

# Euler's Partition Theorem

Lukas Bulwahn

May 21, 2019

## Abstract

Euler's Partition Theorem states that the number of partitions with only distinct parts is equal to the number of partitions with only odd parts. The combinatorial proof follows John Harrison's pre-existing HOL Light formalization [1]. To understand the rough idea of the proof, I read the lecture notes of the MIT course 18.312 on Algebraic Combinatorics [2] by Gregg Musiker. This theorem is the 45th theorem of the Top 100 Theorems list.

## Contents

<b>1 Euler's Partition Theorem</b>	<b>1</b>
1.1 Preliminaries	2
1.1.1 Additions to Divides Theory	2
1.1.2 Additions to Groups-Big Theory	2
1.1.3 Additions to Set-Interval Theory	2
1.1.4 Additions to Nat Theory or Power Theory	2
1.1.5 Additions to Finite-Set Theory	2
1.2 Binary Encoding of Natural Numbers	2
1.3 Decomposition of a Number into a Power of Two and an Odd Number	4
1.4 Partitions With Only Distinct and Only Odd Parts	5
1.5 Euler's Partition Theorem	6

## 1 Euler's Partition Theorem

```
theory Euler-Partition
imports
  Main
  Card-Number-Partitions.Number-Partition
begin
```

## 1.1 Preliminaries

### 1.1.1 Additions to Divides Theory

**lemma** *power-div-nat*:  
 assumes  $c \leq b$   
 assumes  $a > 0$   
 shows  $(a :: nat) \wedge b \text{ div } a \wedge c = a \wedge (b - c)$   
 *<proof>*

### 1.1.2 Additions to Groups-Big Theory

**lemma** *sum-div*:  
 assumes *finite*  $A$   
 assumes  $\bigwedge a. a \in A \implies (b :: 'b :: euclidean-semiring) \text{ dvd } f a$   
 shows  $(\sum a \in A. f a) \text{ div } b = (\sum a \in A. (f a) \text{ div } b)$   
 *<proof>*

**lemma** *sum-mod*:  
 assumes *finite*  $A$   
 assumes  $\bigwedge a. a \in A \implies f a \text{ mod } b = (0 :: 'b :: unique-euclidean-semiring)$   
 shows  $(\sum a \in A. f a) \text{ mod } b = 0$   
 *<proof>*

### 1.1.3 Additions to Set-Interval Theory

**lemma** *geometric-sum-2nat*:  
  $(\sum i < n. (2 :: nat) \wedge i) = (2 \wedge n - 1)$   
 *<proof>*

### 1.1.4 Additions to Nat Theory or Power Theory

**lemma** *n-leq-2-pow-n*:  
  $n \leq 2 \wedge n$   
 *<proof>*

### 1.1.5 Additions to Finite-Set Theory

**lemma** *finite-exponents*:  
 *finite*  $\{i. 2 \wedge i \leq (n :: nat)\}$   
 *<proof>*

## 1.2 Binary Encoding of Natural Numbers

**definition** *bitset* ::  $nat \Rightarrow nat \text{ set}$   
**where**  
  $bitset\ n = \{i. \text{odd } (n \text{ div } (2 \wedge i))\}$

**lemma** *in-bitset-bound*:  
  $b \in \text{bitset } n \implies 2 \wedge b \leq n$   
 *<proof>*

**lemma** *in-bitset-bound-weak*:

$b \in \text{bitset } n \implies b \leq n$

*<proof>*

**lemma** *finite-bitset*:

$\text{finite } (\text{bitset } n)$

*<proof>*

**lemma** *bitset-0*:

$\text{bitset } 0 = \{\}$

*<proof>*

**lemma** *binary-induct* [*case-names zero even odd*]:

**assumes**  $P (0 :: \text{nat})$

**assumes**  $\bigwedge n. P n \implies P (2 * n)$

**assumes**  $\bigwedge n. P n \implies P (2 * n + 1)$

**shows**  $\bigwedge n. P n$

*<proof>*

**lemma** *bitset-2n*:  $\text{bitset } (2 * n) = \text{Suc } ' (\text{bitset } n)$

*<proof>*

**lemma** *bitset-Suc*:

**assumes** *even*  $n$

**shows**  $\text{bitset } (n + 1) = \text{insert } 0 (\text{bitset } n)$

*<proof>*

**lemma** *bitset-2n1*:

$\text{bitset } (2 * n + 1) = \text{insert } 0 (\text{Suc } ' (\text{bitset } n))$

