The Perfect Number Theorem

Mark IJbema

August 16, 2018

Abstract

This document presents the formal proof of the Perfect Number Theorem. The result can also be found as number 70 on the list of “top 100 mathematical theorems” [Wie]. This document was produced as result of a B.Sc. Thesis under supervision of Jaap Top and Wim H. Hesselink (University of Groningen) in 2009.

Contents

1 Basics needed 1
2 Sum of divisors function 3
3 Perfect Number Theorem 7

1 Basics needed

theory PerfectBasics
imports Main HOL–Computational-Algebra.Primes HOL–Algebra.Exponent
begin

lemma sum-mono2-nat: finite (B::nat set) ==> A < B ==> \sum A < \sum B
by (auto simp add: sum-mono2)

lemma multiplicity-0 [simp]: multiplicity 0 x = 0
by (cases x = 0) (auto intro: not-dvd-imp-multiplicity-0)

lemma exp-is-max-div:
  assumes m0: m \neq 0 and p: prime p
  shows ~ p dvd (m div (p ^ (multiplicity p m)))
proof (rule contr)
  assume ~ ~ p dvd (m div (p ^ (multiplicity p m)))
  hence a:p dvd (m div (p ^ (multiplicity p m))) by auto
  from m0 have p ^ (multiplicity p m) dvd m by (auto simp add: multiplicity-dvd)
lemma coprime-multiplicity:
  assumes prime (p::nat) and m > 0
  shows coprime p (m div (p ^ multiplicity p m))
proof (rule ccontr)
  assume ¬ coprime p (m div p ^ multiplicity p m)
  with (prime p) have ∃ q. prime q ∧ q dvd p ∧ q dvd m div p ^ multiplicity p m
    by (metis dvd-refl prime-imp-coprime)
  with (prime p) have ∃ q. q = p ∧ q dvd m div p ^ multiplicity p m
    by (metis not-prime-1 prime-nat-iff)
  then have p dvd m div p ^ multiplicity p m
    by auto
  with assms show False
    by (auto simp add: exp-is-max-div)
qed

lemma add-mul-distrib-three: (x::nat)*(a+b+c) = x*a + x*b + x*c
proof
  have (x::nat)*(a+b+c) = x*((a+b)+c) by auto
  hence x*(a+b+c) = x*(a+b)+x*c by (simp add: algebra-simps)
  thus x*(a+b+c) = x*a + x*b + x*c by (simp add: algebra-simps)
qed

lemma nat-interval-minus-zero: {0..Suc n} = {0} Un {Suc 0..Suc n} by auto
lemma nat-interval-minus-zero2:
  assumes n>0
  shows {0..n} = {0} Un {Suc 0..n} by (auto simp add: nat-interval-minus-zero)

theorem simplify-sum-of-powers: (x - 1::nat) * (∑ i=0..n . x ^ i) = x ^ (n + 1) - 1 (is ?l = ?r)
proof (cases)
  assume n = 0
  thus ?l = x ^ (n + 1) - 1 by auto
next
  assume n ^ = 0
  hence n0: n>0 by auto
  have ?l = (x::nat) * (∑ i=0..n . x ^ i) - (∑ i=0..n . x ^ i)
    by (metis diff-mult-distrib nat-mult-1)
  also have ... = (∑ i=0..n . x ^ (Suc i)) - (∑ i=0..n . x ^ i)
    by (simp add: sum-distrib-left)
  also have ... = (∑ i=Suc 0..Suc n . x ^ i) - (∑ i=0..n . x ^ i)
    by (metis sum-shift-bounds-cl-Suc-ivl)
  also with n0
  have ... = ((∑ i=Suc 0..n . x ^ i) + x ^ (Suc n)) - (x ^ 0 + (∑ i=Suc 0..n . x ^ i))
by \(\text{auto simp add: sum.union-disjoint nat-interval-minus-zero2}\)

finally show \(?thesis by auto\)

qed

end

2 Sum of divisors function

theory Sigma

imports PerfectBasics HOL\Library.Infinite-Set

begin

definition divisors :: nat \Rightarrow nat set where
divisors \((m::nat)\) == \{\(n \mid n \text{ dvd } m\}\)

definition sigma :: nat \Rightarrow nat where
sigma \(m\) == \(\sum n \mid n \text{ dvd } m \cdot n\)

