *euler-witness x p x ∈ carrier G*  
**shows**  $2 \leq \text{card } (\text{rcosets}_G H)$   
*<proof>*

**lemma** *card-euler-liars-cosets-limit*:  
**assumes**  $\neg \text{prime } p$   $2 < p$   
**shows**  $2 \leq \text{card } (\text{rcosets}_G H)$   $\text{card } H < (p - 1) \text{ div } 2$   
*<proof>*

**lemma** *prime-imp-G-is-H*:  
**assumes** *prime p 2 < p*  
**shows** *carrier G = H*  
*<proof>*

**end**

**end**

## 7 Carmichael Numbers

**theory** *Carmichael-Numbers*  
**imports**  
  *Residues-Nat*  
**begin**

A Carmichael number is a composite number  $n$  that Fermat's test incorrectly labels as primes no matter which witness  $a$  is chosen (except in the case that  $a$  shares a factor with  $n$ ). [9, 14]

**definition** *Carmichael-number* :: *nat*  $\Rightarrow$  *bool* **where**  
  *Carmichael-number*  $n \longleftrightarrow n > 1 \wedge \neg \text{prime } n \wedge (\forall a. \text{coprime } a \ n \longrightarrow [a \wedge (n - 1) = 1] \pmod n)$

**lemma** *Carmichael-number-0*[*simp, intro*]:  $\neg \text{Carmichael-number } 0$   
*<proof>*

**lemma** *Carmichael-number-1*[*simp, intro*]:  $\neg \text{Carmichael-number } 1$   
*<proof>*

**lemma** *Carmichael-number-Suc-0*[*simp, intro*]:  $\neg \text{Carmichael-number } (\text{Suc } 0)$   
*<proof>*

**lemma** *Carmichael-number-not-prime*: *Carmichael-number*  $n \implies \neg \text{prime } n$

*<proof>*

**lemma** *Carmichael-number-gt-3*: *Carmichael-number*  $n \implies n > 3$   
*<proof>*

The proofs are inspired by [9, 12].

**lemma** *Carmichael-number-imp-squarefree-aux*:  
**assumes** *Carmichael-number*  $n$   
**assumes**  $n: n = p^r * l$  **and** *prime*  $p \neg p \text{ dvd } l$   
**assumes**  $r > 1$   
**shows** *False*  
*<proof>*

**theorem** *Carmichael-number-imp-squarefree*:  
**assumes** *Carmichael-number*  $n$   
**shows** *squarefree*  $n$   
*<proof>*

**corollary** *Carmichael-not-primelow*:  
**assumes** *Carmichael-number*  $n$   
**shows**  $\neg \text{primelow } n$   
*<proof>*

**lemma** *Carmichael-number-imp-squarefree-alt-weak*:  
**assumes** *Carmichael-number*  $n$   
**shows**  $\exists p l. (n = p * l) \wedge \text{prime } p \wedge \neg p \text{ dvd } l$   
*<proof>*

**theorem** *Carmichael-number-odd*:  
**assumes** *Carmichael-number*  $n$   
**shows** *odd*  $n$   
*<proof>*

**lemma** *Carmichael-number-imp-squarefree-alt*:  
**assumes** *Carmichael-number*  $n$   
**shows**  $\exists p l. (n = p * l) \wedge \text{prime } p \wedge \neg p \text{ dvd } l \wedge 2 < l$   
*<proof>*

**lemma** *Carmichael-number-imp-dvd*:  
**fixes**  $n :: \text{nat}$   
**assumes** *Carmichael-number*: *Carmichael-number*  $n$  **and** *prime*  $p \text{ dvd } n$   
**shows**  $p - 1 \text{ dvd } n - 1$   
*<proof>*

The following lemma is also called Korselt's criterion.

