A Page in Number Theory

Andrea Asperti, Cristian Armentano
Dipartimento di Scienze dell'Informazione
Università degli Studi di Bologna

We discuss the formalization, in the Matita Interactive Theorem Prover, of a few elementary results in number theory about the Möbius μ function and the Euler ϕ function.

1. INTRODUCTION

The title of the paper must be understood in a strictly literal sense: we discuss the formalization, in the Matita Interactive Theorem Prover [2, 3], of *one* page of a traditional graduate textbook in Number Theory, and precisely the portion of text spanning from line −9 of page 19 to line −6 of page 20 in Ireland and Rosen's book "A Classical Introduction to Modern Number Theory" [10] [1](see appendix A).
The page is quite dense; the main notions and results are the following:

—definition of the Möbius μ function
—proof of $\Sigma_{d|n}\mu(d) = 0$
—definition of the Dirichlet composition
—proof of the Möbius Inversion Theorem
—definition of the Euler ϕ function
—proof of $\Sigma_{d|n}\phi(d) = n$

Although the results are relatively elementary, the only formalization we are aware of is by J.Avigad and other people, in Isabelle, as part of their proof of the prime number theorem [6] . Intentionally, we did not look at [6] during our development, since we did not want to be biased in our choices by a previous formalization. However, a detailed comparison was done after the completion of the work, as discussed in salient points along the paper. The style of the presentation will strictly follow the original mathematical text, quoting the source (in framed boxes) and discussing step by step its formalization. We shall just make a single large detour concerning iteration and several theorems about its invariance under permutation of the inputs (under the suitable conditions).

At the moment we started the work, the mathematical library of Matita already contained the proof of the Fundamental Theorem of Arithmetic (unicity of the decomposition in prime factors), Fermat's Little Theorem and its generalization to the Euler function, namely the fact that for any a coprime to n, $a^{\phi(n)} \equiv 1 \; mod \; n$.

The results contained in this paper have been presented in a two-hour lesson given by the first author at the Types Summer School held Bertinoro in August 2007. All the scripts are accessible from the Matita Home Page.

[1]Among the several texts we consulted, we found [10] particularly accessible, only requiring a familiarity with basic abstract algebra.

2. THE MÖBIUS μ FUNCTION

> "... We now introduce a very important arithmetic function, the Möbius μ function.
> For $n \in Z^+$, $\mu(1) = 1$, $\mu(n) = 0$ if n is not square-free, and $\mu(p_1 p_2 \dots p_l) = (-1)^l$, where the p_i are distinct positive primes."

The above definition of the μ function is essentially based on the composition of two other functions: the factorization function (already available in Matita), and a function counting the number of the distinct primes (provided no prime has an order greater than 1 in the decomposition, otherwise the result is 0). We could formalize μ as such a composition[2] or give a more direct definition, essentially obtained by a partial evaluation of the former. According to our experience, working with a direct definition is usually simpler.
The definition relies on the following auxiliary functions of the Matita library :

—`(p_ord n p)` returns a pair $\langle q, r\rangle$, where q is the *order* (or multiplicity) of p in n and r is the remaining, i.e. $n = p^q r$, where p does not divide r (for $n = 0$ it returns the default pair $\langle 0, 0\rangle$);

—`(nth_prime i)` is the i-th prime number in progressive order;

—`(max_prime_factor n)` is the index of the largest prime in n, in the enumeration of all primes;

—`Z`, `OZ`, `oneZ` and `Zopp` are respectively the type of integers, its zero, one, and the function taking n to $-n$.

The definition is straightforward: we start looking for the max prime factor p_k in n, and then call an auxiliary function, defined by primitive recursion over k that computes the order of p_k in n, returns 0 if it is greater than 1 and otherwise recurs on the remainder, changing the sign of the result.

```
let rec moebius_aux p n : Z :=
  match p with
  [ O ⇒ oneZ
  | (S p1) ⇒
    match p_ord n (nth_prime p1) with
    [ (pair q r) ⇒
      match q with
      [ O ⇒ moebius_aux p1 r
      | (S q1) ⇒
        match q1 with
        [ O ⇒ Zopp (moebius_aux p1 r)
        | (S q2) ⇒ OZ
        ]
      ]
    ]
  ].
```

[2] This is the approach in Isabelle, by D.Gray.

```
definition moebius n: Z :=
  let p := max_prime_factor n in moebius_aux (S p) n.
```

We remind the reader that Matita is based on the same foundational framework as Coq - the Calculus of Inductive Constructions - thus embedding a notion of normalization into its logic. As a consequence, all definitions are *live* and can be evaluated just normalizing the involved expression.

3. $\Sigma_{D|N}\mu(D) = 0$

Proposition. If $n > 1$, $\Sigma_{d|n}\mu(d) = 0$.
Proof. If $n = p_1^{a_1}p_2^{a_2}\dots p_l^{a_l}$, then

$$\Sigma_{d|n}\mu(d) = \Sigma_{(\epsilon_1,\dots,\epsilon_l)}\mu(p_1^{\epsilon_1}\dots p_l^{\epsilon_l})$$

where the ϵ_i are zero or one. Thus

$$\Sigma_{d|n}\mu(d) = 1 - l + \binom{l}{2} - \binom{l}{3} + \dots + (-1)^l = (1-1)^l = 0$$

REMARK 1. *The previous proof is standard. For instance, it can also be found in [9] (Theorem 262, p.235), [11] (Theorem 2.2.5, p.63), [1](Theorem 2.1, p.25).*

Existence and unicity of the decomposition into prime factors has been already proved in Matita, and we may assume to have the list $[a_1; a_2; \dots a_l]$ such that $n = p_1^{a_1}p_2^{a_2}\dots p_l^{a_l}$. We may also easily (re-)define μ taking as input $[a_1; a_2; \dots a_l]$, and prove that if $\mu(n) \neq 0$ then a_i is either one or zero (for the square-free condition). It is also possible to prove that d divides n if and only if d admits a decomposition of the kind $p_1^{b_1}p_2^{b_2}\dots p_l^{b_l}$ where $b_i \leq a_i$. Notwithstanding all these property, the equation

$$\Sigma_{d|n}\mu(d) = \Sigma_{(\epsilon_1,\dots,\epsilon_l)}\mu(p_1^{\epsilon_1}\dots p_l^{\epsilon_l}) \tag{1}$$

is still far away. The point is that in order to formalize the statement we would need a multi-dimensional sum $\sum_{\vec{x}\leq\vec{a}} f(\vec{x})$ parametric in n, $\vec{a} \in N^n$ and $f : N^n \to N$, whose definition is relatively complex, due to the dependent type N^n.
The alternative approach followed in [6] is the following. Define the *radical* $rad(n)$ of a number n to be the greatest square free number dividing n. Then $\sum_{d|n}\mu(d) = \sum_{d|rad(d)}\mu(d)$. Finally, for some prime factor p of n write

$$\sum_{d|rad(d)}\mu(d) = \sum_{d|rad(d),p|d}\mu(d) + \sum_{d|rad(d),p\not|d}\mu(d)$$

and prove that each term in the first sum is canceled by a corresponding term of the second.
We followed still another route. Let p be a prime factor of n, and assume its order in n is a, i.e. $n = p^a r$ and $p \not| r$. Then

$$\Sigma_{d|p^a r}\mu(d) = \Sigma_{d|r}\Sigma_{i\leq a}\mu(p^i d) \tag{2}$$

Since $p \not| r$ then $p \not| d$ and hence $\mu(pd) = -\mu(d)$.
Moreover, for any $i > 1$, $\mu(p^i d) = 0$ since it is not square free. Hence, for any d,

$$\begin{aligned}\Sigma_{i\leq a}\mu(p^i d) &= \mu(d)+\mu(pd)+\Sigma_{2\leq i\leq a}\mu(p^i d)\\ &= \mu(d)-\mu(d)\\ &= 0\end{aligned}$$

Formalizing the last row of equations is not particularly problematic. More complex is the formal proof of equation (2). The problem can be essentially decomposed into two subproblems:

(1) getting rid of the nested sum;
(2) proving that the sum is invariant under permutation of its addends.

