Bounding LCM with Triangles

How the Proof becomes a Pearl

Hing-Lun Chan and Michael Norrish

College of Engineering and Computer Science
Australian National University

October 17, 2018, ANU.

Refresh

How to play with numbers

$$1 \quad 2 \quad 3 \quad \ldots \quad n$$

How to play with numbers

$$\frac{1 \quad 2 \quad 3 \quad \ldots \quad n}{1 + 2 + 3 + \ldots + n}$$

How to play with numbers

$$\frac{\begin{array}{ccccc} 1 & 2 & 3 & \ldots & n \end{array}}{\frac{1}{2}n^2 \leq 1 + 2 + 3 + \ldots + n = \frac{1}{2}n(n+1)}$$

How to play with numbers

$$\begin{array}{rcl} & 1 \quad 2 \quad 3 \quad \ldots \quad n & \\ \hline \frac{1}{2}n^2 \le & 1 + 2 + 3 + \ldots + n & = \frac{1}{2}n(n+1) \\ & 1 \times 2 \times 3 \times \ldots \times n & \end{array}$$

How to play with numbers

$$
\begin{array}{rcl}
 & 1 \quad 2 \quad 3 \quad \ldots \quad n & \\
\hline
\frac{1}{2}n^2 \leq & 1 + 2 + 3 + \ldots + n & = \frac{1}{2}n(n+1) \\
\sqrt{2\pi}\sqrt{n}\,(n/e)^n \leq & 1 \times 2 \times 3 \times \ldots \times n & \leq \; e\sqrt{n}\,(n/e)^n
\end{array}
$$

How to play with numbers

$$
\begin{array}{rcl}
 & 1 \quad 2 \quad 3 \quad \ldots \quad n & \\
\hline
\frac{1}{2}n^2 \le & 1 + 2 + 3 + \ldots + n & = \frac{1}{2}n(n+1) \\
\sqrt{2\pi}\sqrt{n}(n/e)^n \le & 1 \times 2 \times 3 \times \ldots \times n & \le e\sqrt{n}(n/e)^n \\
 & 1 \circ 2 \circ 3 \circ \ldots \circ n &
\end{array}
$$

$a \circ b = \texttt{gcd}(a, b)$, their greatest common divisor.

How to play with numbers

$$\begin{array}{rcl} & 1 \quad 2 \quad 3 \quad \ldots \quad n & \\ \hline \frac{1}{2}n^2 \leq & 1 + 2 + 3 + \ldots + n & = \frac{1}{2}n(n+1) \\ \sqrt{2\pi}\sqrt{n}(n/e)^n \leq & 1 \times 2 \times 3 \times \ldots \times n & \leq e\sqrt{n}(n/e)^n \\ 1 = & 1 \circ 2 \circ 3 \circ \ldots \circ n & = 1 \end{array}$$

$a \circ b = \texttt{gcd}(a, b)$, their greatest common divisor.

How to play with numbers

$$
\begin{array}{rcl}
 & 1 \quad 2 \quad 3 \quad \ldots \quad n & \\
\hline
\frac{1}{2}n^2 \leq & 1 + 2 + 3 + \ldots + n & = \frac{1}{2}n(n+1) \\
\sqrt{2\pi}\sqrt{n}(n/e)^n \leq & 1 \times 2 \times 3 \times \ldots \times n & \leq e\sqrt{n}(n/e)^n \\
1 = & 1 \circ 2 \circ 3 \circ \ldots \circ n & = 1 \\
 & 1 \diamond 2 \diamond 3 \diamond \ldots \diamond n &
\end{array}
$$

$a \circ b = \texttt{gcd}(a, b)$, their greatest common divisor.
$a \diamond b = \texttt{lcm}(a, b)$, their least common multiple.

How to play with numbers

$$
\begin{array}{rcl}
 & 1 \quad 2 \quad 3 \quad \ldots \quad n & \\
\hline
\frac{1}{2}n^2 \leq & 1 + 2 + 3 + \ldots + n & = \frac{1}{2}n(n+1) \\
\sqrt{2\pi}\sqrt{n}\,(n/e)^n \leq & 1 \times 2 \times 3 \times \ldots \times n & \leq\ e\sqrt{n}\,(n/e)^n \\
1 = & 1 \circ 2 \circ 3 \circ \ldots \circ n & = 1 \\
\frac{1}{2}2^n \leq & 1 \diamond 2 \diamond 3 \diamond \ldots \diamond n & \leq 4^n
\end{array}
$$

$a \circ b = \texttt{gcd}(a, b)$, their greatest common divisor.
$a \diamond b = \texttt{lcm}(a, b)$, their least common multiple.

