# The Perfect Number Theorem 

Mark IJbema

September 13, 2023


#### 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 2
3 Perfect Number Theorem 6

## 1 Basics needed

```
theory PerfectBasics
imports Main HOL-Computational-Algebra.Primes HOL-Algebra.Exponent
begin
lemma exp-is-max-div:
    assumes \(m 0: m \neq 0\) and \(p\) : prime \(p\)
    shows \(\sim p \operatorname{dvd}\left(m \operatorname{div}\left(p^{\wedge}(\right.\right.\) multiplicity \(\left.\left.p m)\right)\right)\)
proof (rule ccontr)
    assume \(\sim \sim p d v d\left(m \operatorname{div}\left(p^{\wedge}(\right.\right.\) multiplicity \(\left.\left.p m)\right)\right)\)
    hence a:p dvd ( \(m\) div ( \(\left.p^{\wedge}(m u l t i p l i c i t y ~ p m)\right)\) ) by auto
    from \(m 0\) have \(p^{\wedge}\) (multiplicity \(\left.p m\right) d v d m\) by (auto simp add: multiplicity-dvd)
    with \(a\) have \(p^{\wedge}\) Suc (multiplicity \(p m\) ) dvd \(m\)
        by (subst (asm) dvd-div-iff-mult) auto
    with \(m 0 p\) show False
        by (subst (asm) power-dvd-iff-le-multiplicity) auto
qed
lemma coprime-multiplicity:
    assumes prime ( \(p:: n a t\) ) and \(m>0\)
```

```
    shows coprime p (m div ( }\mp@subsup{p}{}{`}\mathrm{ multiplicity p m))
proof (rule ccontr)
    assume }\neg\mathrm{ coprime p (m div p` multiplicity p m)
    with〈prime p> have \existsq. prime q}\wedgeq\mathrm{ dvd p}\wedgeq\mathrm{ dvd m div p^ multiplicity p m
    by (metis dvd-refl prime-imp-coprime)
    with <prime p> have \existsq. q=p\wedge q dvd m div p^ multiplicity p m
        by (metis not-prime-1 prime-nat-iff)
    then have p dvd m div p^ multiplicity pm
        by auto
    with assms show False
        by (auto simp add: exp-is-max-div)
qed
theorem simplify-sum-of-powers: (x-1::nat)*(\sumi=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\not=0
    hence n0: n>0 by auto
    have ?l = (x::nat )*(\sumi=0 .. n. x`i) - (\sumi=0 .. n. x`i)
        by (metis diff-mult-distrib nat-mult-1)
    also have ... = (\sumi=0 .. n. x` (Suc i)) - (\sumi=0 .. n . x`i)
    by (simp add: sum-distrib-left)
    also have ... = (\sumi=Suc 0 .. Suc n . x`i) - (\sumi=0 .. n . x`i)
    by (metis sum.shift-bounds-cl-Suc-ivl)
    also have ... = ((\sumi=Suc 0 .. n. x`i) +x`(Suc n)) - (x`0 + (\sumi=Suc 0 .. n.
x`i))
    by (simp add: sum.union-disjoint diff-add-inverse sum.atLeast-Suc-atMost)
    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 $m \equiv\{n . n d v d m\}$
abbreviation sigma $::$ nat $\Rightarrow$ nat where
sigma $m \equiv \sum(\operatorname{divisors}(m))$
lemma divisors-eq-dvd[iff]: $(a \in \operatorname{divisors}(n)) \longleftrightarrow(a$ dvd $n)$
by (simp add: divisors-def)