We shall separately discuss the two problems in the following subsections. For the sake of clarity, we shall stick to sums, but all results may be generalized to products and, more generally, to arbitrary iterative constructs, provided the iterated function is commutative, associative, and admits a neutral element. In Matita, all results have been proved in the generalized form; specific cases are derived by instatiation (they are still usefull for readibility, and notational purposes).

3.1 Nested Sums

Let us write

$$\sum_{i<n,p(i)} g(i)$$

for the sum of all $g(i)$ where i ranges over all integers less than n which satisfy the boolean condition p (we shall sometimes omit the bound when it can be inferred by the condition). We prefer boolean conditions to generic predicates for computability reasons, but the following discussion can be easily generalized to the formers.
The above function is easily defined in a formal way by induction on n:

```
let rec sigma_p n p g =
match n with
[ O ⇒ O
| S k ⇒ match p k with
    [true ⇒ g(k)+(sigma_p k p g)
    | false ⇒ (sigma_p k p g)
    ]
].
```

The following statement reduces a nested sum to a single sum:

$$\sum_{i<n,p_1(i)}\ \sum_{j<m,p_2(j)} g(i,j) \;=\; \sum_{\substack{k<n\cdot m\\ p_1(k/m)\\ p_2(k\ mod\ m)}} g(k/m, k\ mod\ m) \tag{3}$$

Formally:

```
theorem sigma_p2:
∀ n,m: nat.
∀ p1: nat → bool.
∀ p2: nat → nat → bool.
∀ g: nat → nat → nat.
```

```
sigma_p n p1
  (λ x.sigma_p m (p2 x) (g x)) =
sigma_p (n*m)
  (λ x.andb (p1 (div x m)) (p2 (div x m) (mod x m)))
  (λ x.g (div x m) (mod x m)).
```

The only interesting remark, here, is the type of the inner boolean test function p_2, that does also depend, in general, on the outer index (a fact that is often transparent in the usual mathematical notation). The theorem is proved by induction on n and relies on the following lemma (proved, in turn, by induction over k):

$$\sum_{i<k+n,p(i)} g(i) = \sum_{i<k,p(i+n)} g(i+n) + \sum_{i<k,p(i)} g(i)$$

Here is the formal statement:

```
theorem sigma_p_plus: ∀n,k:nat.∀ p:nat → bool.
∀ g: nat → nat.
  sigma_p (k + n) p g
  = sigma_p k (λ x.p (x+n)) (λ x.g (x+n)) + sigma_p n p g.
```

3.2 Invariance under permutation

The second important result is the invariance of the sum under permutation of its addends:

$$\sum_{x<n_1,p_1(x)} g(h(x)) = \sum_{x<n_2,p_2(x)} g(x) \tag{4}$$

provided h is a bijection from $\{x < n_2 : p_2(x)\}$ to $\{x < n_1 : p_1(x)\}$. The bijectivity of h can be formalized in many different ways; we choose to explicitly provide the inverse function:

```
theorem eq_sigma_p_gh:
∀ g: nat → Z.
∀ h,hinv: nat → nat.∀ n1,n2.
∀ p1,p2:nat → bool.
(∀ i. i < n1 → p1 i = true → p2 (h i) = true) →
(∀ i. i < n1 → p1 i = true → hinv (h i) = i) →
(∀ i. i < n1 → p1 i = true → h i < n2) →
(∀ j. j < n2 → p2 j = true → p1 (hinv j) = true) →
(∀ j. j < n2 → p2 j = true → h (hinv j) = j) →
(∀ j. j < n2 → p2 j = true → hinv j < n) →
  sigma_p n1 p1 (λ x.g(h x)) = sigma_p n2 (λ x.p2 x) g.
```

There is an interesting remark to be made here. Consider the same result in the particular case when the boolean conditions p_1 and p_2 are true. That implies that $n_1 = n_2 = n$ and h must be a permutation of the first n integers. The idea of proving the result by induction on n does obviously face the difficulty that h is not, in general, a permutation of the first $n-1$ integers. The most elegant solution, in this case, is to introduce the elementary notion of *transposition*

$$\begin{pmatrix} n \\ m \end{pmatrix}$$

between two natural numbers n and m, that is the operation swapping n and m. This allows us to reduce the problem to the following simpler statement: for all $n, m < k$

$$\sum_{i<k} g(i) = \sum_{i<k} g(\begin{pmatrix} n \\ m \end{pmatrix} i) \tag{5}$$

Still, the previous equation is not entirely trivial, requiring a complete (course of values) induction on the distance between n and m.

Somewhat surprisingly, the most general case of conditional sums turns out to be simpler, since we may take advantage of the boolean condition to get rid of part of the problems. The key lemma, in this case, is the following, almost trivial remark: for any $a < k$ such that $p(a)$

$$\sum_{i<k, p(i)} g(i) = g(a) + \sum_{\substack{i<k \\ p(i) \\ i \neq a}} g(i) \tag{6}$$

Then, the proof of equation (3) can be done by induction over n_2. If $n_2 = 0$, then it is easy to prove that $p_1(x)$ must be false for any $x < n_1$. Suppose $n_2 = k + 1$. If $p_2(k+1) = \mathit{false}$ then

$$\sum_{i<k+1, p_2(i)} g(i) = \sum_{i<k, p_2(i)} g(i)$$

and the results follows by induction (provided you prove that h and h^{-1} still define a bijection in the restricted interval). If $p_2(k+1) = \mathit{true}$ then

$$\begin{aligned}
\sum_{x<k+1, p_2(x)} g(x) &= g(k+1) + \sum_{x<k, p_2(x)} g(x) \\
&= g(k+1) + \sum_{\substack{x < n_1, p_1(x) \\ x \neq h^{-1}(k+1)}} g(h(x)) && \text{by induction} \\
&= g(h(h^{-1}(k+1))) + \sum_{\substack{x < n_1, p_1(x) \\ x \neq h^{-1}(k+1)}} g(h(x)) \\
&= \sum_{x<n_1, p_1(x)} g(h(x)) && \text{by (5)}
\end{aligned}$$

3.3 A composite statement

It is worth to combine together the two equations 3 and 4:

$$\sum_{x<k, p_1(x)} g(x) = \sum_{i<n, p_{21}(i)} \sum_{j<m, p_{22}(j)} g(h_2(i,j)) \tag{7}$$

provided h_2 is a bijection between

$$\{(i,j) < (n,m) : p_{21}(i) \wedge p_{22}(i,j)\}$$

and

$$\{x < k : p_1(x)\}$$

Formally:

```
theorem sigma_p_knm:
∀ g: nat → nat.
∀ h2:nat → nat → nat.
∀ h11,h12:nat → nat.
∀ k,n,m.
∀ p1,p21:nat → bool.
∀ p22:nat → nat → bool.
(∀ x. x < k → p1 x = true →
  p21 (h11 x) = true ∧p22 (h11 x) (h12 x) = true
  ∧ h2 (h11 x) (h12 x) = x ∧(h11 x) < n ∧(h12 x) < m) →
(∀ i,j. i < n → j < m
  → p21 i = true → p22 i j = true →
  p1 (h2 i j) = true ∧h11 (h2 i j) = i ∧ h12 (h2 i j) = j ∧ h2 i j < k) →
sigma_p k p1 g = sigma_p n p21 (λ x:nat.sigma_p m (p22 x) (λ y. g (h2 x y))).
```

Starting from 7, equation 2 is now easy to prove: it amounts to check that the function $h_1(d,i) = p^i d$ defines an isomorphism between $\{(d,i) \leq (r,a) : d|r\}$ and $\{x \leq p^a r : x|p^a r\}$ (under the essential assumption that p does not divides r). The inverse function is just (the symmetric of) `p_ord`, and the injective nature of `p_ord` was a key lemma for the fundamental theorem of arithmetic, already proved in the library.

