Technical Report

UCAM-CL-TR-502
ISSN 1476-2986

Number 502

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Formalizing basic number theory

Thomas Marthedal Rasmussen

September 2000

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

ISSN 1476-2986

Formalizing Basic Number Theory

Thomas Marthedal Rasmussen[*]
Computer Laboratory
University of Cambridge

September 2000

Abstract

This document describes a formalization of basic number theory including two theorems of Fermat and Wilson.

Most of this have (in some context) been formalized before but we present a new generalized approach for handling some central parts, based on concepts which seem closer to the original mathematical intuition and likely to be useful in other (similar) developments.

Our formalization has been mechanized in the Isabelle/HOL system.

Contents

[*] Permanent address: Dept. of Information Technology, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark. E-mail: `tmr@it.dtu.dk`

1 Introduction

Based on a theory of the integers up to and including the operators mod and div we formalize basic parts of Number Theory.

We formalize the notions of divides, greatest common divisor and congruence on which we then build to formalize the Chinese Remainder Theorem and two theorems of Fermat and Wilson.

Most of these results have been formalized and mechanized before (most notably using the Boyer-Moore Theorem Prover [2, 3, 4, 8]). Part of our contribution is to give a coherent presentation of our development, which has been mechanized using the Isabelle/HOL theorem proving system [7].

More interesting though, is the way we have formalized important parts of the theorems of Fermat and Wilson. Both use notions of "pairing off" elements of sets in a "one-to-one" manner. We have developed a generalized approach to handle these concepts. Once the machinery is in place, this seems to be more intuitive and closer to the original mathematical proofs. We also believe that this approach can be used when formalizing similar concepts in other contexts.

In number theory it is customary to mix natural numbers and integers freely; thus one uses whatever is most appropriate in a given situation. In a thorough formalization this is difficult to do — one has to choose either. The formalizations and mechanizations of [2, 3, 4, 8, 9, 10] are all restricted to the natural numbers. In [6] the formalization and mechanization is based on the integers but only the Prime Factorization Theorem is proved. This has also been done in Boyer-Moore and Isabelle/HOL. To the author's knowledge the work presented in this document is the most comprehensive development based on the integers.

Using the natural numbers makes some parts of the reasoning simpler but also puts some limitations on the expressiveness. In formalizations and mechanizations based on integers, if needed, natural numbers are represented as non-negative integers.

This document is organized as follows: Section 2 presents those parts of number theory we have formalized and mechanized. We give a fairly thorough mathematical presentation. In Section 3 we turn to the question of formalizing these concepts. Some parts do not need much work in the sense that the mathematical development is formal enough to act as a basis for mechanization, whereas other parts need some substantial additional theory. Finally, in Section 4 we discuss how the formalization is mechanized in Isabelle/HOL. The presentation of the formalization is fairly detailed such that most of the mechanization is essentially straightforward based on that.

2 Basic Number Theory

This section contains material which can be found in all introductory texts on number theory, e.g. [1, 5]. On the other hand, we have made the development here more elaborate and some proofs more detailed than what can be found in those books. Importantly, we have tried to keep this in the spirit of a mathematician being asked to make the development more rigorous, i.e. without any specific thoughts of mechanization. We will later see that this further rigour is useful in mechanization but there are still some gaps around due to their seemingly obviousness to a mathematician.

Unless otherwise stated we work solely with integers in the following.

Definition 2.1 *We say that*

1. a divides b *(written $a \mid b$) if there is a k such that $b = ak$*

2. a *is* congruent *to* b modulo m *(written $a \equiv b \pmod{m}$) if $m \mid (a - b)$*

3. *the* greatest common divisor *of a and b is c (written $\gcd(a, b) = c$) if $c \mid a$ and $c \mid b$ and for all d, if $d \mid a$ and $d \mid b$ then $d \mid c$*

It is easy to show that $\mid$ is reflexive and transitive, and that $\equiv$ is an equivalence relation.

There exists an algorithm (known as *Euclid's Algorithm*) for computing the greatest common divisor of two numbers. This algorithm can be extended to compute two numbers f, g such that

If $\gcd(a, b) = c$ then $c = af + bg$

Below we state some properties (without proof) of the modulo operator. Notice that we always assume m to be positive in any expression involving "mod m" (including congruence expressions).

Proposition 2.2

1. $(ab) \bmod m = (a(b \bmod m)) \bmod m$

2. $(a + b) \bmod m = (a + (b \bmod m)) \bmod m$

3. $a \equiv b \pmod{m}$ *iff* $a \equiv (b \bmod m) \pmod{m}$

The following proposition lists properties of congruence and greatest common divisor.

Proposition 2.3

1. *If* $\gcd(a, m) = 1$ *and* $\gcd(b, m) = 1$ *then* $\gcd(ab, m) = 1$

2. *If* $a \equiv b \pmod{m}$ *and* $c \equiv d \pmod{m}$ *then* $ac \equiv bd \pmod{m}$

3. *If* $\gcd(k, m) = 1$ *and* $ka \equiv kb \pmod{m}$ *then* $a \equiv b \pmod{m}$

4. *If* $\gcd(m, n) = 1$ *and* $a \equiv b \pmod{m}$ *and* $a \equiv b \pmod{n}$ *then* $a \equiv b \pmod{mn}$

5. *If* $\gcd(a, m) = 1$ *and* $a \equiv b \pmod{m}$ *then* $\gcd(b, m) = 1$

6. *For any a there exists a unique b such that* $0 \leq b < m$ *and* $a \equiv b \pmod{m}$

7. *If* $\gcd(a,m) = 1$ *then for any* b *there exists a unique* x *such that* $0 \leq x < m$ *and* $ax \equiv b \pmod{m}$

We will not give proofs for these properties except for 7. for which we later give two different proofs to illustrate some important points.

Euler's *totient function* $\phi(m)$ is defined as follows:

Definition 2.4 (Totient)

$$\phi(m) \mathrel{\widehat{=}} \mid \{n \mid \gcd(n,m) = 1 \wedge 0 \leq n < m\} \mid$$

In other words, $\phi(m)$ is the number of non-negative integers less than and relatively prime to m. Note, $\phi(p) = p - 1$ when p is prime.