lemma sigma-divisors: sigma\((n)\) = \(\sum (\text{divisors}(n))\)
by (auto simp: sigma-def divisors-def)

lemma divisors-eq-dvd[iff]: \((a::\text{divisors}(n)) = (a \text{ dvd } n)\)
by (simp add: divisors-def)

lemma mult-divisors: \((a::nat)\cdot b = c\Rightarrow a \text{ divisors } c\)
by (unfold divisors-def dvd-def, blast)

lemma mult-divisors2: \((a::nat)\cdot b = c\Rightarrow b \text{ divisors } c\)
by (unfold divisors-def dvd-def, auto)

lemma divisorsfinite[simp]:
assumes \(n > 0\)
shows \(\text{finite } (\text{divisors } n)\)
proof -
  from assms have \(\text{divisors } n = \{m \mid m \text{ dvd } n \& m \leq n\}\)
  by (auto simp only: divisors-def dvd-imp-le)
  hence \(\text{divisors } n \leq \{m \mid m \leq n\}\) by auto
  thus \(\text{finite } (\text{divisors } n)\)
  by (metis finite-Collect-le-nat finite-subset)
qed

lemma divs-of-zero-UNIV[simp]: \(\text{divisors}(0) = \text{UNIV}\)
by (auto simp add: divisors-def)

lemma sigma0[simp]: \(\sigma(0) = 0\)
by (simp add: sigma-def)

lemma sigma1[simp]: \(\sigma(1) = 1\)
by (simp add: sigma-def)

lemma prime-divisors: prime \((p::nat)\) \(\iff\) \(\text{divisors } p = \{1,p\} \& p > 1\)
by (auto simp add: divisors-def prime-nat-iff)

lemma prime-imp-sigma: prime (p::nat) ==> sigma(p) = p+1
proof
  assume prime (p::nat)
  hence p>1 ∧ divisors(p) = \{1,p\} by (simp add: prime-divisors)
  hence p>1 ∧ sigma(p) = ∑ \{1,p\} by (auto simp only: sigma-divisors divisors-def)
  thus sigma(p) = p+1 by simp
qed

lemma sigma-third-divisor:
  assumes 1 < a a < n a : divisors n
  shows 1+a+n <= sigma(n)
proof
  from assms have finite {1,a,n} & finite (divisors n) & {1,a,n} <= divisors n
  by auto
  hence ∑ {1,a,n} <= ∑ (divisors n) by (simp only: sum-mono2)
  hence ∑ {1,a,n} <= sigma n by (simp add: sigma-divisors)
  with assms show ?thesis by auto
qed

lemma sigma-imp-divisors: sigma(n)=n+1 ==> n>1 & divisors n = \{n,1\}
proof
  assume ass: sigma(n)=n+1
  hence n\#0 & n\#1
    by (metis Suc-eq-plus1 n-not-Suc-n sigma0 sigma1)
  thus concl: n>1 by simp

show divisors n = \{n,1\}
proof (rule ccontr)
  assume divisors n ≠ \{n,1\}
  with concl have divisors n ≠ \{n,1\} & 1<n by auto
  moreover
  from ass concl have 1 : divisors(n) & n : divisors n & ~0 : divisors n
    by (simp add: dvd-def divisors-def)
  ultimately
  have (∃ a. a\#n & a\#1 & 1<n & a : divisors n) & 0 ~: divisors n by auto
  hence (∃ a. a\#n & a\#1 & 1<n & a\#0 & a : divisors n) by metis
  hence ∃ a . a\#n & a\#1 & 1\#n & a\#0 & finite {1,a,n} & finite (divisors n)
  & {1,a,n} <= divisors n by auto
  hence ∃ a. a\#n & a\#1 & 1\#n & a\#0 & ∑ {1,a,n} <= sigma n
    by (metis sum-mono2-nat sigma-divisors)
  hence ∃ a. a\#0 & (1+a+n) <= sigma n by auto
  hence 1+n<sigma n by auto
  with ass show False by auto
qed
qed
lemma sigma-imp-prime: \( \sigma(n) = n+1 \implies \text{prime } n \)
proof -
  assume \( \alpha: \sigma(n) = n+1 \)
  hence \( n > 1 \) \& \( \text{divisors}(n) = \{1, n\} \) by (metis insert-commute sigma-imp-divisors)
  thus \( \text{prime } n \) by (simp add: prime-divisors)
qed