4. DIRICHLET MULTIPLICATION

> "The full significance of the Möbius μ function can be understood most clearly when its connection with Dirichlet multiplication is brought to light. Let f and g be complex valued functions on Z^+. The Dirichlet product of f and g is defined by the formula $f \otimes g(n) = \Sigma f(d_1)g(d_2)$ where the sum is over all pairs (d_1, d_2) of positive integers such that $d_1 d_2 = n$."

As remarked by the authors themselves a few lines later, the same definition can be given whenever the codomain of f and g is an any abelian group, preserving all properties of the multiplication. For the sake of simplicity, we consider Z itself. This is our definition in Matita.

```
definition dirichlet_product f g n: Z :=
  sigma_p (S n) (λ d.divides_b d n) (λ d. (f d)*(g (div n d))).
```

Up to our knowledge, there is no *explicit* definition of the Dirichlet composition in [6], although it is implicitly used in most of their results.

> "This product is associative, as one can see by checking that $f\otimes(g\otimes h)(n) = (f \otimes g) \otimes h(n) = \Sigma f(d_1)g(d_2)h(d_3)$ where the sum is over all 3-tuples (d_1, d_2, d_3) of positive integers such that $d_1 d_2 d_3 = n$."

The associativity property looks absolutely trivial. Unfortunately, its formalization is not. The gap between the simple mathematical remark behind the previous "proof", and the actual complexity of its formal counterpart is one of the most astonishing points in the formalization of these results in number theory. Avigad and his coauthors experienced a similar puzzling sensation:

> "This type of "reindexing" is often so transparent in mathematical arguments that when we first came across an instance where we needed it [...], it took some thought to identify the relevant principle". [6]

The point is the following. By definition

$$f \otimes g(n) = \Sigma_{d|n} f(d) g(n/d)$$

Hence

$$\begin{aligned} f \otimes (g \otimes h)(n) = \\ &= \Sigma_{d_1|n} f(d_1) g \otimes h(n/d_1) \\ &= \Sigma_{d_1|n} f(d_1) (\Sigma_{d_2|(n/d_1)} g(d_2) h((n/d_1)/d_2)) \\ &= \Sigma_{d_1|n} \Sigma_{d_2|(n/d_1)} f(d_1) g(d_2) h((n/d_1)/d_2) \end{aligned}$$

while

$$\begin{aligned} (f \otimes g) \otimes h(n) = \\ &= \Sigma_{d_1|n} f \otimes g(d_1) h(n/d_1) \\ &= \Sigma_{d_1|n} (\Sigma_{d_2|d_1} f(d_2) g(d_1/d_2)) h(n/d_1) \\ &= \Sigma_{d_1|n} \Sigma_{d_2|d_1} f(d_2) g(d_1/d_2)) h(n/d_1) \end{aligned}$$

In order to prove the equivalence of the two summations, we must hence exploit the bijection between the following sets:

$$A = \{(d_1, d_2) \leq (n, n) : d_1|n, d_2|(n/d_1)\}$$

$$B = \{(d_1, d_2) \leq (n, n) : d_1|n, d_2|d_1\}$$

The bijection is easily found: $h : A \to B$ and $h^{-1} : B \to A$ are the following functions

$$h(d_1, d_2) = (d_1 d_2, d_1)$$

$$h^{-1}(d_1, d_2) = (d_2, d_1/d_2)$$

However, checking that they are inverse of each other (on the given domains) require quite a lot of tedious computations. Even worse, with the results mentioned so far, we should first get rid of (at least one of the) nested sum, that would furtherly complicate the notation, mixing the division operations of the Dirichlet product with that of equation 3. All statements would become quite unreadable. The only solution is to give still another version of equations 4 and 7, dealing with nested sums on both sides:

$$\sum_{i<n_1, p_{11}(i)} \sum_{j<m_1, p_{12}(i,j)} g(i,j) = \sum_{i<n_2, p_{21}(i)} \sum_{j<m_2, p_{22}(i,j)} g(h_{21}(i,j), h_{22}(i,j)) \qquad (8)$$

provided $\langle h_{21}, h_{22} \rangle$ is a bijection between

$$\{(i,j) < (n_1, m_1) : p_{11}(i) \wedge p_{12}(i,j)\}$$

and

$$\{(i,j) < (n_2, m_2) : p_{21}(i) \wedge p_{22}(i,j)\}$$

(with inverse function $\langle h_{11}, h_{12} \rangle$). Formally:

```
theorem sigma_p2_eq:
∀ g: nat → nat → Z.
∀ h11,h12,h21,h22: nat → nat → nat.
∀ n1,m1,n2,m2.
∀ p11,p21:nat → bool.
∀ p12,p22:nat → nat → bool.
(∀ i,j. i < n2 → j < m2 → p21 i = true → p22 i j = true →
  p11 (h11 i j) = true ∧p12 (h11 i j) (h12 i j) = true
  ∧ h21 (h11 i j) (h12 i j) = i ∧ h22 (h11 i j) (h12 i j) = j
  ∧ h11 i j < n1 ∧h12 i j < m1) →
(∀ i,j. i < n2 → j < m2 → p11 i = true → p12 i j = true →
  p21 (h21 i j) = true ∧p22 (h21 i j) (h22 i j) = true
  ∧ h11 (h21 i j) (h22 i j) = i ∧ h12 (h21 i j) (h22 i j) = j
  ∧ h21 i j < n2 ∧h22 i j < m2) →
sigma_p n1 p11 (λ x:nat .sigma_p m1 (p12 x) (λ y. g x y)) =
sigma_p n2 p21 (λ x:nat .sigma_p m2 (p22 x) (λ y. g (h11 x y) (h12 x y))).
```

5. THE INVERSION THEOREM

"Define the function T by $T(1) = 1$ and $T(n) = 0$ for $n > 1$. Then $f \otimes T = T \otimes f = f$."

$$\begin{aligned} f \otimes T(n) &= \sum_{d|n} f(d)T(n/d) && \text{by def. of } \mu \\ &= f(n)T(n/n) + \sum_{d<n, d|n} f(d)T(n/d) && \text{by eq. 6} \\ &= f(n) + \sum_{d<n, d|n} 0 \\ &= f(n) \end{aligned}$$

The proof that $T \otimes f = f$ is slightly more involved. The best solution is to prove once and for all the commutativity of the Dirichlet composition. It is interesting to observe that, from the mathematical point of view, such a property is *so evident* that the authors do not even care to mention it! For the formal proof you have to exploit the permutation mapping d in n/d between divisors of n.
A second remark to do is about extensionality. Matita is based on the Calculus of Inductive Constructions, which is not an extensional theory. So, by the fact that for any n, $f \otimes T(n) = f(n)$ we are not authorized to conclude that $f \otimes T = f$, but we must stumble along with the former, weaker equivalence.

"Define I by $I(n) = 1$ for all n. Then $f \otimes I(n) = I \otimes f(n) = \Sigma_{d|n} f(d)$."