```
lemma finite-divisors [simp]:
    assumes n>0 shows finite (divisors n)
    by (simp add: assms divisors-def)
lemma divs-of-zero-UNIV[simp]: divisors(0) = UNIV
    by(auto simp add: divisors-def)
lemma sigma0[simp]: sigma(0) = 0
    by simp
lemma sigma1[simp]: sigma(Suc 0) = 1
    by (simp add: sum-eq-Suc0-iff)
lemma prime-divisors: prime p\longleftrightarrow divisors }p={1,p}\wedgep>
    by (auto simp add: divisors-def prime-nat-iff)
lemma prime-imp-sigma: prime (p::nat) \Longrightarrow sigma( }p)=p+
proof -
    assume prime (p::nat)
    hence p>1 ^ divisors(p)={1,p} by (simp add: prime-divisors)
    hence p>1^\operatorname{sigma}(p)=\sum{1,p} by (auto simp only: divisors-def)
    thus }\operatorname{sigma}(p)=p+1 by sim
qed
lemma sigma-third-divisor:
    assumes 1<a a<n a dvd n
    shows }1+a+n\leq\operatorname{sigma}(n
proof -
    from assms have {1,a,n}\leqdivisors n
        by auto
    hence \sum{1,a,n}\leqsigma n
        by (meson {a<n〉 finite-divisors order.strict-trans1 sum-mono2 zero-le)
    with assms show ?thesis by auto
qed
proposition prime-iff-sigma: prime n}\longleftrightarrow\operatorname{sigma}(n)=\mathrm{ Suc n
proof
    assume L: sigma(n)=Suc n
    then have n>1
        using less-linear sigma1 by fastforce
    moreover
    have m=Suc 0 if m dvd n m\not=n for m
    proof -
        have 0<m
            using that by auto
        then have }\neg1+m+n\leq1+
            by linarith
        then show ?thesis
```

using sigma-third-divisor [of m]
by (metis L One-nat-def Suc-lessD Suc-lessI $\langle n>1\rangle d v d-i m p-l e\langle 0<m\rangle$ less-le plus-1-eq-Suc that)
qed
then have divisors $n=\{n, 1\}$
by (auto simp: divisors-def)
ultimately show prime $n$
by (simp add: insert-commute prime-divisors)
qed (use prime-divisors in auto)
lemma dvd-prime-power-iff:
fixes $p:: n a t$
assumes prime: prime $p$
shows $\left\{d . d\right.$ dvd $\left.p^{\wedge} n\right\}=\left(\lambda k . p^{\wedge} k\right) ‘\{0 . . n\}$
using divides-primepow-nat prime by (auto simp add: le-imp-power-dvd)
lemma rewrite-sum-of-powers:
assumes $p:(p::$ nat $)>1$
shows $\sum\left((\mathcal{\wedge}) p^{\prime}\{0 . . n\}\right)=\left(\sum i=0 . . n \cdot p \uparrow i\right)($ is $? l=? r)$
by (metis inj-on-def p power-inject-exp sum.reindex-cong)
lemma sum-of-powers-int: $(x-1::$ int $) *\left(\sum i=0 . . n . x^{\wedge} i\right)=x^{\wedge}$ Suc $n-1$
by (metis atLeast0AtMost lessThan-Suc-atMost power-diff-1-eq)
lemma sum-of-powers-nat: $(x-1::$ nat $) *\left(\sum i=0 . . n \cdot x \widehat{\imath}\right)=x^{\wedge}$ Suc $n-1$
(is ? $l=? r$ )
proof (cases $x=0$ )
case False
then have $\operatorname{int}\left((x-1) * \operatorname{sum}\left(\left({ }^{\wedge}\right) x\right)\{0 . . n\}\right)=\operatorname{int}\left(x^{\wedge}\right.$ Suc $\left.n-1\right)$
using sum-of-powers-int [of int $x \quad n$ ] by (simp add: of-nat-diff)
then show ?thesis
using of-nat-eq-iff by blast
qed auto
theorem sigma-primepower:
assumes prime $p$
shows $(p-1) * \operatorname{sigma}\left(p^{\wedge} e\right)=p^{\wedge}(e+1)-1$
proof -
have $\operatorname{sigma}\left(p^{\wedge} e\right)=\left(\sum i=0 . . e \cdot p \widehat{i}\right)$
using assms divisors-def dvd-prime-power-iff prime-nat-iff rewrite-sum-of-powers
by auto
thus $(p-1) * \operatorname{sigma}\left(p^{\wedge} e\right)=p^{\wedge}(e+1)-1$
using sum-of-powers-nat by auto
qed

proof -
have $(2-1) * \operatorname{sigma}(2 \widehat{2})=$ 2^ $^{\wedge}(n+1)-1$