lemma pr-pow-div-eq-sm-pr-pow:
fixes \( p \) :: nat
assumes \( \text{prime } p \)
shows \( \{ d : d \mid d \vdots p^m \} = \{ p^f : f < n \} \)
proof
  show \( \{ p^f : f < n \} \subseteq \{ d : d \mid d \vdots p^n \} \)
  proof
    fix \( x \)
    assume \( x : \{ p^f : f < n \} \)
    hence \( \exists i. x = p^i \) \& \( i < n \) by auto
    with \( \text{prime } p \) have \( x \vdots p^n \)
      by (metis \text{le-imp-power-dvd})
    thus \( x : \{ d : d \mid d \vdots p^n \} \) by auto
  qed
next
  show \( \{ d : d \mid d \vdots p^n \} \subseteq \{ p^f : f < n \} \)
  proof
    fix \( x \)
    assume \( x : \{ d : d \mid d \vdots p^n \} \)
    hence \( x \vdots p^n \) by auto
    with \( \text{prime } p \) obtain \( i \) where \( i < n \) \& \( x = p^i \) using \text{prime-dvd-power-nat-iff}
    prime-dvd-power-nat
      by (auto simp only: \text{divides-primepow-nat})
    hence \( x = p^i \) \& \( i < n \) by auto
    thus \( x : \{ p^f : f < n \} \) by auto
  qed
qed

lemma rewrite-sum-of-powers:
assumes \( p : (p::nat) > 1 \)
shows \( \sum \{ p^m : m \mid m < (n::nat) \} \) = \( \sum \{ i = 0 \ldots n : p^i \} \) (is \( ?l = ?r \))
proof -
  have \( \forall l = \text{sum } (\%x. x) \{ (\_\_ ) p \} \) \( m \mid m < n \) by auto
  also have \( \ldots = \text{sum } (\%x. x) \{ (\_\_ ) p \} \{ m \mid m < n \} \)
    by (simp add: \text{setcompr-eq-image})
  moreover with \( p \) have \( \text{inj-on } (\_\_ ) p \) \( \{ m \mid m < n \} \)
    by (simp add: \text{inj-on-def})
  ultimately have \( \forall l = \text{sum } (\_\_ ) p \) \( m \mid m < n \)
    by (simp add: \text{sum.reindex})
  moreover have \( \{ m::nat : m < n \} = \{ 0 \ldots n \} \) by auto
  ultimately show \( \forall l = \sum \{ i = 0 \ldots n : p^i \} \) by auto
qed
theorem sigma-primepower:

\( \text{prime } p \implies (p - 1) \ast \text{sigma}(p^{(e::nat)}) = (p^{(e+1)} - 1) \)

proof –

- assume prime p
- hence \( \text{sigma}(p^{(e::nat)}) = \sum_{i=0}^{e \cdot p^i} \)
  by (simp add: pr-pow-div-eq-sm-pr-pow sigma-def rewrite-sum-of-powers prime-nat-iff)

thus \( (p - 1) \ast \text{sigma}(p^e) = p^{(e+1)} - 1 \) by (simp only: simplify-sum-of-powers)

qed

lemma sigma-prime-power-two: \( \text{sigma}(2^{(n::nat)}) = 2^{(n+1)} - 1 \)

proof –

- have \( (2 - 1) \ast \text{sigma}(2^{(n::nat)}) = 2^{(n+1)} - 1 \)
  by (auto simp only: sigma-primepower two-is-prime-nat)

thus \( ? \)thesis by simp

qed

lemma prodsums-eq-sumprods:

- fixes \( p::nat \text{ and } m::nat \)
- assumes coprime \( p \text{ and } m \)
- shows \( \sum \{ p^f \ast b \mid f,b \leq n \} \ast \sum \{ b \text{ dvd } m \} = \sum \{ p^f \ast b \mid f,b \leq n \text{ and } b \text{ dvd } m \} \)

proof –

- have coprime \( p \text{ and } m \) if \( x \text{ dvd } m \) for \( x \)
  using assms by (rule coprime-imp-coprime) (auto intro: dvd-trans that)

- then have coprime \( (p^f) \text{ and } m \) if \( x \text{ dvd } m \) for \( f \)
  using that by simp