Prove $f \otimes I(n) = f(n)$ and then use commutativity for the other equality. Again, proving directly $I \otimes f(n) = f(n)$ is far more complex.

"Lemma. $I \otimes \mu = \mu \otimes I = T$.
Proof. $\mu \otimes I(1) = \mu(1)I(1) = 1$. If $n > 1$, $\mu \otimes I(n) = \Sigma_{d|n}\mu(d) = 0$.

This is easy.

> "Theorem (Möbius Inversion Theorem).
> Let $F(n) = \Sigma_{d|n} f(d)$. Then $f(n) = \Sigma_{d|n} \mu(d) F(n/d)$.
> Proof. $F = f \otimes I$. Thus $F \otimes \mu = (f \otimes I) \otimes \mu = f \otimes (I \otimes \mu) = f \otimes T = f$.
> This shows that $f(n) = F \otimes \mu(n) = \Sigma_{d|n} \mu(d) F(n/d)$.

The formal proof essentially follows the same line, but is not as elegant, mostly due to extensionality problems. Let us write $\cong$ for the extensional equivalence of functions, i.e. $f \cong g$ iff $\forall n. f(n) = g(n)$. For instance, if we only know $F \cong f \otimes I$ we cannot just *rewrite* $F \otimes \mu$ into $(f \otimes I) \otimes \mu$, since rewriting only works with Leibniz (intensional) equality. The right way to proceed, here, is to prove first prove the congruence of $\otimes$ with respect to extensional equality, namely

$$f \cong f' \to g \cong g' \to f \otimes g \cong f' \otimes g'$$

The complete Matita proof can be found in Appendix B.

6. $\sum_{D|N} \phi(D) = N$

> "... The Möbius inversions has many applications. We shall use it to obtain formula for yet another arithmetic function, the Euler ϕ function. For $n \in Z^+$, $\phi(n)$ is defined to be the number of integers between 1 and n relative prime to n. For example, $\phi(1) = 1$, $\phi(5) = 4$, $\phi(6) = 2$ and $\phi(9) = 6$. If p is a prime, it is clear that $\phi(p) = p - 1$."

The Euler ϕ function (sometimes called *totient* function) can be simply expressed in terms of our sum operation as follows:

$$\phi(n) = \sum_{gcd(i,n)=1}^{n} 1$$

where $gcd(m, n)$ is the *greatest common divisor* of m and n.
Since the `gcd` function already belongs to the Matita library, the previous definition can be directly translated into its formal counterpart:

```
definition totient n: nat :=
  sigma_p n (λ m. eqb (gcd m n) 1) (λ m.1).
```

Note that, for simplicity, the sum stops at $n - 1$, so we do not count n as one of the possible integers relative prime to itself: if $n = 1$, our sum correctly returns 1, while if $n > 1$, $gcd(n, n) = 1$ is obviously false.

> **Proposition**. $\sum_{d|n} \phi(d) = n$.
> Proof. Consider the n rational numbers $\frac{1}{n}$, $\frac{2}{n}$, $\frac{3}{n}$, ..., $\frac{n-1}{n}$, $\frac{n}{n}$. Reduce each to lowest terms; i.e., express each number as quotient of relative prime integers. The denominators will all be divisors of n. If $d|n$, exactly $\phi(d)$ of our numbers will have d in the denominator after reducing to lowest terms. Thus $\sum_{d|n} \phi(d) = n$.

Before describing the formal version of this proof, let us try to make the previous proof a bit more explicit.

Let us define P_d as the number of fractions having, after the reduction, d in the denominator. We have to prove that $\forall d, d|n.P_d = \phi(d)$.
It is clear that after reducing to lowest terms, we obtain fractions in the form $\frac{p_i}{d_i}$, where $d_i|n$ and $gcd(p_i, d_i) = 1$. Let x be a divisor of n; at least P_x integers will be less or equal than n, and relative prime to it (that is the numerators of those fractions having x in the denominator), and so $P_x \geq \phi(x)$.
On the other hand, after the reduction there can't be any other fraction with denominator x, apart from those already considered. Suppose, towards obtaining a contradiction, that there's another fraction $\frac{k}{x}$, where $gcd(k, x) = 1$, and k isn't any of the P_x numerators introduced before. We are sure that $k \leq x$, because none of the initial fractions was greater than 1. Since $x|n$, and $n > 0$ by hypothesis, $\exists l > 0.n = x * l$. So the fraction $\frac{k*l}{x*l} = \frac{k*l}{n}$ is one of the n fraction before the reduction to lowest terms, because $k*l \leq x*l$, that is $k*l \leq n$. This fact guarantees that l must be one of the P_x numerator achieved after reducing to lowest terms all the fractions. So then $P_x \leq \phi(x)$.
Since $P_x \geq \phi(x)$ and $P_x \leq \phi(x)$, we have $P_x = \phi(x)$. It is now easy to obtain the thesis of the theorem.

6.1 The formal proof

The formal proof strongly relies on `sigma_p` properties. By unfolding ϕ with its definition we obtain

$$\sum_{d<n+1, d|n} \sum_{i<d, gcd(i,d)=1} 1 = n \tag{9}$$

It is clear how these nested sums represent the scenario of the fractions reduced to their lowest terms. The outer one is indexed over divisors of n, that is all the possible denominators. For each outer index d', the inner sum ranges over the possible numerators of fractions having, after the reduction, d' in the denominator. Inner indexes, in fact, must be relative prime to d', and less than it (here we have a little difference with the "handwritten" proof, since we don't work with $\frac{n}{n}$, but we consider $\frac{0}{n}$ instead, according to the definition of *totient* introduced before).
Our first problem is that the upper bound d of the inner sum depends on (in this case, *is*) the index of the outer one, while the library theorems dealing with nested sums (in particular, equation 7) do not consider this possibility. Instead of generalizing those results, in this case it is simpler to change the bound adjusting the boolean predicate of the inner sum (in order to maintain the same semantics) in the following way

$$\sum_{d<n+1, d|n} \sum_{i<d, gcd(i,d)=1} 1 = \sum_{d<n+1, d|n} \sum_{\substack{i<n+1, \\ gcd(i,d)=1 \\ i<d}} 1 \tag{10}$$

The previous transformation, as straightforward as it may seem, actually relies on a couple of technical lemmas, whose proof is yet another tedious exercise.
The first one states that you may extend the bound, provided the boolean condition

is false:

$$\sum_{i<m,p(i)} g(i) = \sum_{i<n,p(i)} g(i) \tag{11}$$

if $n \leq m$ and $p(i) = \mathit{false}$ for $n \leq i < m$.

```
theorem false_to_eq_sigma_p:
∀ n,m:nat.n \le m →
∀ p:nat → bool.
∀ g: nat → nat.
(∀ i:nat. n \le i → i < m → p i = false) →
  sigma_p m p g = sigma_p n p g.
```

The second one states that you may change the boolean condition (and the body of the sum), provided they have the same semantics on the given interval:

$$\sum_{i<n,p_1(i)} g_1(i) = \sum_{i<n,p_2(i)} g_2(i) \tag{12}$$

if $p_1(i) = p_2(i)$ for any $i < n$, and $g_1(i) = g_2(i)$ for any $i < n$ satisfying $p_1(i)$:

```
theorem eq_sigma_p1: ∀p1,p2:nat → bool.
∀ g1,g2: nat → nat.∀ n.
(∀ x.  x < n → p1 x = p2 x) →
(∀ x.  x < n → p1 x = true → g1 x = g2 x) →
  sigma_p n p1 g1 = sigma_p n p2 g2.
```