Definition 2.5 (Residue Sets) *A* complete set of residues $\pmod{m}$ *is any set of* m *numbers mutually non-congruent* $\pmod{m}$*.*

A reduced set of residues $\pmod{m}$ *is any set of* $\phi(m)$ *numbers mutually non-congruent* $\pmod{m}$ *and relatively prime to* m*.*

We note that complete sets of residues are "maximal" in the sense that *any* number will be congruent $\pmod{m}$ to some number in every complete set of residues. This stems from the fact that when doing arithmetic $\pmod{m}$ there are essentially only m distinct numbers.

Below we give the two most important examples of a complete residue set and a reduced residue set, respectively:

- $\Upsilon_m \mathrel{\widehat{=}} \{n \mid 0 \leq n < m\}$
- $\Phi_m \mathrel{\widehat{=}} \{n \mid \gcd(n,m) = 1 \wedge 0 \leq n < m\}$

Note by definition, $\phi(m) = |\Phi_m|$.

If x is some integer and A is some set of integers then $xA \mathrel{\widehat{=}} \{xa \mid a \in A\}$.

Lemma 2.6 *Let* m *be positive and let* x *be some integer with* $\gcd(x,m) = 1$*. Let* A *be a complete set of residues* $\pmod{m}$ *and let* B *be a reduced set of residues* $\pmod{m}$*. Then*

- xA *is a complete set of residues*
- xB *is a reduced set of residues*

Proof Let $a_1, a_2 \in A$ and assume $xa_1 \equiv xa_2 \pmod{m}$. By Proposition 2.3-3 we get $a_1 \equiv a_2 \pmod{m}$ and therefore $a_1 = a_2$ as all elements in A are mutually non-congruent. Thus we conclude that all elements in xA are mutually non-congruent too. Clearly $|A| = |xA|$ as $x \neq 0$ (which follows from the assumptions), hence xA is a complete set of residues.

To show that xB is a reduced set of residues we show as above that all elements are mutually non-congruent and that $|B| = |xB|$. We are thus done if we can show $\gcd(xb, m) = 1$ for all $b \in B$ but this follows from Proposition 2.3-1. □

Proposition 2.7 *Let A and B be two complete sets of residues* (mod m)*. Then the elements of A and B can be put in a unique one-to-one correspondence with respect to congruence* (mod m)*.*

Proof Let $a \in A$. *Existence*: It follows immediately from maximality of B that there is a $b \in B$ such that $a \equiv b \pmod{m}$.

Uniqueness: Assume $a \equiv b_1 \pmod{m}$ and $a \equiv b_2 \pmod{m}$ where $b_1, b_2 \in B$. By symmetry and transitivity of $\equiv$ we get $b_1 \equiv b_2 \pmod{m}$. But this means $b_1 = b_2$ because of the mutual non-congruence of elements in B. We can similarly show that no two distinct elements of A are congruent to the same element of B. □

Proposition 2.8 *Let A and B be two reduced sets of residues* (mod m)*. Then the elements of A and B can be put in a unique one-to-one correspondence with respect to congruence* (mod m)*.*

Proof We show that an arbitrary reduced set of residues (mod m) (let A be such a set and let $a \in A$) can be put in a unique one-to-one correspondence with Φ_m. Then the proposition follows from symmetry and transitivity of one-to-one correspondence.

Existence: By Proposition 2.3-6 there is a b such that $0 \leq b < m$ and $a \equiv b \pmod{m}$. If $\gcd(b, m) = 1$ then $b \in \Phi_m$ by definition. But we know $\gcd(a, m) = 1$ and the result then follows from Proposition 2.3-5.

Uniqueness: Similar to the uniqueness part of the proof of Proposition 2.7. □

To illustrate the use of Proposition 2.7 we give below a proof (A) of Proposition 2.3-7 using complete residue sets. We also give a quite different proof (B) using the extended gcd algorithm.

Proof A of Proposition 2.3-7 Let X be a complete set of residues (mod m). By maximality of X we have that $b \equiv x \pmod{m}$ for some $x \in X$. By Lemma 2.6 we know that aX is a complete set of residues too. Using Proposition 2.7 we conclude that there is $x' \in X$ such that $ax' \equiv x \pmod{m}$, hence $ax' \equiv b \pmod{m}$. □

Proof B of Proposition 2.3-7 We first show that $ax \equiv 1 \pmod{m}$ has a solution. Using the extended gcd algorithm we can find f, g such that $af + mg = 1$ from which easily follows $af \equiv 1 \pmod{m}$, thus f is a solution. We now show that fb is a solution to $ax \equiv b \pmod{m}$:

$$a(fb) \equiv b \pmod{m} \quad \text{if} \quad af \equiv 1 \pmod{m} \text{ and } b \equiv b \pmod{m}$$

by Proposition 2.3-2. □

In both proofs we have actually not made sure that the solution x we exhibits satisfies $0 \leq x < m$. But $x \bmod m$ satisfies this and is still a solution which can easily be shown using Proposition 2.2. In this case it is also straightforward to show uniqueness.

2.1 The Chinese Remainder Theorem

Theorem 2.9 (Chinese Remainder) *Let $m_0, m_1, \ldots, m_n$ be positive integers with $\gcd(m_i, m_j) = 1$, $i \neq j$. Let $k_0, k_1, \ldots, k_n$ be integers where $\gcd(k_i, m_i) = 1$. Finally, let $b_0, b_1, \ldots, b_n$ be integers.*

Then the system

$$k_i x \equiv b_i \pmod{m_i}$$

has a unique solution x with $0 \leq x < m_0 m_1 \cdots m_n$

Proof Let $m \mathrel{\widehat{=}} m_0 m_1 \cdots m_n$. *Uniqueness*: Assume $k_i x \equiv b_i \pmod{m_i}$ and $k_i y \equiv b_i \pmod{m_i}$, $0 \leq i \leq n$. By transitivity and symmetry of $\equiv$ we have $k_i x \equiv k_i y \pmod{m_i}$ and then Proposition 2.3-3 gives

$$x \equiv y \pmod{m_i}$$

for $0 \leq i \leq n$. By Proposition 2.3-4 we now have $x \equiv y \pmod{m}$ from which we conclude $x = y$ as both $x < m$ and $y < m$.