**lemma** *Carmichael-numberI*:  
**fixes**  $n :: \text{nat}$   
**assumes**  $\neg \text{prime } n \text{ squarefree } n 1 < n$  **and**  
 $\text{DIV: } \bigwedge p. p \in \text{prime-factors } n \implies p - 1 \text{ dvd } n - 1$

**shows** *Carmichael-number*  $n$   
 ⟨*proof*⟩

**theorem** *Carmichael-number-iff*:

*Carmichael-number*  $n \iff$   
 $n \neq 1 \wedge \neg \text{prime } n \wedge \text{squarefree } n \wedge (\forall p \in \text{prime-factors } n. p - 1 \text{ dvd } n - 1)$   
 ⟨*proof*⟩

Every Carmichael number has at least three distinct prime factors.

**theorem** *Carmichael-number-card-prime-factors*:

**assumes** *Carmichael-number*  $n$   
**shows**  $\text{card } (\text{prime-factors } n) \geq 3$   
 ⟨*proof*⟩

**lemma** *Carmichael-number-iff'*:

**fixes**  $n :: \text{nat}$   
**defines**  $P \equiv \text{prime-factorization } n$   
**shows** *Carmichael-number*  $n \iff$   
 $n > 1 \wedge \text{size } P \neq 1 \wedge (\forall p \in \#P. \text{count } P p = 1 \wedge p - 1 \text{ dvd } n - 1)$   
 ⟨*proof*⟩

The smallest Carmichael number is 561, and it was found and proven so by Carmichael in 1910 [6].

**lemma** *Carmichael-number-561*: *Carmichael-number 561 (is Carmichael-number ?n)*  
 ⟨*proof*⟩

**end**

## 8 Fermat Witnesses

**theory** *Fermat-Witness*

**imports** *Euler-Witness Carmichael-Numbers*  
**begin**

**definition** *divide-out* ::  $'a :: \text{factorial-semiring} \Rightarrow 'a \Rightarrow 'a \times \text{nat}$  **where**  
 $\text{divide-out } p x = (x \text{ div } p \wedge \text{multiplicity } p x, \text{multiplicity } p x)$

**lemma** *fst-divide-out [simp]*:  $\text{fst } (\text{divide-out } p x) = x \text{ div } p \wedge \text{multiplicity } p x$   
**and** *snd-divide-out [simp]*:  $\text{snd } (\text{divide-out } p x) = \text{multiplicity } p x$   
 ⟨*proof*⟩

**function** *divide-out-aux* ::  $'a :: \text{factorial-semiring} \Rightarrow 'a \times \text{nat} \Rightarrow 'a \times \text{nat}$  **where**  
 $\text{divide-out-aux } p (x, \text{acc}) =$   
 (if  $x = 0 \vee \text{is-unit } p \vee \neg p \text{ dvd } x$  then  $(x, \text{acc})$  else  $\text{divide-out-aux } p (x \text{ div } p,$   
 $\text{acc} + 1)$ )  
 ⟨*proof*⟩



**termination**  $\langle proof \rangle$

**lemmas**  $[simp\ del] = divide-out-aux.simps$

**lemma** *divide-out-aux-correct*:

$divide-out-aux\ p\ z = (fst\ z\ div\ p\ \wedge\ multiplicity\ p\ (fst\ z),\ snd\ z + multiplicity\ p\ (fst\ z))$   
 $\langle proof \rangle$