- then show \( ? \)thesis
  by (auto simp: imp-ex sum-mult-sum-if-inj [OF mult-inj-if-coprime-nat] intro!: arg-cong [where \( f = \sum (\lambda x \cdot x) ]] )

qed

declare [[simproc add: finite-Collect]]

lemma rewrite-for-sigma-semmultiplicative:

- fixes \( p::nat \)
- assumes prime \( p \)
- shows \( \{ p^f \ast b \mid f,b \leq n \text{ and } b \text{ dvd } m \} = \{ a\ast b \mid a \text{ and } a \text{ dvd } p^{(n)} \text{ and } b \text{ dvd } m \} \)

proof

- show \( \{ p^f \ast b \mid f,b \leq n \text{ and } b \text{ dvd } m \} \ast \{ a\ast b \mid a \text{ and } a \text{ dvd } p \text{ and } n \text{ and } b \text{ dvd } m \} \)
  proof
    - fix \( x \)
      - assume \( x : \{ p^f \ast b \mid f,b \leq n \text{ and } b \text{ dvd } m \} \)
      - then obtain \( b \text{ and } f \text{ where } x = p^f \ast b \text{ and } f \leq n \text{ and } b \text{ dvd } m \) by auto
        with \( \text{prime } p \) show \( x : \{ a \ast b \mid a \text{ and } a \text{ dvd } p \text{ and } n \text{ and } b \text{ dvd } m \} \)
          by (auto simproc add: divides-primepow-nat)
  qed

next

- show \( \{ a\ast b \mid a \text{ and } a \text{ dvd } p \text{ and } n \text{ and } b \text{ dvd } m \} \leq \{ p^f \ast b \mid f,b \leq n \text{ and } b \text{ dvd } m \} \)

using ⟨prime p⟩ by auto (metis assms divides-primepow-nat)
qed

lemma div-decomp-comp:
  fixes a :: nat
  shows coprime m n ⇒ a dvd m * n ←→ (∃ b c. a = b * c & b dvd m & c dvd n)
  by (auto simp only: division-decomp mult-dvd-mono)

theorem sigma-semimultiplicative:
  assumes p: prime p and cop: coprime p m
  shows sigma (p^n) * sigma m = sigma (p^n * m) (is ?l = ?r)
  proof
    from cop have cop2: coprime (p^n) m by simp
    have ?l = (∑ {a . a dvd p^n}) * (∑ {b . b dvd m}) by (simp add: sigma-def)
    also from p have ... = (∑ {p^f | f. f<=n}) * (∑ {b . b dvd m})
      by (simp add: pr-pow-div-eq-sm-pr-pow)
    also from cop have ... = (∑ {p^f*b | f b. f<=n & b dvd m})
      by (auto simp add: prodsums-eq-sumprods prime-nat-iff)
    also have ... = (∑ {a*b | a b . a dvd (p^n) & b dvd m})
      by (simp add: p rewrite-for-sigma-semimultiplicative)
    finally have ?l = (∑ {c . c dvd (p^n*m)}) by (subst div-decomp-comp[OF cop2])
    thus ?l = sigma (p^n*m) by (auto simp add: sigma-def)
  qed

end

3 Perfect Number Theorem

theory Perfect
  imports Sigma
  begin

definition perfect :: nat => bool where
  perfect m == m>0 & 2*m = sigma m

theorem perfect-number-theorem:
  assumes even: even m and perfect: perfect m
  shows ∃ n. m = 2^n*(2^(n+1) - 1) ∧ prime ((2::nat)^((n+1) - 1))
  proof
    from perfect have m0: m>0 by (auto simp add: perfect-def)
    let ?n = multiplicity 2 m
    let ?A = m div 2^?n
    let ?np = (2::nat)^((?n+1) - 1)
    from even m0 have n1: ?n >= 1 by (simp add: multiplicity-geI)
have $2^\cdot n \div d \cdot m$ by (rule multiplicity-dvd)

hence $m = 2^\cdot n \cdot A$ by (simp only: dvd-mult-div-cancel)

with $m\theta$ have $\text{ndef: } m=2^\cdot n\cdot A \& \text{ coprime } 2 \cdot A$

using multiplicity-decompose [of $m \cdot 2$] by simp

moreover with $m\theta$ have $a\theta: A>0$ by (metis nat-0-less-mult-iff)

moreover

{ from perfect have $2\cdot m=\sigma(m)$ by (simp add: perfect-def)