Existence: Let $\widehat{m_i} \mathrel{\widehat{=}} m \text{ div } m_i$. By Proposition 2.3-7 there is a unique solution x_i to

$$k_i \widehat{m_i} x_i \equiv b_i \pmod{m_i} \tag{1}$$

as $\gcd(k_i \widehat{m_i}, m_i) = 1$ (Proposition 2.3-1). We now show that

$$x \mathrel{\widehat{=}} \left(\sum_{i=0}^{n} \widehat{m_i} x_i \right) \bmod m$$

is the required solution to $k_i x \equiv b_i \pmod{m_i}$. First, as $m_i | \widehat{m_j}$ when $i \neq j$ we get from Proposition 2.2-2 that

$$\left(\sum_{i=0}^{n} \widehat{m_i} x_i \right) \bmod m_i = \widehat{m_i} x_i \bmod m_i \tag{2}$$

Now, by Proposition 2.2-1 and Proposition 2.2-3

$$k_i x \equiv b_i \pmod{m_i} \quad \text{iff} \quad (k_i \bmod m_i)(x \bmod m_i) \equiv (b_i \bmod m_i) \pmod{m_i}$$

As $m_i | m$ it follows that

$$(k_i \bmod m_i)(\left(\sum_{i=0}^{n} \widehat{m_i} x_i \right) \bmod m_i) \equiv (b_i \bmod m_i) \pmod{m_i}$$

From (2) we finally arrive at

$$k_i \widehat{m_i} x_i \equiv b_i \pmod{m_i}$$

which is (1). □

2.2 Fermat's Little Theorem

Theorem 2.10 (Fermat) *Let p be prime and let x be some integer. Assume p does not divide x. Then*

$$x^{p-1} \equiv 1 \pmod{p}$$

We give two proofs of this theorem. The first proof (A) uses the binomial expansion; the second (B) uses complete residue sets.

Proof A of Theorem 2.10 This proof only works for x positive. By Proposition 2.3-3 we are done if we can show $x^p \equiv x \pmod{p}$. We do this by induction on x. The base case ($x = 1$) is trivial. The induction hypothesis is $(x-1)^p \equiv x - 1 \pmod{p}$ which is equivalent to $(x-1)^p + 1 \equiv x \pmod{p}$. By transitivity of $\equiv$ we are done if we can show $x^p \equiv (x-1)^p + 1 \pmod{p}$. Consider x^p: By using the binomial expansion we get

$$\begin{array}{rcl} x^p = ((x-1)+1)^p & = & \sum_{k=0}^{p} \binom{p}{k}(x-1)^{p-k}1^k \\ & = & (x-1)^p + \left(\sum_{k=1}^{p-1} \binom{p}{k}(x-1)^{p-k}\right) + 1 \end{array}$$

We have that $p|\binom{p}{k}$ when $0 < k < p$ hence $x^p \equiv (x-1)^p + 1 \pmod{p}$. □

Proof B of Theorem 2.10 This proof works for any integer x. By Proposition 2.7 we know that the elements of Υ_p and $x\Upsilon_p$ (that $x\Upsilon_p$ is a complete set of residues follows from Lemma 2.6) can be paired of uniquely with respect to congruence (mod p). As we clearly have $0 \equiv x0 \pmod{p}$ also the sets $\Upsilon_p \setminus \{0\}$ and $x(\Upsilon_p \setminus \{0\})$ can be paired of in this manner. From Proposition 2.3-2 follows that their products are congruent:

$$\prod(\Upsilon_p \setminus \{0\}) \equiv \prod x(\Upsilon_p \setminus \{0\}) \pmod{p}$$

hence

$$\prod(\Upsilon_p \setminus \{0\}) \equiv x^{p-1} \prod(\Upsilon_p \setminus \{0\}) \pmod{p}$$

The Theorem now follows using Proposition 2.3-2 because $\gcd(\prod(\Upsilon_p \setminus \{0\}), p) = 1$ by Proposition 2.3-1. □

The following theorem generalizes Theorem 2.10 to arbitrary moduli, i.e. not only primes.

Theorem 2.11 (Euler-Fermat) *Let m be positive with* $\gcd(x, m) = 1$ *for some integer x. Then*

$$x^{\phi(m)} \equiv 1 \pmod{m}$$

Proof Let Y be a reduced set of residues (mod m). By Lemma 2.6 it follows that xY is a reduced set of residues (mod m) too. Using Proposition 2.8 we have that Y and xY can be paired of uniquely with respect to congruence (mod m). Thus (Proposition 2.3-2),

$$\prod Y \equiv \prod xY \pmod{m}$$

hence

$$\prod Y \equiv x^{\phi(m)} \prod Y \pmod{m}$$

The Theorem now follows using Proposition 2.3-2 because $\gcd(\prod Y, m) = 1$ by Proposition 2.3-1. □

Note that the proof is a generalization of proof B of Theorem 2.10. Proof A cannot be generalized in a similar way [5].

2.3 Wilson's Theorem

Lemma 2.12 *Let p be prime and assume $0 \leq a < p$. Then*

$$a^2 \equiv 1 \pmod{p} \quad \textit{iff} \quad a = 1 \textit{ or } a = p - 1$$

Proof *If*: Trivially $1 \equiv 1 \pmod{p}$. As $(p-1)(p-1) = p(p-2)+1$ we also have $(p-1)(p-1) \equiv 1 \pmod{p}$.

Only if: By definition $a^2 \equiv 1 \pmod{p}$ iff $p \mid a^2 - 1$. But as p is prime we have $p \mid (a-1)$ or $p \mid (a+1)$ which is only possible if $a = 1$ or $a = p-1$. □

Theorem 2.13 (Wilson) *Let p be prime. Then*

$$(p-1)! \equiv -1 \pmod{p}$$

Proof Assume $0 < a < p$. Then there exists unique a' such that $0 < a' < p$ $aa' \equiv 1 \pmod{p}$ (Proposition 2.3-7). Clearly, $a' \neq 0$. If $a = a'$ we know by Lemma 2.12 that $a = 1$ or $a = p-1$. This means that the set $\Theta_p = \{n \mid 1 < n < p-1\}$ can be divided into $\frac{p-3}{2}$ pairs (a, a') with $aa' \equiv 1 \pmod{p}$. By Proposition 2.3-2 we thus get $\prod \Theta_p \equiv 1 \pmod{p}$, hence $(p-2)! \equiv 1 \pmod{p}$. As $p - 1 \equiv -1 \pmod{p}$ we finally have $(p-1)! \equiv -1 \pmod{p}$. This proof assumes $p \geq 5$. Clearly the Theorem holds for $p = 2$ and $p = 3$. □

3 Formalization

In this section we revisit the development of the previous section and fill the gaps necessary for a rigorous formalization. This is done with an eye to an eventual mechanization.