Using equation 10 we are left to prove

$$\sum_{d<n+1,d|n} \sum_{\substack{i<n+1\\ gcd(i,d)=1\\ i<d}} 1 = n \tag{13}$$

Let now t be the constant true boolean predicate; by induction over n it is very easy to prove that

$$n = \sum_{j<n,t(j)} 1$$

so that 13 can be rewritten as

$$\sum_{d<n+1,d|n} \sum_{\substack{i<n+1\\ gcd(i,d)=1\\ i<d}} 1 = \sum_{j<n+1,t(j)} 1$$

As described before, the nested sums allow us to represent reduced fractions. In this way its useful to think that the right dummy sum is indexed over the initial numerators (before reduction numerators are all the values in $[0, n[$). The aim is to complete the proof using theorem 7, finding a bijection between the couples of indexes in the nested sums and the indexes in the right, dummy sum. The variables of *theorem* 7 must be instantiated as follows:

—$g = (\lambda x.1)$

—$p_1 = (\lambda x.true)$
—$p_{21} = (\lambda d.d|n)$
—$p_{22} = (\lambda m, d.m < d \wedge gcd(m,d) = 1)$
—$h_2 = (\lambda d, i.i * (n/d))$
—$h_{11} = (\lambda j.n/(gcd(j,n)))$
—$h_{12} = (\lambda j.j/(gcd(j,n)))$

Let us also define sets A_1 and A_2 in this way:

$$A_1 = \{(d,i) < (n+1, n+1) | p_{21}(d) \wedge p_{22}(d,i)\}$$
$$A_2 = \{j | p_1(j)\} = [0, n[$$

Each couple $(a,b) \in A_1$ represents the fraction $\frac{a}{b}$ after the reduction, while each element in A_2 simply represents the numerator of one of the initial fractions (having n in the denominator) before the reduction. It is easy to see that h_2, applied to a couple $(x,y) \in A_1$, returns the numerator of fraction $\frac{x}{y}$ before the reduction, while h_{11} and h_{12}, applied to an element $w \in A_2$, return, respectively, the denominator and the numerator of the fraction $\frac{w}{n}$ after the reduction. This bijection between A_1 and A_2 is formalized in the two main hypothesis of theorem 7, and can be proved using simple properties of `gcd` and `divides` listed in appendix C.

6.2 Isabelle proof

We shall now have a look at the proof of $\sum_{d|n} \phi(d) = 0$ in Isabelle Library[3], making a comparison with our proof in Matita, and trying to show how these two different works share the same basic ideas.

While our proof is based on properties of sums, this one has a more set-theoretic flavor. Even ϕ is defined as the cardinality of a particular set, in the following way:

$$\phi(p) = |\{x \in N. 1 \leq x \leq p \wedge gcd(p,x) = 1\}| \tag{14}$$

The complete proof is composed of 23 elementary lemmas. We shall not see all them in detail but we shall just describe the main ideas of the proof, emphasizing the connections with the classic one introduced at the beginning of section 6.

Given $n \in N^+$, the proof starts with a few initial definitions:

—$S = \{x \in Z. 1 \leq x \leq n\}$
—$I = \{d \in Z. 1 \leq d \wedge d|n\}$
—$\forall d \in I. A(d) = \{k \in S. gcd(k,n) = d\}$
—$\forall d \in I. B(d) = \{m \in N. 1 \leq m \leq (n/d) \wedge gcd(m,(n/d)) = 1\}$
—$f : I \to N. f = \{\lambda x : I. n/x\}$

Even if they are not described explicitly in the original work, it is very useful to also define the following functions, for $d \in I$:

$$r_d(k) = k/d : A(d) \to N$$

[3] The complete proof is a part of Avigad's work, and can be found at
http://www.andrew.cmu.edu/user/avigad/isabelle/NumberTheory/EulerPhi.html

The two lemmas `A_inj_prop` and `image_A_eq_B` state, respectively, the following results:

$$\forall d \in I, r_d \text{ is injective on } A(d) \tag{15}$$
$$\forall d \in I, r_d(A(d)) = B(d) \tag{16}$$

It is very easy to see that the family of sets $A(i)$ (with $i \in I$) is a partition of S, and so

$$|S| = \sum_{i \in I} |A(i)| \tag{17}$$

Hence, we have:

$$\begin{array}{rcll} n & = & |S| & \\ & = & \sum_{d \in I} |A(d)| & \text{by property 17} \\ & = & \sum_{d \in I} |r_d(A(d))| & \text{by injectivity of } r_d \text{ on } A(d) \\ & = & \sum_{d \in I} |B(d)| & \text{by property 16} \end{array}$$

There is a clear relationship between sets $B(i)$ and the reduction to lowest terms described in the paper proof; for each divisor d of n, $B(d)$ contains all and only the possible numerators of reduced fractions, having $\frac{n}{d}$ in the denominator. In fact a generic element $x \in B(d)$ must be, by definition of $B(d)$, less or equal than $\frac{d}{n}$ (because all the considered fractions are not greater than 1), and relative prime to it (because these fractions are reduced to lowest terms).

We also see a clear connection with the nested sums 9 in Matita proof. In both cases the outer sum ranges over the possible divisors of n, and the inner sum gives a particular representation of the possible numerators in the reduced fractions. There is just a little difference: while our proof considers the divisors of n as possible denominators, in this proof the authors, given $d|n$, work with $\frac{n}{d}$ as denominator.

Of course, this is not a real difference because, if d ranges over the set of the possible natural divisor of n, $\frac{n}{d}$ ranges over the same set.

The remaining part of Isabelle proof is essentially meant to formalize this obvious equivalence.

Given $d|n$, according to definition 14 we have

$$|B(d)| = \phi(\frac{n}{d})$$

So, by definition of f:

$$n = \sum_{d \in I} \phi(\frac{n}{d}) \tag{18}$$
$$= \sum_{d \in I} \phi(f(d)) \tag{19}$$

Two more lemmas `I_inj_prop` and `f_image_I_eq_I` prove, respectively:

$$f \text{ is injective on } I \tag{20}$$
$$f(I) = I \tag{21}$$

Now 19 becomes

$$\begin{aligned} n &= \sum_{d\in f(I)} \phi(d) \quad \text{by property 20} \\ &= \sum_{d\in I} \phi(d) \quad \text{by property 21} \\ &= \sum_{d|n} \phi(d) \end{aligned}$$

that is the thesis of the theorem.

7. CONCLUSIONS

In this paper we described the formalization, in the Matita interactive theorem prover, of a few elementary results in number theory about the Möbius μ function, the Inversion Theorem, and the Euler ϕ function. Although our work and that of Avigad et al. [6] are the only published formalisations of the mathematics in question, some of the problems and of the techniques used for their solutions frequently occur in different mathematical domains. For instance, Dirichlet convolution is meant to equip a certain function space with ring properties, and has strong similarities with polynomial multiplication

$$p \cdot q = (n \mapsto \sum_{k=0}^{n} p_k \cdot q_{n-k})$$

where the computation of the $n-th$ coefficient requires a sum of addends indexed over all possible pairs (k, j) such that $k + j = n$.
Formal proofs of the convolution of polynomials can e.g. be found in [14] or [7]. The latter proof is by a brute force induction; details of the former proof are not given in the paper, but the following remark confirms the complexity of issue, independently from the underlying proof assistant in use:

> *"We would like to remark that proving the associativity of this convolution product presented a technical challenge as it turned out to be extremely tedious for the authors of [13]".*

Coming to the resources required for our work, it is clear that the amount of time invested largely depends by the skill of the authors; moreover, most of the time is usually spent not on the results themselves, but in integrating the library with the needed lemmas or, even worse, revising, cleaning or generalizing already proved statements (hence, with no clear trace, in size, of the work that was done). So, any measure must be taken with the due caution. As a really rough estimation, our whole contribution to the Matita library was likely to be around 4-5000 lines of script code, for an effort in time of about 150-200 hours. That makes 2-3 minutes per (script) line, and an average of 250-300 minutes for each line of the source mathematical text. The former datum is particularly impressive, and it is obviously the main obstacle towards a larger diffusion of automatic provers in the mathematical community. Most of the research effort in the area of formalized reasoning is finally aimed to reduce this cost; hence, inspite of the fact that its measure is clearly very sensible to the nature of the mathematical text being formalized, this value provides an interesting estimate of the state of the art.