  with $\text{ndef}$ have $2^\cdot (n+1)\cdot A=\sigma(2^\cdot n\cdot A)$ by auto

} ultimately have $2^\cdot (n+1)\cdot A=\sigma(2^\cdot n\cdot A)$

  by (simp add: sigma-semimultiplicative)

hence formula: $2^\cdot (n+1)\cdot A=(?np)*\sigma(?A)$

  by (simp only: sigma-prime-power-two)

from $n1$ have $(2::nat) \cdot (n+1) \geq 2^\cdot 2$ by (simp only: power-increasing)

hence $\text{mplarger: } ?np \geq 3$ by auto

let $B = ?A \div ?np$

from formula have $?np \div d \cdot ?A \cdot 2^\cdot (n+1)$

  by (auto simp add: ac-simps)

then have $?np \div d \cdot ?A$

  using coprime-diff-one-left-nat [of $2 \cdot (\text{multiplicity } 2 \cdot m + 1)$]

  by (auto simp add: coprime-dvd-mult-left-iff)

then have $b\text{ndef: } ?np \cdot B = ?A$

  by simp

with $a\theta$ have $b\theta: B>0$ by (metis gr0I mult-is-0)

from $\text{mplarger}$ $a\theta$ have $b\text{smallera: } B < ?A$ by auto

have $B = 1$

proof (rule ccontr)

  assume $\neg B = 1$

  with $b\theta \text{smallera}$ have $1 < ?B \cdot ?B < ?A$ by auto

  moreover from $b\text{ndef}$ have $?B : \text{ divisors } ?A$ by (rule mult-divisors2)

  ultimately have $1+?B+?A \leq \sigma ?A$ by (rule sigma-third-divisor)

  with $\text{mplarger}$ have $?np*(1+?A+?B) \leq ?np*(\sigma ?A)$

    by (auto simp only: nat-mult-le-cancel1)

  with $b\text{ndef}$ have $?np+B A \cdot ?A+1 \leq ?np*(\sigma ?A)$

    by (simp only: add-mult-distrib-three mult_commute)

  hence $?np+B A \cdot (?np + 1) \leq ?np*(\sigma ?A)$ by (simp only: add-mult-distrib2)

  with $\text{mplarger}$ have $2^\cdot (n+1)\cdot ?A < ?np*(\sigma ?A)$ by simp;auto

  with formula show False by auto

qed

with $b\text{ndef}$ have $a\text{ndef: } ?A=?np$ by auto

with formula have $?np\cdot 2^\cdot (n+1) = (?np)*\sigma(?A)$ by auto

with $\text{mplarger}$ $a\text{ndef}$ have $?A + 1=\sigma(?A)$ by auto

with $a\theta$ have $\text{prime } ?A$ by (simp add: sigma-imp-prime)

with $\text{ndef}$ $a\text{ndef}$ show $m = 2^\cdot n\cdot (?np)$ & $\text{prime } ?np$ by simp
theorem Euclid-book9-prop36:
  assumes p: prime (2^\(\mathit{n}+1\) - 1)
  shows perfect ((2^n)*(2^{\mathit{n}+1} - 1))
proof (unfold perfect-def, auto)
  from assms show (2^n)*2^n > Suc 0 by (auto simp add: prime-nat-iff)
next
  have 2 ~= ((2::nat)^{(n+1)} - 1) by simp arith
  then have coprime (2::nat) (2^{\mathit{n}+1} - 1)
    by (metis p primes-coprime-nat two-is-prime-nat)
  moreover with p have 2^{\mathit{n}+1} - 1 > (0::nat)
    by (auto simp add: prime-nat-iff)
  ultimately have sigma (2^n*(2^{\mathit{n}+1} - 1)) = (sigma(2^n))*(sigma(2^{\mathit{n}+1} - 1))
    by (metis sigma-semimultiplicative two-is-prime-nat)
  also from assms have ... = (sigma(2^n))*(2^{\mathit{n}+1})
    by (auto simp add: prime-imp-sigma)
  also have ... = (2^{\mathit{n}+1} - 1)*(sigma(2^{\mathit{n}+1})) by(simp add: sigma-prime-power-two)
  finally show 2*(2^n * (2*2^n - Suc 0)) = sigma(2^n*(2*2^n - Suc 0)) by auto
qed
end

References