The presentation is fairly elaborate and the proofs all quite detailed. This presentation has been chosen as it can then be seen as "documenting" the mechanization, thus the steps of the mechanization follows from the detailed formalization in a straightforward way.

The major gaps are the notions and reasoning of one-to-one correspondences concerning residue sets and the "pairing off" in Wilson's theorem. This can be handled in a general framework with the notion of *bijection relations* as defined in the following section.

3.1 Bijection Relations

When referring to sets in the following we always mean *finite sets of integers* unless otherwise stated.

Definition 3.1 *Let $P \subseteq \mathbb{Z} \times \mathbb{Z}$. The relation $\sim_P \subseteq \mathcal{P}(\mathbb{Z}) \times \mathcal{P}(\mathbb{Z})$ is inductively defined as follows*

$$\frac{}{\emptyset \sim_P \emptyset} \qquad \frac{P(a,b) \quad a \notin A \quad b \notin B \quad A \sim_P B}{(\{a\} \cup A) \sim_P (\{b\} \cup B)}$$

It is straightforward to show by induction that $\sim_P$ is symmetric and transitive if P is.

Proposition 3.2 *Let f be an injective function with domain A. Assume $P(a, f(a))$ for all $a \in A$. Then $A \sim_P f(A)$.*

Proof By induction on the size of A. Assume $A = \emptyset$. Then $\emptyset \sim_P f(\emptyset)$ because $\emptyset \sim_P \emptyset$. Now, assume $A = \{a\} \cup A'$ where $a \notin A'$. We must show $(\{a\} \cup A') \sim_P f(\{a\} \cup A')$ But as $f(\{a\} \cup A') = f(a) \cup f(A')$ we are done if (by definition of $\sim_P$): $P(a, f(a))$, $a \notin A'$, $f(a) \notin f(A')$ and $A' \sim_P f(A')$. The first two requirements follow by assumption, the third because f is injective and the last follows by induction. □

Definition 3.3 *Let $P \subseteq \mathbb{Z} \times \mathbb{Z}$. The set $\mathrm{BS}_P \subseteq \mathcal{P}(\mathbb{Z})$ is inductively defined as follows*

$$\frac{}{\emptyset \in \mathrm{BS}_P} \qquad \frac{P(a,a') \quad a \notin A \quad a' \notin A \quad A \in \mathrm{BS}_P}{(\{a,a'\} \cup A) \in \mathrm{BS}_P}.$$

Proposition 3.4

If $A \sim_P A$ then $A \in \mathrm{BS}_P$

Proof Assume $A \sim_P A$. We now show $A \in \mathrm{BS}_P$ by induction on the definition of $\sim_P$. Assume $A = \emptyset$. Then trivially $\emptyset \in \mathrm{BS}_P$. Now, assume $A = \{a'\} \cup A'$ and $A = \{a''\} \cup A''$ where $P(a', a'')$, $a' \notin A'$, $a'' \notin A''$ and $A' \sim_P A''$. Assume $a' = a''$. Then $A' = A''$ and by induction $A' \in \mathrm{BS}_P$ hence $(\{a'\} \cup A') \in \mathrm{BS}_P$. Now, assume $a' \neq a''$. Then there is $\overline{A}$ such that $A' = \{a''\} \cup \overline{A}$, $A'' = \{a'\} \cup \overline{A}$, $a' \notin \overline{A}$, $a'' \notin \overline{A}$. As $A' \sim_P A''$ we furthermore have $\overline{A} \sim_P \overline{A}$ and then by induction $\overline{A} \in \mathrm{BS}_P$ from which we finally conclude $(\{a', a''\} \cup \overline{A}) \in \mathrm{BS}_P$ □

3.2 Fermat's Little Theorem

We concentrate here on the formalization of the generalized version of Fermat's Little Theorem (Theorem 2.10).

We first formalize the notion of a reduced set of residues (mod m) (cf. Definition 2.5).

Definition 3.5 *The set* $\mathrm{RR}_m \subseteq \mathcal{P}(\mathbb{Z})$ *is inductively defined as follows*

$$\frac{}{\emptyset \in \mathrm{RR}_m} \qquad \frac{\gcd(a,m)=1 \quad \forall a' \in A.\ a \not\equiv a' \pmod{m} \quad A \in \mathrm{RR}_m}{(\{a\} \cup A) \in \mathrm{RR}_m}$$

A set A is thus a reduced set of residues (mod m) *if* $A \in \mathrm{RR}_m$ *and* $|A| = \phi(m)$.

Given a recursive definition of Φ_m it is easy to show that Φ_m is a reduced residue set (mod m) according to the above definition.

We now turn to a formalization of the notion of one-to-one correspondence between reduced sets of residues.

Lemma 3.6 *If* $\gcd(a,m) = 1$ *then there exists a unique b such that* $a \equiv b \pmod{m}$ *and* $b \in \Phi_m$. *Let $\epsilon_m(a)$ denote this b when it exists.*

Proof Straightforward by Proposition 2.3-6 and Proposition 2.3-5 remembering the definition of Φ_m □

Notice that, $\epsilon_m(a)$ is essentially just a mod m.

Proposition 3.7 *Let A be a reduced set of residues* (mod m). *Then ϵ_m is injective on A and*

$$\epsilon_m(A) = \Phi_m$$

Proof First, notice that ϵ_m is well-defined on A (because for all $a \in A$, $\gcd(a,m) = 1$).

Assume $a_1, a_2 \in A$ and $\epsilon_m(a_1) = \epsilon_m(a_2)$. By definition we have $a_1 \equiv \epsilon_m(a_1) \pmod{m}$ and $a_2 \equiv \epsilon_m(a_2) \pmod{m}$, thus $a_1 \equiv a_2 \pmod{m}$. Hence, as all elements of A are mutually non-congruent (mod m) we get $a_1 = a_2$.