How to play with numbers

$$
\begin{array}{rcccccccccl}
 & 1 & & 2 & & 3 & & \ldots & & n & \\
\hline
\frac{1}{2}n^2 \leq & 1 & + & 2 & + & 3 & + & \ldots & + & n & = \frac{1}{2}n(n+1) \\
\sqrt{2\pi}\sqrt{n}(n/e)^n \leq & 1 & \times & 2 & \times & 3 & \times & \ldots & \times & n & \leq e\sqrt{n}(n/e)^n \\
1 = & 1 & \circ & 2 & \circ & 3 & \circ & \ldots & \circ & n & = 1 \\
\frac{1}{2}2^n \leq & 1 & \diamond & 2 & \diamond & 3 & \diamond & \ldots & \diamond & n & \leq 4^n
\end{array}
$$

$a \circ b = \texttt{gcd}(a, b)$, their greatest common divisor.
$a \diamond b = \texttt{lcm}(a, b)$, their least common multiple.

Theorem (Lower Bound for Consecutive LCM)

$\vdash \texttt{LCM [}1\ \texttt{..}\ \texttt{(}n\ \texttt{+}\ 1\texttt{)]} \geq 2^n$

How to play with triangles

						1						
					1		1					
				1		2		1				
			1		3		3		1			
		1		4		6		4		1		
	1		5		10		10		5		1	
1		6		15		20		15		6		1

Pascal's Triangle

- Boundary entry: always 1.
- Inside entry: sum of two immediate parents.

How to play with triangles

						1						
					1		1					
				1		2		1				
			1		3		3		1			
		1		4		6		4		1		
	1		5		10		10		5		1	
1		6		15		20		15		6		1

Pascal's Triangle

- Boundary entry: always 1.
- Inside entry: sum of two immediate parents.

How to play with triangles

$$
\begin{array}{ccccccccccccc}
 & & & & & & 1 & & & & & & \\
 & & & & & 1 & & 1 & & & & & \\
 & & & & 1 & & 2 & & 1 & & & & \\
 & & & 1 & & 3 & & 3 & & 1 & & & \\
 & & 1 & & 4 & & 6 & & 4 & & 1 & & \\
 & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & \\
1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1
\end{array}
$$

Pascal's Triangle

- Boundary entry: always 1.
- Inside entry: sum of two immediate parents.

Sum of the n-th row:

$$\sum_{k=0}^{n} \binom{n}{k} = (1+1)^n = 2^n$$

Triangle Pattern

$$
\begin{array}{ccccccccccccc}
 & & & & & & 1 & & & & & & \\
 & & & & & 1 & & 1 & & & & & \\
 & & & & 1 & & 2 & & 1 & & & & \\
 & & & 1 & & 3 & & 3 & & 1 & & & \\
 & & 1 & & 4 & & 6 & & 4 & & 1 & & \\
 & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & \\
1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1
\end{array}
\Rightarrow
\begin{array}{ccccccc}
1 & & & & & & \\
1 & 1 & & & & & \\
1 & 2 & 1 & & & & \\
1 & 3 & 3 & 1 & & & \\
1 & 4 & 6 & 4 & 1 & & \\
1 & 5 & 10 & 10 & 5 & 1 & \\
1 & 6 & 15 & 20 & 15 & 6 & 1
\end{array}
$$

Pascal's Triangle: symmetrical to vertical-horizontal.

Triangle Pattern

$$
\begin{array}{ccccccccccccc}
&&&&&&1&&&&&& \\
&&&&&1&&1&&&&& \\
&&&&1&&2&&1&&&& \\
&&&1&&3&&3&&1&&& \\
&&1&&4&&6&&4&&1&& \\
&1&&5&&10&&10&&5&&1& \\
1&&6&&15&&20&&15&&6&&1
\end{array}
\quad \Rightarrow \quad
\begin{array}{lllllll}
1 &&&&&& \\
1 & 1 &&&&& \\
1 & 2 & 1 &&&& \\
1 & 3 & 3 & 1 &&& \\
1 & 4 & 6 & 4 & 1 && \\
1 & 5 & 10 & 10 & 5 & 1 & \\
1 & 6 & 15 & 20 & 15 & 6 & 1
\end{array}
$$

Pascal’s Triangle: symmetrical to vertical-horizontal.
Turns a triplet (entry with parents) into an inverted-L.

Playing with Patterns

1						
1	1					
1	2	1				
1	3	3	1			
1	4	6	4	1		
1	5	10	10	5	1	
1	6	15	20	15	6	1

Multiplying each row by the number of elements in that row.