**lemma** *divide-out-code*  $[code]$ :  $divide-out\ p\ x = divide-out-aux\ p\ (x, 0)$

$\langle proof \rangle$

**lemma** *multiplicity-code*  $[code]$ :  $multiplicity\ p\ x = snd\ (divide-out-aux\ p\ (x, 0))$

$\langle proof \rangle$

**lemma** *multiplicity-times-same-power*:

**assumes**  $x \neq 0\ \neg is-unit\ p\ p \neq 0$

**shows**  $multiplicity\ p\ (p\ \wedge\ k * x) = multiplicity\ p\ x + k$

$\langle proof \rangle$

**lemma** *divide-out-unique-nat*:

**fixes**  $n :: nat$

**assumes**  $\neg is-unit\ p\ p \neq 0\ \neg p\ dvd\ m$  **and**  $n = p\ \wedge\ k * m$

**shows**  $m = n\ div\ p\ \wedge\ multiplicity\ p\ n$  **and**  $k = multiplicity\ p\ n$

$\langle proof \rangle$

**context**

**fixes**  $a\ n :: nat$

**begin**

**definition** *fermat-liar*  $\longleftrightarrow [a\ \wedge\ (n-1) = 1] (mod\ n)$

**definition** *fermat-witness*  $\longleftrightarrow [a\ \wedge\ (n-1) \neq 1] (mod\ n)$

**definition** *strong-fermat-liar*  $\longleftrightarrow$

$(\forall k\ m.\ odd\ m \longrightarrow n - 1 = 2\ \wedge\ k * m \longrightarrow$

$[a\ \wedge\ m = 1] (mod\ n) \vee (\exists i \in \{0..< k\}.\ [a\ \wedge\ (2\ \wedge\ i * m) = n-1] (mod\ n)))$

**definition** *strong-fermat-witness*  $\longleftrightarrow \neg\ strong-fermat-liar$

**lemma** *fermat-liar-witness-of-composition*  $[iff]$ :

$\neg\ fermat-liar = fermat-witness$

$\neg\ fermat-witness = fermat-liar$

$\langle proof \rangle$

**lemma** *strong-fermat-liar-code*  $[code]$ :

$strong-fermat-liar \longleftrightarrow$

(let (m, k) = divide-out 2 (n - 1)  
 in [a<sup>m</sup> = 1](mod n) ∨ (∃ i ∈ {0..<k}. [a<sup>(2<sup>i</sup> \* m)</sup> = n-1] (mod n)))  
 (is ?lhs = ?rhs)  
 ⟨proof⟩

**context**

**assumes** \* : a ∈ {1 ..<n}

**begin**

**lemma** strong-fermat-witness-iff:

strong-fermat-witness =  
 (∃ k m. odd m ∧ n - 1 = 2<sup>k</sup> \* m ∧ [a<sup>m</sup> ≠ 1] (mod n) ∧  
 (∀ i ∈ {0..<k}. [a<sup>(2<sup>i</sup> \* m)</sup> ≠ n-1] (mod n)))  
 ⟨proof⟩

**lemma** not-coprime-imp-witness: ¬coprime a n ⇒ fermat-witness

⟨proof⟩

**corollary** liar-imp-coprime: fermat-liar ⇒ coprime a n

⟨proof⟩

**lemma** fermat-witness-imp-strong-fermat-witness:

**assumes** 1 < n fermat-witness

**shows** strong-fermat-witness

⟨proof⟩

**corollary** strong-fermat-liar-imp-fermat-liar:

**assumes** 1 < n strong-fermat-liar

**shows** fermat-liar

⟨proof⟩

**lemma** euler-liar-is-fermat-liar:

**assumes** 1 < n euler-liar a n coprime a n odd n

**shows** fermat-liar

⟨proof⟩

**corollary** fermat-witness-is-euler-witness:

**assumes** 1 < n fermat-witness coprime a n odd n

**shows** euler-witness a n

⟨proof⟩

**end**

**end**

**lemma** one-is-fermat-liar[simp]: 1 < n ⇒ fermat-liar 1 n

⟨proof⟩

**lemma** one-is-strong-fermat-liar[simp]: 1 < n ⇒ strong-fermat-liar 1 n

*<proof>*

**lemma** *prime-imp-fermat-liar*:

*prime p*  $\implies a \in \{1 ..< p\} \implies \text{fermat-liar } a \ p$

*<proof>*

**lemma** *not-Carmichael-numberD*:

**assumes**  $\neg \text{Carmichael-number } n \ \neg \text{prime } n$  **and**  $1 < n$

**shows**  $\exists a \in \{2 ..< n\} . \text{fermat-witness } a \ n \wedge \text{coprime } a \ n$

*<proof>*

**proposition** *prime-imp-strong-fermat-witness*:

**fixes**  $p :: \text{nat}$

**assumes** *prime p*  $2 < p$   $a \in \{1 ..< p\}$

**shows** *strong-fermat-liar a p*

*<proof>*

**lemma** *ignore-one*:

**fixes**  $P :: - \implies \text{nat} \implies \text{bool}$

**assumes**  $P \ 1 \ n \ 1 < n$

**shows**  $\text{card } \{a \in \{1..<n\} . P \ a \ n\} = 1 + \text{card } \{a . 2 \leq a \wedge a < n \wedge P \ a \ n\}$

*<proof>*

Proofs in the following section are inspired by [10, 7, 1].