As ϵ_m is injective on A and $|A| = |\Phi_m|$ we have $\epsilon_m(A) = \Phi_m$ if $\epsilon_m(A) \subseteq \Phi_m$. But this is clearly the case by definition of ϵ_m. □

We are interested in the relation $\sim_{C_m}$ where

$$C_m(a,b) \quad \widehat{=} \quad a \equiv b \pmod{m}$$

Proposition 3.8 *Let A and B be reduced sets of residues* (mod m). *Then*

$$A \sim_{C_m} B$$

Proof Clearly, by definition $C_m(a, \epsilon_m(a))$ for all $a \in A$. Thus, $A \sim_{C_m} \epsilon_m(A)$ by Proposition 3.2 and Proposition 3.7. Using the second part of Proposition 3.7 we get $A \sim_{C_m} \Phi_m$. We can similarly show $B \sim_{C_m} \Phi_m$. Finally, by symmetry and transitivity of $\sim_{C_m}$ we get $A \sim_{C_m} B$ □

Proposition 3.9

$$\textit{If} \quad A \sim_{C_m} B \quad \textit{then} \quad \prod A \equiv \prod B \pmod{m}$$

Proof Assume $A \sim_{C_m} B$. We now show $\prod A \equiv \prod B \pmod{m}$ by induction on the definition of $\sim_{C_m}$. Assume $A = B = \emptyset$. Then trivially $\prod \emptyset \equiv \prod \emptyset \pmod{m}$. Now assume $A = \{a\} \cup A'$ and $B = \{b\} \cup B'$ where $a \equiv b \pmod{m}$, $a \notin A$, $b \notin B$ and $A' \sim_{C_m} B'$. By induction $\prod A' \equiv \prod B' \pmod{m}$. Finally (using Proposition 2.3-2),

$$a \prod A' \equiv b \prod B' \pmod{m} = \prod A \equiv \prod B \pmod{m}$$

□

Proposition 3.10 *Assume* $\gcd(x, m) = 1$. *Then*

1. *If* $A \in \mathrm{RR}_m$ *then* $xA \in \mathrm{RR}_m$
2. $|A| = |xA|$
3. *If A is a reduced set of residues* $\pmod{m}$ *then xA is a reduced set of residues* $\pmod{m}$

Proof *1.* By induction on the definition of RR_m. Assume $A = \emptyset \in \mathrm{RR}_m$. Then clearly $x\emptyset = \emptyset \in \mathrm{RR}_m$. Now, assume $A \in \mathrm{RR}_m$ because $A = \{a\} \cup A'$, $\gcd(a, m) = 1$, $\forall a' \in A'.\ a \not\equiv a' \pmod{m}$ and $A' \in \mathrm{RR}_m$. By induction, $xA' \in \mathrm{RR}_m$. We have $\gcd(ax, m) = 1$ by Proposition 2.3-1, and $\forall a' \in A'.\ xa \not\equiv xa' \pmod{m}$ by Proposition 2.3-3. We therefore conclude $\{xa\} \cup xA' = x(\{a\} \cup A') \in \mathrm{RR}_m$.

2. Follows immediately as multiplication by a constant x is an injective function if $x \neq 0$ (which is the case by the assumption).

3. Follows from *2.* and *3.* (cf. Definition 3.5) □

By induction on the size of a set A we can show:

$$\prod xA = x^{|A|} \prod A$$

Revisiting the proof of Theorem 2.11 we see that the above equality together with Propositions 3.8, 3.9 and 3.10 formalizes the essential parts of that proof.

Note that the formalization of this section can be used (with minor changes) to prove both proof A of Proposition 2.3-7 and proof B of Theorem 2.10.

3.2.1 Boyer–Moore's proof

In this subsection we sketch a proof of Fermat's Little Theorem as formalized and mechanized by Boyer and Moore in [3, 4].

Let $\Psi_p = \{x\gamma \bmod p \mid \gamma \in (\Upsilon_p \setminus \{0\})\}$ Clearly, $0 \leq x\gamma \bmod p < p$ for all $\gamma \in (\Upsilon_p \setminus \{0\})$. Now $x\gamma \bmod p = 0$ iff $p \mid x\gamma$ iff $p \mid x$ or $p \mid \gamma$, hence $0 < x\gamma \bmod p < p$. Assume $x\gamma_1 \bmod p = x\gamma_2 \bmod p$. This is the case iff $x\gamma_1 \equiv x\gamma_2 \pmod{p}$ iff $\gamma_1 \equiv \gamma_2 \pmod{p}$ iff $\gamma_1 = \gamma_2$. In conclusion $\Psi_p = \Upsilon_p \setminus \{0\}$

Clearly, $\prod \Psi_p \equiv \prod(\Upsilon_p \setminus \{0\}) \pmod{p}$. By Proposition 2.2 this is equivalent to $\prod x(\Upsilon_p \setminus \{0\}) \equiv \prod(\Upsilon_p \setminus \{0\}) \pmod{p}$ and the rest of the proof is now similar to the last part of proof B of Theorem 2.10.

If we compare this proof with proof B of Theorem 2.10 the essential difference is the use of the set Ψ_p which is constructed in a way such that it is easy to establish a one-to-one correspondence as this is now simply equality of elements.

This trick makes the formalization simpler but less faithful to the original proof and less general.

3.3 Wilson's Theorem

Lemma 3.11 *Let p be prime and let $1 < a < p - 1$. Then*

$$1 < a_p^{-1} < p - 1 \quad \textit{and} \quad a_p^{-1} \neq a$$

where a_p^{-1} is the unique solution to $ax \equiv 1 \pmod{p}$ (Proposition 2.3-7).

Proof By Proposition 2.3-7 we know $0 \leq a_p^{-1} < p$. Clearly, $a_p^{-1} \neq 0$. If $a_p^{-1} = 1$ then $a \equiv 1 \pmod{p}$ which is not possible for $1 < a < p - 1$. Now, if $a_p^{-1} = p - 1$ then $a(p-1) \equiv 1 \pmod{p}$ iff $a \equiv p - 1 \pmod{p}$ which again is impossible. Finally, if $a_p^{-1} = a$ we arrive at a contradiction using Lemma 2.12. □

We start by revisiting the proof of Wilson's Theorem (Theorem 2.13). The gap in this proof is the "pairing off" of elements in the set Θ_p. We thus need to formalize this notion so as to prove $\prod \Theta_p \equiv 1 \pmod{p}$ in a more rigorous way.

We present two different ways of doing this. The first one (the "concrete" approach) is based on the work of Russinoff in [8]. The second one (the "abstract" approach) uses the notion of bijection relations.

3.3.1 The "Concrete" Approach

We give a concrete definition of a_p^{-1} as follows:

$$a_p^{-1} \hat{=} a^{p-2} \bmod p$$

Clearly, $0 \leq a_p^{-1} < p$. Together with part 1. of Lemma 3.12 below we see that this definition is correct.