Playing with Patterns

1							$\Leftarrow \times 1$	1							
2	2						$\Leftarrow \times 2$	1	1						
3	6	3					$\Leftarrow \times 3$	1	2	1					
4	12	12	4				$\Leftarrow \times 4$	1	3	3	1				
5	20	30	20	5			$\Leftarrow \times 5$	1	4	6	4	1			
6	30	60	60	30	6		$\Leftarrow \times 6$	1	5	10	10	5	1		
7	42	105	140	105	42	7	$\Leftarrow \times 7$	1	6	15	20	15	6	1	

Multiplying each row by the number of elements in that row.

Playing with Patterns

$$
\begin{array}{ccccccccl}
1 & & & & & & & \Leftarrow \times 1 & 1 \\
2 & 2 & & & & & & \Leftarrow \times 2 & 1 \;\; 1 \\
3 & 6 & 3 & & & & & \Leftarrow \times 3 & 1 \;\; 2 \;\; 1 \\
4 & 12 & 12 & 4 & & & & \Leftarrow \times 4 & 1 \;\; 3 \;\; 3 \;\; 1 \\
5 & 20 & 30 & 20 & 5 & & & \Leftarrow \times 5 & 1 \;\; 4 \;\; 6 \;\; 4 \;\; 1 \\
6 & 30 & 60 & 60 & 30 & 6 & & \Leftarrow \times 6 & 1 \;\; 5 \;\; 10 \;\; 10 \;\; 5 \;\; 1 \\
7 & 42 & 105 & 140 & 105 & 42 & 7 & \Leftarrow \times 7 & 1 \;\; 6 \;\; 15 \;\; 20 \;\; 15 \;\; 6 \;\; 1
\end{array}
$$

Multiplying each row by the number of elements in that row.
Result: Leibniz's Triangle, in denominator form.

Leibniz's Harmonic Triangle

$$
\begin{array}{lllllll}
1 \\
2 & 2 \\
3 & 6 & 3 \\
4 & 12 & 12 & 4 \\
5 & 20 & 30 & 20 & 5 \\
6 & 30 & 60 & 60 & 30 & 6 \\
7 & 42 & 105 & 140 & 105 & 42 & 7
\end{array}
\quad \Rightarrow \quad
\begin{array}{ccccccccccccc}
 & & & & & & \frac{1}{1} \\
 & & & & & \frac{1}{2} & & \frac{1}{2} \\
 & & & & \frac{1}{3} & & \frac{1}{6} & & \frac{1}{3} \\
 & & & \frac{1}{4} & & \frac{1}{12} & & \frac{1}{12} & & \frac{1}{4} \\
 & & \frac{1}{5} & & \frac{1}{20} & & \frac{1}{30} & & \frac{1}{20} & & \frac{1}{5} \\
 & \frac{1}{6} & & \frac{1}{30} & & \frac{1}{60} & & \frac{1}{60} & & \frac{1}{30} & & \frac{1}{6} \\
\frac{1}{7} & & \frac{1}{42} & & \frac{1}{105} & & \frac{1}{140} & & \frac{1}{105} & & \frac{1}{42} & & \frac{1}{7}
\end{array}
$$

Leibniz's Triangle in usual form (*e.g.*, Wikipedia).

Leibniz's Harmonic Triangle

1						
2	2					
3	6	3				
4	12	12	4			
5	20	30	20	5		
6	30	60	60	30	6	
7	42	105	140	105	42	7

$\Rightarrow$

$$
\begin{array}{ccccccccccccc}
 & & & & & & \frac{1}{1} & & & & & & \\
 & & & & & \frac{1}{2} & & \frac{1}{2} & & & & & \\
 & & & & \frac{1}{3} & & \frac{1}{6} & & \frac{1}{3} & & & & \\
 & & & \frac{1}{4} & & \frac{1}{12} & & \frac{1}{12} & & \frac{1}{4} & & & \\
 & & \frac{1}{5} & & \frac{1}{20} & & \frac{1}{30} & & \frac{1}{20} & & \frac{1}{5} & & \\
 & \frac{1}{6} & & \frac{1}{30} & & \frac{1}{60} & & \frac{1}{60} & & \frac{1}{30} & & \frac{1}{6} & \\
\frac{1}{7} & & \frac{1}{42} & & \frac{1}{105} & & \frac{1}{140} & & \frac{1}{105} & & \frac{1}{42} & & \frac{1}{7}
\end{array}
$$

Leibniz's Triangle in usual form (*e.g.*, Wikipedia).
Build-up unit: L-triplet (an entry with children).