**proposition** *not-Carmichael-number-imp-card-fermat-witness-bound*:

**fixes**  $n :: \text{nat}$

**assumes**  $\neg \text{prime } n \ \neg \text{Carmichael-number } n \ \text{odd } n \ 1 < n$

**shows**  $(n-1) \ \text{div } 2 < \text{card } \{a \in \{1 ..< n\} . \text{fermat-witness } a \ n\}$

**and**  $(\text{card } \{a . 2 \leq a \wedge a < n \wedge \text{strong-fermat-liar } a \ n\}) < \text{real } (n-2) / 2$

**and**  $(\text{card } \{a . 2 \leq a \wedge a < n \wedge \text{fermat-liar } a \ n\}) < \text{real } (n-2) / 2$

*<proof>*

**proposition** *Carmichael-number-imp-lower-bound-on-strong-fermat-witness*:

**fixes**  $n :: \text{nat}$

**assumes** *Carmichael-number: Carmichael-number n*

**shows**  $(n-1) \ \text{div } 2 < \text{card } \{a \in \{1..<n\} . \text{strong-fermat-witness } a \ n\}$

**and**  $\text{real } (\text{card } \{a . 2 \leq a \wedge a < n \wedge \text{strong-fermat-liar } a \ n\}) < \text{real } (n-2) / 2$

*<proof>*

**corollary** *strong-fermat-witness-lower-bound*:

**assumes** *odd n n > 2*  $\neg \text{prime } n$

**shows**  $\text{card } \{a . 2 \leq a \wedge a < n \wedge \text{strong-fermat-liar } a \ n\} < \text{real } (n-2) / 2$

*<proof>*

**end**

## 9 A Generic View on Probabilistic Prime Tests

**theory** *Generalized-Primality-Test*

**imports**

*HOL-Probability.Probability*

*Algebraic-Auxiliaries*

**begin**

**definition** *primality-test* ::  $(\text{nat} \Rightarrow \text{nat} \Rightarrow \text{bool}) \Rightarrow \text{nat} \Rightarrow \text{bool}$  pmf **where**

*primality-test*  $P$   $n =$

(if  $n < 3 \vee \text{even } n$  then *return-pmf*  $(n = 2)$  else

do {

$a \leftarrow \text{pmf-of-set } \{2..< n\};$

*return-pmf*  $(P \ n \ a)$

})

**lemma** *expectation-of-bool-is-pmf*: *measure-pmf.expectation*  $M$  of-bool = pmf  $M$

*True*

*<proof>*

**lemma** *eq-bernoulli-pmfI*:

**assumes** pmf  $p$  *True* =  $x$

**shows**  $p = \text{bernoulli-pmf } x$

*<proof>*

We require a probabilistic primality test to never classify a prime as composite. It may, however, mistakenly classify composites as primes.

**locale** *prob-primality-test* =

**fixes**  $P :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{bool}$  **and**  $n :: \text{nat}$

**assumes** *P-works*:  $\text{odd } n \implies 2 \leq a \implies a < n \implies \text{prime } n \implies P \ n \ a$

**begin**

**lemma** *FalseD*:

**assumes** *false*:  $\text{False} \in \text{set-pmf } (\text{primality-test } P \ n)$

**shows**  $\neg \text{prime } n$

*<proof>*

**theorem** *prime*:

**assumes** *odd-prime*:  $\text{prime } n$

**shows** *primality-test*  $P \ n = \text{return-pmf } \text{True}$

*<proof>*

**end**

We call a primality test  $q$ -good for a fixed positive real number  $q$  if the probability that it mistakenly classifies a composite as a prime is less than  $q$ .