Lemma 3.12 *Let $p \geq 5$ be prime. Assume $0 < a < p$.*

1. $aa_p^{-1} \equiv 1 \pmod{p}$
2. $(a_p^{-1})_p^{-1} = a$

Proof Using Proposition 2.2 we get

$$aa_p^{-1} \equiv a(a^{p-2} \bmod p) \equiv a^{p-1} \equiv 1 \pmod{p}$$

which is true by Theorem 2.10.

Extending Proposition 2.2 to exponentiation we get

$$\begin{array}{rcl} (a_p^{-1})_p^{-1} & = & (a^{p-2} \bmod p)^{p-2} \bmod p = a^{(p-2)(p-2)} \bmod p \\ & = & a(a^{p-1})^{p-3} \bmod p = a \quad \text{iff} \quad a(a^{p-1})^{p-3} \equiv a \pmod{p} \\ & \text{iff} & (a^{p-1})^{p-3} \equiv 1 \pmod{p} \quad \text{if} \quad a^{p-1} \equiv 1 \pmod{p} \end{array}$$

which again is true by Theorem 2.10. Note that we have to assume $p \geq 5$. □

Definition 3.13

$$\begin{array}{rcl} \omega(p,a) & = & \textit{if } a > 1 \textit{ then let } w = \omega(p, a-1) \textit{ in} \\ & & \quad \textit{if } a \in w \textit{ then } w \textit{ else } \{a\} \cup \{a_p^{-1}\} \cup w \\ & & \textit{else } \emptyset \end{array}$$

We are interested in the set $\omega(p, p-2)$. By induction on the definition of $\omega(p,a)$ (using Lemma 3.11) we can show

$$b \in \omega(p, p-2) \quad \text{iff} \quad 1 < b < p-1$$

An immediate consequence is that $\Theta_p = \omega(p, p-2)$.

Proposition 3.14 *Let $p \geq 5$ be prime. Assume $1 < a < p-1$ and $1 < b < p-1$. Then*

$$b \in \omega(p,a) \quad \textit{iff} \quad b_p^{-1} \in \omega(p,a)$$

Proof *Only if*: By induction on the definition of $\omega(p,a)$. The base case ($\omega(p,a) = \emptyset$) is trivial. Assume $b \in \omega(p, a-1)$ implies $b_p^{-1} \in \omega(p, a-1)$. We now have to show that $b \in \omega(p,a)$ implies $b_p^{-1} \in \omega(p,a)$. If $b \in \omega(p, a-1)$ we are done so assume $b \notin \omega(p, a-1)$. This means $b = a$ or $b = a_p^{-1}$. In the first case we are done immediately, in the second case we are done by part 2. of Lemma 3.12. *If*: We must show that $b \in \omega(p,a)$ if $b_p^{-1} \in \omega(p, a-1)$. Using again part 2. of Lemma 3.12 this reduces to the *only if* case. □

Proposition 3.15 *Let $p \geq 5$ be prime. Assume $1 < a < p-1$. Then*

$$\prod \omega(p,a) \equiv 1 \pmod{p}$$

Proof By induction on the definition of $\omega(p, a)$. The base case ($\omega(p, a) = \emptyset$) is true by the convention $\prod \emptyset = 1$. Assume $\prod \omega(p, a-1) \equiv 1 \pmod{p}$. We must show $\prod \omega(p, a) \equiv 1 \pmod{p}$. If $a \in \omega(p, a-1)$ we are done so assume $a \notin \omega(p, a-1)$. By definition we now get $\prod \{a\} \cup \{a_p^{-1}\} \cup \omega(p, a-1) \equiv 1 \pmod{p}$ which again is equivalent to $aa_p^{-1} \prod \omega(p, a-1) \equiv 1 \pmod{p}$ if $a \notin (\{a_p^{-1}\} \cup \omega(p, a-1))$ and $a_p^{-1} \notin \omega(p, a-1)$. Assuming the last two conditions we are done by Proposition 2.3-2 and definition of a_p^{-1}. That $a \notin (\{a_p^{-1}\} \cup \omega(p, a-1))$ follows by assumption and Lemma 3.11. Finally, $a_p^{-1} \notin \omega(p, a-1)$ follows from Proposition 3.14. □

This proposition thus gives $\prod \omega(p, p-2) \equiv 1 \pmod{p}$ hence $\prod \Theta_p \equiv 1 \pmod{p}$

3.3.2 The "Abstract" Approach

In this section we are interested in the following relation

$$R_p(a, b) \mathrel{\widehat{=}} (ab \equiv 1 \pmod{p} \wedge 1 < a < p-1 \wedge 1 < b < p-1)$$

Proposition 3.16 *Let p be prime.*

$$\textit{If} \quad A \ \in \ \mathrm{BS}_{R_p} \quad \textit{then} \quad \prod A \equiv 1 \pmod{p}$$

Proof Assume $A \ \in \ \mathrm{BS}_{R_p}$. We now show $\prod A \equiv 1 \pmod{p}$ by induction on the definition of BS_{R_p}. Assume $A = \emptyset$. Then $\prod \emptyset \equiv 1 \pmod{m}$ because of the convention $\prod \emptyset = 1$.

Now, assume $A = \{a, a'\} \cup A'$ because $aa' \equiv 1 \pmod{p}$, $1 < a < p-1$, $1 < a' < p-1$, $a \notin A'$, $a' \notin A'$ and $A' \ \in \ \mathrm{BS}_{R_p}$. If $a = a'$ we arrive at a contradiction according to Lemma 2.12. Thus, assume $a \neq a'$. By induction $\prod A' \equiv 1 \pmod{p}$. Now (using Proposition 2.3-2),

$$aa' \prod A' \equiv 1 \pmod{p} = \prod A \equiv 1 \pmod{p}$$

□

Proposition 3.17 *The function $(-)_p^{-1}$ is injective on the domain Θ_p. Furthermore*

$$(\Theta_p)_p^{-1} = \Theta_p$$

Proof Assume $x, y \in \Theta_p$ and $x_p^{-1} = y_p^{-1}$. We must show $x = y$. By definition, $xx_p^{-1} \equiv 1 \pmod{p}$ and $yy_p^{-1} \equiv 1 \pmod{p}$. By Proposition 2.3-3 together with Lemma 3.11 we get $x \equiv y \pmod{p}$ and therefore $x = y$. As $(-)_p^{-1}$ is injective on Θ_p we have $(\Theta_p)_p^{-1} = \Theta_p$ if $(\Theta_p)_p^{-1} \subseteq \Theta_p$. But this is clearly the case by Lemma 3.11. □

We have $R_p(a, a_p^{-1})$ for all $a \in \Theta_p$. Now, Proposition 3.2 together with Proposition 3.17 gives $\Theta_p \sim_{R_p} \Theta_p$. Using Proposition 3.4 and then Proposition 3.16 we finally have

$$\prod \Theta_p \equiv 1 \pmod{p}$$

Thus, we have formalized the "pairing off" using a more abstract approach.