Leibniz's Harmonic Triangle

1						
2	2					
3	6	3				
4	12	12	4			
5	20	30	20	5		
6	30	60	60	30	6	
7	42	105	140	105	42	7

$\Rightarrow$

$$
\begin{array}{ccccccccccccc}
 & & & & & & \frac{1}{1} & & & & & & \\
 & & & & & \frac{1}{2} & & \frac{1}{2} & & & & & \\
 & & & & \frac{1}{3} & & \frac{1}{6} & & \frac{1}{3} & & & & \\
 & & & \frac{1}{4} & & \frac{1}{12} & & \frac{1}{12} & & \frac{1}{4} & & & \\
 & & \frac{1}{5} & & \frac{1}{20} & & \frac{1}{30} & & \frac{1}{20} & & \frac{1}{5} & & \\
 & \frac{1}{6} & & \frac{1}{30} & & \frac{1}{60} & & \frac{1}{60} & & \frac{1}{30} & & \frac{1}{6} & \\
\frac{1}{7} & & \frac{1}{42} & & \frac{1}{105} & & \frac{1}{140} & & \frac{1}{105} & & \frac{1}{42} & & \frac{1}{7}
\end{array}
$$

Leibniz's Triangle in usual form (*e.g.*, Wikipedia).
Build-up unit: L-triplet (an entry with children).

For a Leibniz triplet $\begin{pmatrix} a \\ bc \end{pmatrix}$: $\quad \frac{1}{a} = \frac{1}{b} + \frac{1}{c}$

Consecutive LCM

1					
2	2				
3	6	3			
4	12	12	4		
5	20	30	20	5	
6	30	60	60	30	6

How does this triangle relate to our goal?

Consecutive LCM

1					
2	2				
3	6	3			
4	12	12	4		
5	20	30	20	5	
6	30	60	60	30	6

How does this triangle relate to our goal?
Amazing: $\texttt{LCM}$ (vertical column) $=$ $\texttt{LCM}$ (horizontal row).

Consecutive LCM

1					
2	2				
3	6	3			
4	12	12	4		
5	20	30	20	5	
6	30	60	60	30	6

How does this triangle relate to our goal?
Amazing: $\texttt{LCM}$ (vertical column) $=$ $\texttt{LCM}$ (horizontal row).

This is LCM exchange for big-L,
and $\texttt{LCM}$ (horizontal row) is easier to estimate.

Proof Idea for LCM $[1 \ldots (n+1)] \geq 2^n$

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

`LCM [1, 2, 3, 4, 5]`

Proof Idea for LCM $[1 \ldots (n+1)] \geq 2^n$

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

```
  LCM [1, 2, 3, 4, 5]
= LCM [5, 20, 30, 20, 5]
```

by LCM exchange for big-L

Proof Idea for LCM $[1 \ldots (n+1)] \geq 2^n$

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

	LCM [1, 2, 3, 4, 5]	
=	LCM [5, 20, 30, 20, 5]	by LCM exchange for big-L
=	5 × LCM [1, 4, 6, 4, 1]	taking out common factor

Proof Idea for $\mathtt{LCM}\ [1 \ldots (n+1)] \geq 2^n$

$$
\begin{array}{ccccc}
1 & & & & \\
2 & 2 & & & \\
3 & 6 & 3 & & \\
4 & 12 & 12 & 4 & \\
5 & 20 & 30 & 20 & 5
\end{array}
$$

$$
\begin{array}{lll}
 & \mathtt{LCM}\ [1,\ 2,\ 3,\ 4,\ 5] & \\
= & \mathtt{LCM}\ [5,\ 20,\ 30,\ 20,\ 5] & \text{by LCM exchange for big-L} \\
= & 5 \times \mathtt{LCM}\ [1,\ 4,\ 6,\ 4,\ 1] & \text{taking out common factor} \\
= & \quad \mathtt{LCM}\ [\boxed{1},\ 4,\ 6,\ 4,\ 1] & \text{multiply = repeated add} \\
 & + \mathtt{LCM}\ [1,\ \boxed{4},\ 6,\ 4,\ 1] & \\
 & + \mathtt{LCM}\ [1,\ 4,\ \boxed{6},\ 4,\ 1] & \text{“unrolling”} \\
 & + \mathtt{LCM}\ [1,\ 4,\ 6,\ \boxed{4},\ 1] & \\
 & + \mathtt{LCM}\ [1,\ 4,\ 6,\ 4,\ \boxed{1}] & \text{LCM = least common multiple}
\end{array}
$$