*<proof>*

**lemma** *sum-bitset*:

$(\sum_{i \in \text{bitset } n} 2^i) = n$

*<proof>*

**lemma** *binarysum-div*:

**assumes** *finite*  $B$

**shows**  $(\sum_{i \in B} (2^{::\text{nat}})^i \text{ div } 2^j) = (\sum_{i \in B} \text{if } i < j \text{ then } 0 \text{ else } 2^{(i - j)})$

**(is - =**  $(\sum_{i \in -} ?f i)$ )

*<proof>*

**lemma** *odd-iff*:

**assumes** *finite*  $B$

**shows**  $\text{odd } (\sum_{i \in B} \text{if } i < x \text{ then } (0^{::\text{nat}}) \text{ else } 2^{(i - x)}) = (x \in B)$  **(is odd**  $(\sum_{i \in -} ?s i) = -)$

*<proof>*

**lemma** *bitset-sum*:  
**assumes** *finite B*  
**shows**  $\text{bitset } (\sum_{i \in B}. 2^i) = B$   
 $\langle \text{proof} \rangle$

### 1.3 Decomposition of a Number into a Power of Two and an Odd Number

**function** (*sequential*) *index* ::  $\text{nat} \Rightarrow \text{nat}$   
**where**  
*index* 0 = 0  
| *index* n = (if odd n then 0 else Suc (*index* (n div 2)))  
 $\langle \text{proof} \rangle$

**termination**  
 $\langle \text{proof} \rangle$

**function** (*sequential*) *oddp* ::  $\text{nat} \Rightarrow \text{nat}$   
**where**  
*oddp* 0 = 0  
| *oddp* n = (if odd n then n else *oddp* (n div 2))  
 $\langle \text{proof} \rangle$

**termination**  
 $\langle \text{proof} \rangle$

**lemma** *odd-oddp*:  
 $\text{odd } (\text{oddp } n) \longleftrightarrow n \neq 0$   
 $\langle \text{proof} \rangle$

**lemma** *index-oddp-decomposition*:  
 $n = 2^{\text{index } n} * \text{oddp } n$   
 $\langle \text{proof} \rangle$

**lemma** *oddp-leq*:  
 $\text{oddp } n \leq n$   
 $\langle \text{proof} \rangle$

**lemma** *index-oddp-unique*:  
**assumes** *odd (m :: nat) odd m'*  
**shows**  $(2^i * m = 2^{i'} * m') \longleftrightarrow (i = i' \wedge m = m')$   
 $\langle \text{proof} \rangle$

**lemma** *index-oddp*:  
**assumes** *odd m*  
**shows**  $\text{index } (2^i * m) = i \text{ oddpart } (2^i * m) = m$   
 $\langle \text{proof} \rangle$

## 1.4 Partitions With Only Distinct and Only Odd Parts

**definition** *odd-of-distinct* :: (nat  $\Rightarrow$  nat)  $\Rightarrow$  nat  $\Rightarrow$  nat

**where**

$$\text{odd-of-distinct } p = (\lambda i. \text{if odd } i \text{ then } (\sum j \mid p (2^j * i) = 1. 2^j) \text{ else } 0)$$

**definition** *distinct-of-odd* :: (nat  $\Rightarrow$  nat)  $\Rightarrow$  nat  $\Rightarrow$  nat

**where**

$$\text{distinct-of-odd } p = (\lambda i. \text{if index } i \in \text{bitset } (p (\text{oddpart } i)) \text{ then } 1 \text{ else } 0)$$