Note that we do not utilize Fermat's Little Theorem in this formalization.

4 Mechanization

In this section we give an overview of our mechanization in Isabelle/HOL of The Chinese Remainder Theorem, Fermat's Little Theorem and Wilson's Theorem.

The mechanization is based on existing theories of integers in Isabelle/HOL developed up to and including the operators mod and div.

We thus start our mechanization from the same point as the development in Section 2, i.e. with the definition of the divides, congruence and gcd relations.

```
consts
  zprime   :: int set
  dvd      :: [int,int] => bool       (infixl 70)
  zcong    :: [int,int,int] => bool  ("[_ = _]'(mod _')")
  is_zgcd  :: [int,int,int] => bool

defs
  zprime_def    "zprime ==
                   {p. #1<p & (ALL m. m dvd p --> m=#1 | m=p)}"
  dvd_def       "m dvd n == EX k. n=m*k"
  is_zgcd_def   "is_zgcd p m n ==
                   #0 < p & p dvd m & p dvd n &
                   (ALL d. d dvd m & d dvd n --> d dvd p)"
  zcong_def     "[a = b](mod m) == m dvd (a-b)"
```

Some comments on syntax and conventions: The names of the definitions (except dvd) are preceded with a z to signify that the definitions are over integers ($\mathbb{Z}$). A parallel theory defines the same concepts (except congruence) over the natural numbers hence we choose to distinguish the defined names so as to avoid the use of namespaces. In the case of dvd this is not necessary, as we can use a overloaded version for both natural numbers and integers. Note also that we introduce a special (more readable) syntax for congruence. Furthermore, notice that integer numerals are identified by prefixing a #.

We also define Euclid's Algorithm and Euclid's Extended Algorithm by means of general recursive function definitions. Below we only give the definition of the standard algorithm.

```
consts
  zgcd  :: "int*int => int"

recdef zgcd "measure ((%(m,n).(nat n)) ::int*int=>nat)"
    simpset "simpset() addsimps [pos_mod_bound]"
    "zgcd (m, n) = (if n<=#0 then m else zgcd(n, m mod n))"
```

The simpset clause is necessary if we want the termination condition of the function to be proved automatically.

This definition requires the second argument to be positive for the function to be correct. This is no limitation as $\gcd(m, -n) = \gcd(m, n)$ for arbitrary integers m, n.

We have proven many theorems based on these definitions including all those of Proposition 2.2 and Proposition 2.3.

As mentioned, there exists a parallel theory based on the natural numbers. This development on the integers include all theorems proved there plus several others. Furthermore, it includes theorems concerning congruence and the extended gcd.

4.1 The Chinese Remainder Theorem

There are essentially no surprises in the mechanization of Theorem 2.9. The only important decision to make is how to formulate the theorem in the language of Isabelle/HOL. We choose to let the indexed integers (m_i, k_i and b_i) be represented by functions from $\mathbb{N}$ to $\mathbb{Z}$. Then the theorem can be formulated using the following definitions

```
consts
  m_cond      :: [nat,nat => int] => bool
  km_cond     :: [nat,nat => int,nat => int] => bool
  lincong_sol :: [nat,nat => int,nat => int,nat => int,int] => bool

defs
  m_cond_def   "m_cond n mf ==
                  (ALL i. i<=n --> #0 < mf i) &
                  (ALL i j. i<=n & j<=n & i ~= j -->
                            zgcd(mf i,mf j) = #1)"

  km_cond_def  "km_cond n kf mf ==
                  (ALL i. i<=n --> zgcd(kf i,mf i) = #1)"

  lincong_sol_def "lincong_sol n kf bf mf x ==
                  (ALL i. i<=n -->
                          [(kf i)*x = (bf i)](mod (mf i))"
```

such that we can prove

```
Goal "[| 0<n; m_cond n mf; km_cond n kf mf |]
      ==> (EX! x. #0 <= x & x < (funprod mf 0 n) &
                    (lincong_sol n kf bf mf x))";
```

in a mechanized way which follows the mathematical proof quite closely.

The funprod mf 0 n gives the product of the integers mf i where i runs from 0 to n.

4.2 Finite Sets

In both the formalizations of Fermat's and Wilson's Theorems finite sets play an important role. In Isabelle/HOL there is a theory developing notions of finite sets. This development is based on inductive definitions of the finiteness of a set, the cardinality of a finite set, and a fold function on finite set which can be used e.g. to define sums and products (cf. setprod below) of the elements of a finite set.

To use this development in an appropriate way we choose to use a definition of concrete finite sets which matches the inductive definitions. This is done by using recursive definitions of finite sets of integers.

Take e.g. the set $\Theta_p = \{n \mid 1 < n < p-1\}$. We give a recursive definition in Isabelle/HOL as follows:

```
recdef Theta "measure ((%a.(nat a)) ::int=>nat)"
    "Theta a = (if #1<a then insert a (Theta (a-#1)) else {})"
```

By induction over this definition it is now straightforward to show

```
Goal "b:(Theta a) <-> #1<b & b<=a";
```

This means that Θ_p in Isabelle/HOL is represented by the set Theta (p-2).

It is also fairly simple by induction to show the following useful fact

```
Goal "setprod(Theta a) = zfact a";
```

where setprod(A) calculates the product of the elements of A and zfact a calculates the faculty of a.

4.3 Bijection Relations

The mechanization of Fermat's and Wilson's Theorems very closely follows the formalization described in Section 3.

Thus we need to mechanize the bijection relations as developed in Section 3.1. Fortunately, it is very easy to define such inductive definitions as those of Definition 3.1 and Definition 3.3 in Isabelle/HOL.