**locale** *good-prob-primality-test* = *prob-primality-test* +

```

fixes  $q :: \text{real}$ 
assumes  $q\text{-pos}: q > 0$ 
assumes  $\text{composite-witness-bound}$ :
     $\neg \text{prime } n \implies 2 < n \implies \text{odd } n \implies$ 
     $\text{real } (\text{card } \{a . 2 \leq a \wedge a < n \wedge P \ n \ a\}) < q * \text{real } (n - 2)$ 
begin

lemma  $\text{composite-aux}$ :
assumes  $\neg \text{prime } n$ 
shows  $\text{measure-pmf.expectation } (\text{primality-test } P \ n) \ \text{of-bool} < q$ 
   $\langle \text{proof} \rangle$ 

theorem  $\text{composite}$ :
assumes  $\neg \text{prime } n$ 
shows  $\text{pmf } (\text{primality-test } P \ n) \ \text{True} < q$ 
   $\langle \text{proof} \rangle$ 

end

end

```

## 10 Fermat's Test

```

theory  $\text{Fermat-Test}$ 
imports
   $\text{Fermat-Witness}$ 
   $\text{Generalized-Primality-Test}$ 
begin

definition  $\text{fermat-test} = \text{primality-test } (\lambda n \ a. \ \text{fermat-liar } a \ n)$ 

The Fermat test is a good probabilistic primality test on non-Carmichael
numbers.

locale  $\text{fermat-test-not-Carmichael-number} =$ 
  fixes  $n :: \text{nat}$ 
  assumes  $\text{not-Carmichael-number}: \neg \text{Carmichael-number } n \vee n < 3$ 
begin

sublocale  $\text{fermat-test}: \text{good-prob-primality-test } \lambda a \ n. \ \text{fermat-liar } n \ a \ n \ 1 / 2$ 
  rewrites  $\text{primality-test } (\lambda a \ n. \ \text{fermat-liar } n \ a) = \text{fermat-test}$ 
   $\langle \text{proof} \rangle$ 

end

lemma  $\text{not-coprime-imp-fermat-witness}$ :
fixes  $n :: \text{nat}$ 
assumes  $n > 1 \ \neg \text{coprime } a \ n$ 
shows  $\text{fermat-witness } a \ n$ 
   $\langle \text{proof} \rangle$ 

```

**theorem** *fermat-test-prime*:  
**assumes** *prime n*  
**shows** *fermat-test n = return-pmf True*  
*<proof>*

**theorem** *fermat-test-composite*:  
**assumes**  $\neg \text{prime } n \ \neg \text{Carmichael-number } n \ \vee \ n < 3$   
**shows** *pmf (fermat-test n) True < 1 / 2*  
*<proof>*

For a Carmichael number  $n$ , Fermat's test as defined above mistakenly returns 'True' with probability  $(\varphi(n) - 1)/(n - 2)$ . This probability is close to 1 if  $n$  has few and big prime factors; it is not quite as bad if it has many and/or small factors, but in that case, simple trial division can also detect compositeness.

Moreover, Fermat's test only succeeds for a Carmichael number if it happens to guess a number that is not coprime to  $n$ . In that case, the fact that we have found a number between 2 and  $n$  that is not coprime to  $n$  alone is proof that  $n$  is composite, and indeed we can even find a non-trivial factor by computing the GCD. This means that for Carmichael numbers, Fermat's test is essentially no better than the very crude method of attempting to guess numbers coprime to  $n$ .

This means that, in general, Fermat's test is not very helpful for Carmichael numbers.

**theorem** *fermat-test-Carmichael-number*:  
**assumes** *Carmichael-number n*  
**shows** *fermat-test n = bernoulli-pmf (real (totient n - 1) / real (n - 2))*  
*<proof>*

**end**

## 11 The Miller–Rabin Test

**theory** *Miller-Rabin-Test*  
**imports**  
*Fermat-Witness*  
*Generalized-Primality-Test*  
**begin**

**definition** *miller-rabin = primality-test* ( $\lambda n \ a.$  *strong-fermat-liar a n*)

The test is actually  $\frac{1}{4}$  good, but we only show  $\frac{1}{2}$ , since the former is much more involved.