Similarly, it is not so easy to compare the dimension of our development with that in [6] due to the different organization of the subjects. This is a tentative reconstruction:

Content	Isabelle	Matita
Möbius lemma	Mu 526 lines	moebius.ma 365 lines
Dirichlet convolution	Theory Inversion (up to `mu_inversion_nat1`) 714 lines	dirichlet_product.ma 520 lines
Inversion theorem		inversion.ma 100 lines
Euler ϕ lemma	EulerPhi (up to `big_prop1`) 389 lines	totient1.ma 241 lines
Total	1629 lines	1226 lines

Fig. 1. Matita vs. Isabelle

All "reindexing" operations for sums are contained in the file `iter_p_gen`, which is 1632 lines long, but this should be compared with the 2351 lines of Isabelle's "Finite Sets" Theory (integrated by 200 additional lines of Avigad's "FiniteLib" Theory).

The above comparison is not meant to draw conclusions about the relative efficiency of the underlying proof assistants, but only to emphasize the substantial similarity of the two developments, that in turns presumably reflects the overall amount of work required for writing them. At the current state of the art, the complexity of formalizing the results discussed in this paper is not likely to be sensibly reduced by any of the already available tools.

More interesting is the comparison of the formal proof with the original mathematical version. Ireland and Rosen [10] need precisely one page (36 lines) to present all results discussed in this paper. This gives a "De Bruijn" factor of 34 for Matita (45 for Isabelle). Even admitting that the page we considered is unusually dense, the increase in the length of the formal version is much higher than the value of 3-4, that looks typical of other mathematical subjects (see Wiedijk's analysis). Growing factors much higher than usual have been also testified in other parts of the formal proof of the prime number theorem in Isabelle:

> *"In many cases, the increase in length is dramatic: the three and a half pages of text associated with the proof of the error estimate translate to about 1600 lines, or 37 pages, of proof scripts and the five pages of text associated with the final part of the proof translate to about 4000 lines, or 89 pages of proof scripts."* [6]

Even recognizing the extreme conciseness of mathematics, we should not make the mistake to consider it as an *intrinsic* quality of its language. Since the age of 6, humans are trained 5/6 hours a week to learn the idiom of mathematics and its methods. This training is, for most people, quite hard, and not *natural* at all. The compactness of the mathematical notation and the flashing effectiveness of its arguments must be attributed, first of all, to the interpretative ability of the human mind and the suitable, intensive training he received (possibly joined, in a few cases, with some geometric intuition). It is well known that imitating the intellectual capabilities of humans is one of the most difficult tasks for computing machines, hence it is not evident at all that the traditional mathematical language and its

explanatory style, explicitly meant for inter-human communication, should also turn out to be immediately suitable or reusable for human-computer communication or, even worse, for purely automatic elaboration.

It is often heard that mathematicians would not use interactive provers for the reason that the language and techniques adopted but the current tools are too far from their working practice. However, computer scientists normally talk with machines in artificial languages explicitly devised to this aim, and quite far from natural language, or any other scientific idiom. Surely, it is much easier for a human to learn the commands of an interactive prover, than for a machine the cryptic arguments of mathematics. In fact, the large majority of mathematicians are not attracted by mathematical provers since the huge cost of the formalization is not counterbalanced by the additional features (comprising automatic checking) offered by the current tools. If we wish to convince mathematicians of the interest of tools supporting the automation of formal reasoning, the only possibility it to produce, soon or later, a genuinely *new* result. The flyspeck project, aimed to formalize the proof of the Kepler Conjecture [8, 12], partly goes in this direction (even if the proof has not been *devised* with the help of the computer, that would be the crucial step). A more close-to-hand goal would consist to give a new, original *proof* of a known result, or to find a new, unexpected *relation* between different notions. From this point of view, the weaknesses of the current systems could even turn out to be an unexpected feature, forcing the user to look for workarounds, or more elementary arguments. As remarked in [6]:

> *The need to rewrite proofs in such a way may be frustrating, but the task can also be oddly enjoyable: it poses interesting puzzles, and enables one to better understand the relationship of the advanced mathematical methods to the elementary substitutes.*

The authors arrive to regret that, with the progress of the systems and their libraries this re-founding work will gradually be less demanding, losing *"one good reason for investing time in such exercises."*

Actually, it is only desirable that interactive provers will provide in the future more and more functionalities to support that work of re-invention of mathematics that is the real novelty of the formal approach.

References

[1] T.M.Apostol. Introduction to Analytic Number Theory. Springer Verlag, 1976.

[2] A.Asperti, C.Sacerdoti Coen, E.Tassi, S.Zacchiroli. Crafting a Proof Assistant. Proceedings of Types 2006: Types for Proofs and Programs. Nottingham, UK, April 18-21, 2006. LNCS 4502, pp. 18-32, 2007. Springer Verlag.

[3] A.Asperti, C.Sacerdoti Coen, E.Tassi, S.Zacchiroli. User Interaction with the Matita Proof Assistant. Journal of Automated Reasoning, Special Issue on User Interfaces for Theorem Proving, V.39, N.2, August 2007. Springer Verlag.

[4] A.Asperti, F.Guidi, C.Sacerdoti Coen, E.Tassi, S.Zacchiroli. A Content Based Mathematical Search Engine: Whelp. Proceedings of TYPES 2004 conference: Types for Proofs and Programs. Paris, France, December 15-18, 2004. LNCS 3839, Springer Berlin.

[5] J.Avigad. Notes on a formalization of the prime number theorem. Carnegie Mellon Technical Report CMU-PHIL-163, 2004.

[6] J.Avigad, K.Donnelly, D.Gray, P.Raff. A formally verified proof of the prime number theorem. To appear in the ACM Transactions on Computational Logic.

[7] C.Ballarin. Computer Algebra and Theorem Proving. PhD Thesis. University of Cambridge, Computer Laboratory Technical Report n.473, 1999.

[8] T.C.Hales. Formalizing the Proof of the Kepler Conjecture. Proceedings of the 17th International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2004), Park City, Utah, USA, September 14-17, 2004. K.Slind, A.Bunker, G.Gopalakrishnan (Eds.) LNCS 3223 Springer 2004.

[9] G.H.Hardy, E.M.Wright. An Introduction to the Theory of Numbers. Oxford at the Clarendon Press, 1975 (4th edition).

[10] K.Ireland, M.Rosen. A Classical Introduction to Modern Number Theory. Graduate Texts in Mathematics. Springer Verlag.

[11] G.J.O. Jameson. The Prime Number Theorem. London Mathematical Society. Student Texts 53. Cambridge University Press. 2003.

[12] D.Mackenzie. What in the Name of Euclid Is Going On Here? Science, Vol. 307. no. 5714, pp. 1402 - 1403. March 2005.

[13] P.Rudnicki, A.Trybulec. Multivariate Polynomials with Arbitrary Number of Variables. Formalized Mathematics, 8, 317-332, 1999.