```
    by (auto simp only: sigma-primepower two-is-prime-nat)
    thus ?thesis by simp
qed
lemma prodsums-eq-sumprods:
    fixes p :: nat and m :: nat
    assumes coprime p m
    shows \sum((\lambdak. p^k)`{0..n})* sigma m=\sum{p^k*b|kb.k\leqn^b dvdm}
    (is ?lhs = ?rhs)
proof -
    have coprime px if x dvd m for x
    using assms by (rule coprime-imp-coprime) (auto intro:dvd-trans that)
    then have coprime ( }\mp@subsup{p}{}{`}f)x\mathrm{ if }xdvdm\mathrm{ for }x
        using that by simp
    then have ?lhs = \sum{a*b|ab. (\existsf.a= p^f^f\leqn)^bdvdm}
        apply (subst sum-mult-sum-if-inj [OF mult-inj-if-coprime-nat])
            apply (force intro!: sum.cong)+
        done
    also have ... = ?rhs
    by (blast intro: sum.cong)
    finally show ?thesis.
qed
lemma div-decomp-comp:
    fixes a::nat
    shows coprime m n\Longrightarrowa dvd m*n \longleftrightarrow(\existsbc. a=b*c\wedgeb dvd m}\wedgec\mathrm{ 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 ( }\mp@subsup{p}{`}{\prime}n)*\operatorname{sigma}m=\operatorname{sigma}(p`n*m)(is ?lhs = ?rhs
proof -
    from cop have cop2: coprime ( }p\widehat{`}\mathrm{ ) m
        by simp
```



```
        using divisors-def dvd-prime-power-iff by auto
    also from cop have ... =( \sum{p`f*b|fb.f\leqn^b dvd m})
        by (auto simp add: prodsums-eq-sumprods prime-nat-iff)
    also have ... = (\sum{a*b| ab.advd ( p^n)^bdvd m})
        by (metis (no-types, opaque-lifting) le-imp-power-dvd divides-primepow-nat p)
    also have ... = \sum{c.c dvd (p^n*m)}
        using cop2 div-decomp-comp by auto
    finally show ?thesis
        by (simp add: divisors-def)
qed
end
```


## 3 Perfect Number Theorem