The relation $\sim_P$ can be defined almost verbatim as follows

```
inductive "bijR P"
intrs
  empty  "({},{}) : bijR P"
  insert "[| P a b; a ~: A; b ~: B; (A,B) : bijR P |]
        ==> (insert a A, insert b B) : bijR P"
```

such that `(A,B) : bijR P` expresses $A \sim_P B$. We can similarly give an inductive definition such that `A : bijER P` expresses $A \in \mathrm{BS}_P$.

As it turns out, it is not possible to reproduce directly the proof of Proposition 3.4 in Isabelle/HOL. In fact, we have to prove a stronger result:

$$\text{If} \quad A \sim_P B \quad \text{then} \quad \forall F.\ F \subseteq A \wedge F \subseteq B \rightarrow F \in \mathrm{BS}_P$$

But this only holds if we put some extra conditions on P. Their Isabelle/HOL definitions are as follows:

```
consts
  bijP  :: "(['a, 'a] => bool) => 'a set => bool"
  uniqP :: "(['a, 'a] => bool) => bool"
  symP  :: "(['a, 'a] => bool) => bool"

defs
  bijP_def   "bijP P F == (ALL a b. a:F & P a b --> b:F)"
  uniqP_def "uniqP P == (ALL a b c d. P a b & P c d -->
                                         (a=c) = (b=d))"
  symP_def   "symP P  == (ALL a b. (P a b) = (P b a))"
```

We can now prove

```
Goal "[| (A,B) : bijR P; uniqP P; symP P |] \
\      ==> (ALL F. (bijP P F) & F<=A & F<=B --> F : bijER P)";
```

from which trivially follows

```
Goal "[| (A,A) : bijR P; (bijP P A); uniqP P; symP P |]
     ==> A : bijER P";
```

which is Proposition 3.4 with extra conditions on P.

4.4 Fermat's Little Theorem

The mechanization of Euler's generalized version of Fermat's Little Theorem follows the formalization of Section 3.2 closely.

Inductive definitions and recursive definitions of finite sets are handled as discussed in previous sections.

We end up showing

```
Goal "[| #0<m; zgcd(x,m) = #1 |] ==> [x^phi(m) = #1](mod m)";
```

and the easily derivable corollary

```
Goal "[| p:zprime; ~p dvd x |] ==> [x^(nat(p-#1)) = #1](mod p)";
```

for the case of the modulo being prime.

4.5 Wilson's Theorem

We have mechanized both the concrete and the abstract formalizations of Section 3,3.

There are as before no surprises in the mechanization when we follow the formalizations of Section 3.3.

In both cases we end up proving

```
Goal "p:zprime ==> [zfact(p-#1) = #-1](mod p)";
```

Notice that the relation R_p used in the abstract approach easily satisfies the extra conditions required as discussed in Section 4.3

4.6 Related Work

Boyer and Moore have mechanized Fermat's Little Theorem [3, 4] in their theorem prover [2]. They used the approach discussed in Section 3.2.1. They did not prove Euler's generalized version.

Russinoff mechanized Wilson's Theorem [8] in Boyer–Moore's Theorem Prover using the concrete approach discussed in Section 3.3.1.

Théry [10] compares mechanizations in Coq, HOL and PVS of Fermat's Little Theorem based on the proof using the binomial expansion.

5 Conclusion

We have presented a formalization and mechanization (in Isabelle/HOL) of basic number theory including two theorems of Fermat and Wilson.

We used a generalized approach for handling the concepts of one-to-one correspondences and "pairing off".

Comparing the generalized approach with existing approaches is most easily done with respect to the formalizations of Wilson's Theorem as we mechanized both the concrete and the abstract approach in Isabelle/HOL.

It is our claim that the abstract approach gives a cleaner and more modular presentation closer to the original mathematical proof.

When it comes to quantity (number of proof steps) the two developments are comparable but if one ignores the bijection relation part the abstract approach gets noticeably shorter. A reason for doing this is that once the "machinery" for handling the bijection relations is in place it can be used unchanged in other contexts as well (cf. the proof of Fermat's Little Theorem).

We believe that the usefulness of the generalized approach is not limited to the developments of this document. Russinoff has given a mechanical proof of Quadratic Reciprocity [9] (using the Boyer-Moore system) where at least one place the idea of "pairing off" is used (we have not studied this mechanization in detail). It seems very likely that a similar development (in eg. Isabelle/HOL) could benefit from the generalized approach using bijection relations.

Acknowledgements

This work was carried out while the author was visiting the Computer Laboratory, University of Cambridge. The author would like to thank Larry Paulson for many valuable comments and suggestions, and for always being willing to discuss Isabelle and formalized mathematics in general.

References

[1] Alan Baker. *A Concise Introduction to the Theory of Numbers.* Cambridge University Press, 1984.

[2] R.S. Boyer and J.S. Moore. *A Computational Logic.* Academic Press, 1979.

[3] R.S. Boyer and J.S. Moore. Proof Checking the RSA Public Key Encryption Algorithm. Technical Report ICSCA-CMP-33, Institute for Computing Science and Computer Applications, University of Texas at Austin, 1982.

[4] R.S. Boyer and J.S. Moore. Proof Checking the RSA Public Key Encryption Algorithm. *American Mathematical Monthly*, 91(3):181–189, 1984.

[5] H. Davenport. *The Higher Arithmetic.* Cambridge University Press, sixth edition, 1992.

[6] Douglas J. Howe. Implementing Number Theory: An Experiment with Nuprl. In *Automated Deduction, CADE-8*, volume 230 of *Lecture Notes in Computer Science*, pages 404–415. Springer-Verlag, 1986.

[7] Lawrence C. Paulson. *Isabelle, A Generic Theorem Prover*, volume 828 of *Lecture Notes in Computer Science.* Springer-Verlag, 1994.

[8] David M. Russinoff. An Experiment with the Boyer-Moore Theorem Prover: A Proof of Wilson's Theorem. *Journal of Automated Reasoning*, 1:121–139, 1985.

[9] David M. Russinoff. A Mechanical Proof of Quadratic Reciprocity. *Journal of Automated Reasoning*, 8:3–21, 1992.

[10] Laurent Théry. *Comparing Coq, HOL, PVS on a simple proof of the RSA Public Key Encryption Algorithm.* INRIA Sophia-Antipolis, March 2000. `coq.inria.fr/seminaires/comparaison/`.