[14] P.Rudnicki, C.Schwarzweller, A.Trybulec. Commutative Algebra in the Mizar System, Journal of Symbolic Computation, vol. 32, pp. 143-169, 2001.

A. IRELAND AND ROSEN'S SOURCE PAGES

Proposition 2.2.2. *If n is a positive integer, let $n = p_1^{a_1}p_2^{a_2}\cdots p_l^{a_l}$ be its prime decomposition. Then*

(a) $\nu(n) = (a_1 + 1)(a_2 + 1)\cdots(a_l + 1)$.
(b) $\sigma(n) = ((p_1^{a_1+1} - 1)/(p_1 - 1))((p_2^{a_2+1} - 1)/(p_2 - 1))\cdots$
$((p_l^{a_l+1} - 1)/(p_l - 1))$.

PROOF. To prove part (a) notice that $m|n$ iff $m = p_1^{b_1}p_2^{b_2}\cdots p_l^{b_l}$ and $0 \le b_i \le a_i$ for $i = 1, 2, \ldots, l$. Thus the positive divisors of n are one-to-one correspondence with the n-tuples $(b_1, b_2, \ldots, b_l)$ with $0 \le b_i \le a_i$ for $i = 1, \ldots, l$, and there are exactly $(a_1 + 1)(a_2 + 1)\cdots(a_l + 1)$ such n-tuples.

To prove part (b) notice that $\sigma(n) = \sum p_1^{b_1}p_2^{b_2}\cdots p_l^{b_l}$, where the sum is over the above set of n-tuples. Thus, $\sigma(n) = (\sum_{b_1=0}^{a_1} p_1^{b_1})(\sum_{b_2=0}^{a_2} p_2^{b_2})\cdots(\sum_{b_l=0}^{a_l} p_l^{b_l})$, from which the result follows by use of the summation formula for the geometric series. □

There is an interesting and unsolved problem connected with the function $\sigma(n)$. A number n is said to be perfect if $\sigma(n) = 2n$. For example, 6 and 28 are perfect. In general, if $2^{m+1} - 1$ is a prime, then $n = 2^m(2^{m+1} - 1)$ is perfect, as can be seen by applying part (b) of Proposition 2.2.2. This fact is already in Euclid. L. Euler showed that any even perfect number has this form. Thus the problem of even perfect numbers is reduced to that of finding primes of the form $2^{m+1} - 1$. Such primes are called Mersenne primes. The two outstanding problems involving perfect numbers are the following: Are there infinitely many perfect numbers? Are there any odd perfect numbers?

The multiplicative analog of this problem is trivial. An integer n is called multiplicatively perfect if the product of the positive divisors of n is n^2. Such a number cannot be a prime or a square of a prime. Thus there is a proper divisor d such that $d \ne n/d$. The product of the divisors 1, d, n/d, and n is already n^2. Thus n is multiplicatively perfect iff there are exactly two proper divisors. The only such numbers are cubes of primes or products of two distinct primes. For example, 27 and 10 are multiplicatively perfect.

We now introduce a very important arithmetic function, the Möbius μ function. For $n \in \mathbb{Z}^+$, $\mu(1) = 1$, $\mu(n) = 0$ if n is not square-free, and $\mu(p_1p_2\cdots p_l) = (-1)^l$, where the p_i are distinct positive primes.

Proposition 2.2.3. *If $n > 1$, $\sum_{d|n} \mu(d) = 0$.*

PROOF. If $n = p_1^{a_1}p_2^{a_2}\cdots p_l^{a_l}$, then $\sum_{d|n} \mu(d) = \sum_{(\varepsilon_1,\ldots,\varepsilon_l)} \mu(p_1^{\varepsilon_1}\cdots p_l^{\varepsilon_l})$, where the ε_i are zero or 1. Thus

$$\sum_{d|n} \mu(d) = 1 - l + \binom{l}{2} - \binom{l}{3} + \cdots + (-1)^l = (1 - 1)^l = 0. \qquad \square$$

The full significance of the Möbius μ function can be understood most clearly when its connection with Dirichlet multiplication is brought to light.

Let f and g be complex valued functions on $\mathbb{Z}^+$. The Dirichlet product of f and g is defined by the formula $f \circ g(n) = \sum f(d_1)g(d_2)$, where the sum is over all pairs (d_1, d_2) of positive integers such that $d_1 d_2 = n$. This product is associative, as one can see by checking that $f \circ (g \circ h)(n) = (f \circ g) \circ h(n) = \sum f(d_1)g(d_2)h(d_3)$, where the sum is over all 3-tuples (d_1, d_2, d_3) of positive integers such that $d_1 d_2 d_3 = n$.

Define the function $\mathbf{1}$ by $\mathbf{1}(1) = 1$ and $\mathbf{1}(n) = 0$ for $n > 1$. Then $f \circ \mathbf{1} = \mathbf{1} \circ f = f$. Define I by $I(n) = 1$ for all $n \in \mathbb{Z}^+$. Then $f \circ I(n) = I \circ f(n) = \sum_{d|n} f(d)$.

Lemma. $I \circ \mu = \mu \circ I = \mathbf{1}$.

PROOF. $\mu \circ I(1) = \mu(1)I(1) = 1$. If $n > 1$, $\mu \circ I(n) = \sum_{d|n} \mu(d) = 0$. The same proof works for $I \circ \mu$. □

Theorem 2 (Möbius Inversion Theorem). *Let* $F(n) = \sum_{d|n} f(d)$. *Then* $f(n) = \sum_{d|n} \mu(d)F(n/d)$.

PROOF. $F = f \circ I$. Thus $F \circ \mu = (f \circ I) \circ \mu = f \circ (I \circ \mu) = f \circ \mathbf{1} = f$. This shows that $f(n) = F \circ \mu(n) = \sum_{d|n} \mu(d)F(n/d)$. □

Remark. We have considered complex-valued functions on the positive integers. It is useful to notice that Theorem 2 is valid whenever the functions take their value in an abelian group. The proof goes through word for word.

If the group law in the abelian group is written multiplicatively, the theorem takes the following form: If $F(n) = \prod_{d|n} f(d)$, then $f(n) = \prod_{d|n} F(n/d)^{\mu(d)}$.

The Möbius inversion theorem has many applications. We shall use it to obtain a formula for yet another arithmetic function, the Euler ϕ function. For $n \in \mathbb{Z}^+$, $\phi(n)$ is defined to be the number of integers between 1 and n relatively prime to n. For example, $\phi(1) = 1$, $\phi(5) = 4$, $\phi(6) = 2$, and $\phi(9) = 6$. If p is a prime, it is clear that $\phi(p) = p - 1$.

Proposition 2.2.4. $\sum_{d|n} \phi(d) = n$.