```
theory Perfect
imports Sigma
begin
definition perfect :: nat \(=>\) bool where
perfect \(m \equiv m>0 \wedge 2 * m=\) sigma \(m\)
```

theorem perfect-number-theorem:
assumes even: even $m$ and perfect: perfect $m$
shows $\exists n \cdot m=2 ` n *\left(\mathcal{R}^{\wedge}(n+1)-1\right) \wedge$ prime $((2:: n a t) \uparrow(n+1)-1)$
proof
from perfect have $m 0: m>0$ by (auto simp add: perfect-def)
let $? n=$ multiplicity 2 m
let ? $A=m$ div 2^?n
let $? n p=(2:: n a t) \wedge(? n+1)-1$
from even $m 0$ have $n 1:$ ? $n>=1$ by (simp add: multiplicity-geI)
have $2 \uparrow ? n$ dvd $m$ by (rule multiplicity-dvd)
hence $m=2 \wedge ? n *$ ? A by (simp only: dvd-mult-div-cancel)
with $m 0$ have mdef: $m=\mathcal{Z}^{\wedge}$ ? $n *$ ? $A \wedge$ coprime 2 ? A
using multiplicity-decompose [of m 2] by simp
moreover with $m 0$ have $a 0$ : ? A>0 by (metis nat-0-less-mult-iff)
moreover
\{ from perfect have $2 * m=\operatorname{sigma}(m)$ by (simp add: perfect-def)
with mdef have $\mathcal{Z}^{\wedge}(? n+1) * ? A=\operatorname{sigma}\left(\mathscr{L}^{\wedge} ? n * ? A\right)$ by auto
\} ultimately have $2 \wedge(? n+1) * ? A=\operatorname{sigma}(2 \wedge ? n) * \operatorname{sigma}(? A)$
by (simp add: sigma-semimultiplicative)
hence formula: $\mathfrak{2}^{\wedge}(? n+1) * ? A=(? n p) * \operatorname{sigma}(? A)$
by (simp only: sigma-prime-power-two)
from $n 1$ have (2::nat) $)^{(? n+1)}>=$ 2 $^{2} 2$ by (simp only: power-increasing)
hence nplarger: ?np>=3 by auto
let $? B=? A$ div ? $n p$
from formula have ? $n p$ dvd ? $A * \mathcal{Z}^{\wedge}(? n+1)$
by (auto simp add: ac-simps)
then have ?np dvd?A
using coprime-diff-one-left-nat [of 2 ^ (multiplicity $2 m+1$ )]
by (auto simp add: coprime-dvd-mult-left-iff)
then have bdef: ? $n p * ? B=? A$
by $\operatorname{simp}$
with $a 0$ have $b 0: ? B>0$ by (metis grOI mult-is-0)
from nplarger a 0 have bsmallera: ? $B<$ ?A by auto
have $? B=1$
proof (rule ccontr)
assume ? $B \neq 1$
with b0 bsmallera have $1<? B$ ? $B<? A$ by auto
moreover from bdef have ? $B$ : divisors ? $A$ by (metis divisors-eq-dvd dvd-triv-right)
ultimately have $1+? B+? A \leq$ sigma $? A$ using sigma-third-divisor by blast
with nplarger have ? $n p *(1+? A+$ ? $B) \leq ? n p *($ sigma ? $A)$ by (auto simp only: nat-mult-le-cancel1)
with bdef have ? $n p+? A * ? n p+? A * 1 \leq ? n p *($ sigma ? $A)$ by (simp add: mult.commute distrib-left)
hence ? $n p+? A *(? n p+1) \leq ? n p *($ sigma ? A) by (simp only:add-mult-distrib2)
with nplarger have 2 ^(? $n+1) * ? A<? n p *($ sigma ?A) by (simp add:mult.commute)
with formula show False by auto
qed
with bdef have adef: ? $A=? n p$ by auto
with formula have ? $n p * \mathcal{Z}^{\wedge}(? n+1)=? n p * \operatorname{sigma}(? A)$ by auto
with nplarger adef have ? A $+1=\operatorname{sigma}(? A)$ by auto
with a0 have prime ? A
by (simp add: prime-iff-sigma)
with mdef adef show $m=2^{\wedge} ? n * ? n p \wedge$ prime ? $n p$ by simp qed

```
theorem Euclid-book9-prop36:
    assumes p: prime (2`(n+1) - (1::nat))
    shows perfect (2 ^ n * (2^ (n+1) - 1))
    unfolding perfect-def
proof (intro conjI; simp)
    from assms show 2 * 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`(n+1) - 1)
        by (metis p primes-coprime-nat two-is-prime-nat)
    moreover with p have 2^(n+1) - 1> (0::nat)
        by (auto simp add: prime-nat-iff)
```



```
- 1))
    by (metis sigma-semimultiplicative two-is-prime-nat)
    also from assms have ... = (sigma(2`(n)))*(2^(n+1))
    by (auto simp add: prime-imp-sigma)
    also have ... = (2`(n+1) - 1)*(\mp@subsup{\mathbb{R}}{}{`}(n+1)) by(simp add: sigma-prime-power-two)
    finally show 2*(2`n*(2*2`n}-\mathrm{ Suc 0)) = sigma(2`n*(2*2`n}-Suc 0)) b
auto
qed
end
```


## References

[Wie] Freek Wiedijk. Formalizing 100 theorems. http://www.cs.ru.nl/ $\sim$ freek/100/.