**interpretation** *miller-rabin: good-prob-primality-test*  $\lambda n \ a.$  *strong-fermat-liar a n*  
*n 1 / 2*

**rewrites** *primality-test* ( $\lambda n a. \text{strong-fermat-liar } a \ n$ ) = *miller-rabin*  
 $\langle \text{proof} \rangle$

**end**

## 12 The Solovay–Strassen Test

**theory** *Solovay-Strassen-Test*

**imports**

*Generalized-Primality-Test*

*Euler-Witness*

**begin**

**definition** *solovay-strassen-witness* ::  $\text{nat} \Rightarrow \text{nat} \Rightarrow \text{bool}$  **where**

*solovay-strassen-witness*  $n \ a =$   
 $(\text{let } x = \text{Jacobi } (\text{int } a) \ (\text{int } n) \text{ in } x \neq 0 \wedge [x = \text{int } a \wedge ((n - 1) \text{ div } 2)] \ (\text{mod } n))$

**definition** *solovay-strassen* ::  $\text{nat} \Rightarrow \text{bool}$  **pmf** **where**

*solovay-strassen* = *primality-test solovay-strassen-witness*

**lemma** *prime-imp-solovay-strassen-witness*:

**assumes** *prime*  $p$  *odd*  $p$   $a \in \{2..<p\}$

**shows** *solovay-strassen-witness*  $p \ a$

$\langle \text{proof} \rangle$

**lemma** *card-solovay-strassen-liars-composite*:

**fixes**  $n :: \text{nat}$

**assumes**  $\neg \text{prime } n$   $n > 2$  *odd*  $n$

**shows**  $\text{card } \{a \in \{2..<n\}. \text{solovay-strassen-witness } n \ a\} < (n - 2) \text{ div } 2$

(**is**  $\text{card } ?A < -$ )

$\langle \text{proof} \rangle$

**interpretation** *solovay-strassen*: *good-prob-primality-test solovay-strassen-witness*  
 $n \ 1 / 2$

**rewrites** *primality-test solovay-strassen-witness* = *solovay-strassen*

$\langle \text{proof} \rangle$

**end**

## References

- [1] L. W. after a lecture by Prof. Dr. G. Kemper. Skript: Computer algebra, 2015.
- [2] Brilliant.org. Jacobi symbol, 2018. Last accessed 18 November 2018.

- [3] Brilliant.org. Law of quadratic reciprocity, 2018. Last accessed 18 November 2018.
- [4] Brilliant.org. Legendre symbol, 2018. Last accessed 18 November 2018.
- [5] Brilliant.org. Quadratic residues, 2018. Last accessed 18 November 2018.
- [6] R. D. Carmichael. Note on a new number theory function. *Bulletin of the American Mathematical Society*, 16(5):232–238, 1910.
- [7] K. Conrad. The miller–rabin test, 2018. Last accessed 18 November 2018.
- [8] K. Conrad. The solovay–strassen test, 2018. Last accessed 18 November 2018.
- [9] G. Jameson. Carmichael numbers and pseudoprimes, 2010. Last accessed 18 November 2018.
- [10] B. Kleinberg. The miller-rabin randomized primality test, 2010. Last accessed 18 November 2018.
- [11] mathwizard. Solovay–strassen test, 2013. Last accessed 18 November 2018.
- [12] A. Nicolas. Carmichael number square free, 2016. Last accessed 18 November 2018.
- [13] R. Solovay and V. Strassen. A fast monte-carlo test for primality. *SIAM journal on Computing*, 6(1):84–85, 1977.
- [14] wikipedia. Carmichael number, 2013. Last accessed 18 November 2018.
- [15] wikipedia. Solovay–strassen primality test, 2013. Last accessed 18 November 2018.