**lemma** *odd*:

$$\text{odd-of-distinct } p \ i \neq 0 \implies \text{odd } i$$

*<proof>*

**lemma** *distinct-distinct-of-odd*:

$$\text{distinct-of-odd } p \ i \leq 1$$

*<proof>*

**lemma** *odd-of-distinct*:

**assumes** *odd-of-distinct*  $p \ i \neq 0$

**assumes**  $\bigwedge i. p \ i \neq 0 \implies i \leq n$

**shows**  $1 \leq i \wedge i \leq n$

*<proof>*

**lemma** *distinct-of-odd*:

**assumes**  $\bigwedge i. p \ i * i \leq n \wedge i. p \ i \neq 0 \implies \text{odd } i$

**assumes** *distinct-of-odd*  $p \ i \neq 0$

**shows**  $1 \leq i \wedge i \leq n$

*<proof>*

**lemma** *odd-distinct*:

**assumes**  $\bigwedge i. p \ i \neq 0 \implies \text{odd } i$

**shows** *odd-of-distinct* (*distinct-of-odd*  $p$ ) =  $p$

*<proof>*

**lemma** *distinct-odd*:

**assumes**  $\bigwedge i. p \ i \neq 0 \implies 1 \leq i \wedge i \leq n \wedge i. p \ i \leq 1$

**shows** *distinct-of-odd* (*odd-of-distinct*  $p$ ) =  $p$

*<proof>*

**lemma** *sum-distinct-of-odd*:

**assumes**  $\bigwedge i. p \ i \neq 0 \implies 1 \leq i \wedge i \leq n$

**assumes**  $\bigwedge i. p \ i * i \leq n$

**assumes**  $\bigwedge i. p \ i \neq 0 \implies \text{odd } i$

**shows**  $(\sum i \leq n. \text{distinct-of-odd } p \ i * i) = (\sum i \leq n. p \ i * i)$

*<proof>*

**lemma** *leq-n*:

**assumes**  $\forall i. 0 < p \ i \longrightarrow 1 \leq i \wedge i \leq (n::\text{nat})$

**assumes**  $(\sum i \leq n. p \ i * i) = n$

**shows**  $p\ i * i \leq n$   
*<proof>*

**lemma** *distinct-of-odd-in-distinct-partitions:*

**assumes**  $p \in \{p. p\ \text{partitions}\ n \wedge (\forall i. p\ i \neq 0 \longrightarrow \text{odd}\ i)\}$   
**shows**  $\text{distinct-of-odd}\ p \in \{p. p\ \text{partitions}\ n \wedge (\forall i. p\ i \leq 1)\}$   
*<proof>*

**lemma** *odd-of-distinct-in-odd-partitions:*

**assumes**  $p \in \{p. p\ \text{partitions}\ n \wedge (\forall i. p\ i \leq 1)\}$   
**shows**  $\text{odd-of-distinct}\ p \in \{p. p\ \text{partitions}\ n \wedge (\forall i. p\ i \neq 0 \longrightarrow \text{odd}\ i)\}$   
*<proof>*

## 1.5 Euler's Partition Theorem

**theorem** *Euler-partition-theorem:*

$\text{card}\ \{p. p\ \text{partitions}\ n \wedge (\forall i. p\ i \leq 1)\} = \text{card}\ \{p. p\ \text{partitions}\ n \wedge (\forall i. p\ i \neq 0 \longrightarrow \text{odd}\ i)\}$   
(**is**  $\text{card}\ \text{?distinct-partitions} = \text{card}\ \text{?odd-partitions}$ )  
*<proof>*

**end**

## References

- [1] J. Harrison. Euler's partition theorem and other elementary partition theorems. <https://github.com/jrh13/hol-light/blob/master/100/euler.ml>.
- [2] G. Musiker. Course 18.312: Algebraic combinatorics, 2009. [http://ocw.mit.edu/courses/mathematics/18-312-algebraic-combinatorics-spring-2009/readings-and-lecture-notes/MIT18\\_312S09\\_lec10\\_Patitio.pdf](http://ocw.mit.edu/courses/mathematics/18-312-algebraic-combinatorics-spring-2009/readings-and-lecture-notes/MIT18_312S09_lec10_Patitio.pdf).