PROOF. Consider the n rational numbers $1/n$, $2/n$, $3/n$, ..., $(n-1)/n$, n/n. Reduce each to lowest terms; i.e., express each number as a quotient of relatively prime integers. The denominators will all be divisors of n. If $d|n$, exactly $\phi(d)$ of our numbers will have d in the denominator after reducing to lowest terms. Thus $\sum_{d|n} \phi(d) = n$. □

Proposition 2.2.5. *If* $n = p_1^{a_1} p_2^{a_2} \cdots p_t^{a_t}$, *then*

$$\phi(n) = n(1 - (1/p_1))(1 - (1/p_2)) \cdots (1 - (1/p_t)).$$

PROOF. Since $n = \sum_{d|n} \phi(d)$ the Möbius inversion theorem implies that $\phi(n) = \sum_{d|n} \mu(d)n/d = n - \sum_i n/p_i + \sum_{i<j} n/p_i p_j \cdots = n(1 - (1/p_1))(1 - (1/p_2)) \cdots (1 - (1/p_t))$. □

B. PROOF OF THE MÖBIUS INVERSION THEOREM

```
set "baseuri" "cic:/matita/Z/inversion".

include "Z/dirichlet_product.ma".
include "Z/moebius.ma".

(* moebius inversion theorem *)
theorem inversion: \forall f:nat \to Z.\forall n:nat.O < n \to
dirichlet_product moebius (sigma_div f) n = f n.
intros.
rewrite > commutative_dirichlet_product
  [apply (trans_eq ? ? (dirichlet_product (dirichlet_product f (\lambda n.Zone)) moebius n))
    [unfold dirichlet_product.
     apply eq_sigma_p1;intros
      [reflexivity
      |apply eq_f2
        [apply sym_eq.
         unfold sigma_div.
         apply dirichlet_product_one_r.
         apply (divides_b_true_to_lt_O ? ? H H2)
        |reflexivity
        ]
      ]
    |rewrite > associative_dirichlet_product
      [apply (trans_eq ? ? (dirichlet_product f (sigma_div moebius) n))
        [unfold dirichlet_product.
         apply eq_sigma_p1;intros
          [reflexivity
          |apply eq_f2
            [reflexivity
            |unfold sigma_div.
             apply dirichlet_product_one_l.
             apply (lt_times_n_to_lt x)
              [apply (divides_b_true_to_lt_O ? ? H H2)
              |rewrite > divides_to_div
                [assumption
                |apply (divides_b_true_to_divides ? ? H2)
                ]
              ]
            ]
          ]
        |apply (trans_eq ? ? (dirichlet_product f is_one n))
          [unfold dirichlet_product.
           apply eq_sigma_p1;intros
            [reflexivity
            |apply eq_f2
              [reflexivity
              |apply sigma_div_moebius.
               apply (lt_times_n_to_lt x)
                [apply (divides_b_true_to_lt_O ? ? H H2)
                |rewrite > divides_to_div
                  [assumption
                  |apply (divides_b_true_to_divides ? ? H2)
                  ]
                ]
              ]
            ]
          |apply dirichlet_product_is_one_r
          ]
        ]
      |assumption
      ]
    ]
  |assumption
  ]
qed.
```

C. SOME USEFUL LEMMAS

This appendix contains the list of *all* lemmas used in our development concerning properties of division, modulus, gcd, order and divide. The interest of such a listing is to give a more precise idea of the degree of granularity of the formal proof. The list of lemmas has been obtained automatically using the indexing functionalities provided by Matita (see [4]).

C.1 division and modulus

`le_div`	$\forall n,m.O < n \to m/n \leq m$
`div_n_n`	$\forall n.O < n \to n/n = 1$
`div_mod`	$\forall n,m.O < m \to n = (n/m) * m + (n \; mod \; m)$
`div_plus_times`	$\forall m,q,r : nat.r < m \to (q * m + r)/m = q$

C.2 divides

`divides_n_n`	$\forall n.n \mid n$
`divides_to_mod_O`	$\forall n,m.O < n \to n \mid m \to (m \; mod \; n) = O$
`trans_divides`	$\forall n,m,p.n \mid m \to m \mid p \to n \mid p$
`divides_to_le`	$\forall n,m.O < m \to n \mid m \to n \leq m$
`divides_to_lt_O`	$\forall n,m.O < m \to n \mid m \to O < n$
`divides_to_div`	$\forall n,m.n \mid m \to m/n * n = m$
`div_div`	$\forall n,d : nat.O < n \to d \mid n \to n/(n/d) = d$
`eq_times_div_div_times`	$\forall a,b,c : nat.O < b \to c \mid b \to a * (b/c) = (a * b)/c$

C.3 divides_b

`divides` is a binary property, while `divides_b` is its (computable) characteristic function. We need a few lemmas relating them:

`divides_b_true_to_divides`
$\forall n,m.divides\_b \; n \; m = true \to n \mid m$

`divides_b_false_to_not_divides`
$\forall n,m.divides\_b \; n \; m = false \to n \nmid m$

`divides_to_divides_b_true`
$\forall n,m.O < n \to n \mid m \to divides\_b \; n \; m = true$

`not_divides_to_divides_b_false`
$\forall n,m : nat.O < n \to n \nmid m \to divides\_b \; n \; m = false$

`divides_b_true_to_lt_O`
$\forall n,m.O < n \to divides\_b \; m \; n = true \to O < m$

`divides_b_div_true`
$\forall d,n.O < n \to divides\_b \; d \; n = true \to divides\_b \; (n/d) \; n = true$

C.4 ord

Let us recall that `(p_ord n p)` returns a pair $\langle q, r \rangle$ such that $n = p^q r$ and p does not divide r. The functions `ord` and `ord_rem` are respectively the first and second projection of `p_ord`.

`p_ord_exp1`
$\forall p, n, q, r.O < p \to p \not| r \to n = p^q * r \to p\_ord\ n\ p = \langle q, r \rangle$
`not_divides_to_p_ord_0`
$\forall n, i.(nth\_prime\ i) \not| n \to p\_ord\ n\ (nth\_prime\ i) = \langle O, n \rangle$
`p_ord_to_not_eq_0`
$\forall n, i, q, r.1 < n \to p\_ord\ n\ (nth\_prime\ i) = \langle q, r \rangle \to r \neq O$
`divides_to_le_ord`
$\forall p, n, m : nat.O < n \to O < m \to prime\ p \to n | m \to ord\ n\ p \leq ord\ m\ p$
`exp_ord`
$\forall p, n.1 < p \to O < n \to n = p^{(}ord\ n\ p) * (ord\_rem\ n\ p)$
`lt_0_ord_rem`
$\forall p, n.1 < p \to O < n \to O < ord\_rem\ n\ p$

C.5 max prime factor

The `max_prime_factor` function has been abbreviated here as `mpf` for editorial reasons.

`divides_mpf_n`
$\forall n.1 < n \to nth\_prime(mpf\ n) | n$
`divides_to_mpf1`
$\forall n, m.O < n \to O < m \to n | m \to mpf\ n \leq mpf\ m$
`p_ord_to_lt_mpf1`
$\forall n, p, q, r.O < n \to$
$mpf\ n \leq i \to \langle q, r \rangle = p\_ord\ n\ (nth\_prime\ i) \to 1 < r \to mpf\ r < i$
`lt_mpf_to_not_divides`
$\forall n, i.O < n \to n = 1 \vee mpf\ n < i \to (nth\_prime\ i \not| n)$
`eq_p_mpf`
$\forall n, p, r.O < n \to O < r \to$
$r = 1 \vee (mpf\ r) < i \to i = mpf(r * (nth\_prime\ i)^n)$

C.6 greatest common divisor (gcd)

`divides_gcd_n`
$\forall n, m.gcd\ n\ m | n$
`divides_gcd_m`
$\forall n, m.gcd\ n\ m | m$
`sym_gcd`
$\forall n, m : nat.gcd\ n\ m = gcd\ m\ n$
`eq_gcd_times_times_times_gcd`
$\forall a, b, c : nat.(gcd\ (c * a)\ (c * b)) = c * (gcd\ a\ b)$
`eq_gcd_div_div_div_gcd`
$\forall a, b, m : nat.O < m \to m | a \to m | b \to gcd\ (a/m)\ (b/m) = (gcd\ a\ b)/m$
`divides_times_to_divides_div_gcd`
$\forall a, b, c : nat.a | (b * c) \to (a/(gcd\ a\ b)) | c$