Proof Idea for $\mathtt{LCM}\ [1 \ldots (n+1)] \geq 2^n$

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

$$
\begin{array}{lll}
 & \mathtt{LCM}\ [1,\ 2,\ 3,\ 4,\ 5] & \\
= & \mathtt{LCM}\ [5,\ 20,\ 30,\ 20,\ 5] & \text{by LCM exchange for big-L} \\
= & 5 \times \mathtt{LCM}\ [1,\ 4,\ 6,\ 4,\ 1] & \text{taking out common factor} \\
= & \quad \mathtt{LCM}\ [\ \boxed{1},\ 4,\ 6,\ 4,\ 1\ ] & \text{multiply = repeated add} \\
 & + \mathtt{LCM}\ [\ 1,\ \boxed{4},\ 6,\ 4,\ 1\ ] & \\
 & + \mathtt{LCM}\ [\ 1,\ 4,\ \boxed{6},\ 4,\ 1\ ] & \text{“unrolling”} \\
 & + \mathtt{LCM}\ [\ 1,\ 4,\ 6,\ \boxed{4},\ 1\ ] & \\
 & + \mathtt{LCM}\ [\ 1,\ 4,\ 6,\ 4,\ \boxed{1}\ ] & \text{LCM = least common multiple} \\
\geq & 1 + 4 + 6 + 4 + 1 & \text{picking diagonal elements}
\end{array}
$$

Proof Idea for $\text{LCM}\,[1\ldots(n+1)] \geq 2^n$

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

$$
\begin{aligned}
& \texttt{LCM}\ [1,\ 2,\ 3,\ 4,\ 5] & \\
= \ & \texttt{LCM}\ [5,\ 20,\ 30,\ 20,\ 5] & \text{by LCM exchange for big-L} \\
= \ & 5 \times \texttt{LCM}\ [1,\ 4,\ 6,\ 4,\ 1] & \text{taking out common factor} \\
= \ & \quad \texttt{LCM}\ [\boxed{1},\ 4,\ 6,\ 4,\ 1] & \text{multiply = repeated add} \\
& + \texttt{LCM}\ [1,\ \boxed{4},\ 6,\ 4,\ 1] & \\
& + \texttt{LCM}\ [1,\ 4,\ \boxed{6},\ 4,\ 1] & \text{“unrolling”} \\
& + \texttt{LCM}\ [1,\ 4,\ 6,\ \boxed{4},\ 1] & \\
& + \texttt{LCM}\ [1,\ 4,\ 6,\ 4,\ \boxed{1}] & \text{LCM = least common multiple} \\
\geq \ & 1 + 4 + 6 + 4 + 1 & \text{picking diagonal elements} \\
= \ & (1 + 1)^4 = 2^4 & \text{row sum of Pascal's triangle.}
\end{aligned}
$$

Motivation

AKS mechanisation

PRIMES is in P

Manindra Agrawal Neeraj Kayal
Nitin Saxena*

Abstract

We present an unconditional deterministic polynomial-time algorithm that determines whether an input number is prime or composite.

AKS mechanisation

PRIMES is in P

Manindra Agrawal Neeraj Kayal
Nitin Saxena*

Abstract

We present an unconditional deterministic polynomial-time algorithm that determines whether an input number is prime or composite.

... [the first lemma] ...

We will need the following simple fact about the lcm of first m numbers (see, e.g., [Nai82] for a proof).

Lemma 3.1. *Let $LCM(m)$ denote the lcm of first m numbers. For $m \geq 7$:*

$$LCM(m) \geq 2^m.$$

Need to formalise this LCM lemma, so look up Nair's paper.

Nair's Paper (1982)

ON CHEBYSHEV-TYPE INEQUALITIES FOR PRIMES

M. NAIR
Department of Mathematics, University of Glasgow, Glasgow, Scotland

Proof. For $1 \leqslant m \leqslant n$, consider the integral

$$I = I(m,n) = \int_0^1 x^{m-1}(1-x)^{n-m}\,dx = \sum_{r=0}^{n-m} (-1)^r \binom{n-m}{r} \frac{1}{m+r}. \tag{7}$$

Clearly, $\mathrm{I}d_n \in \mathbb{N}$. On the other hand, repeated integration by parts yields

$$I = 1/m\binom{n}{m}. \tag{8}$$

"Clearly" the integral I, multiplied by $d_n = \texttt{LCM}\,[1 \ldots n]$, is an integer.

Nair's Paper (1982)

ON CHEBYSHEV-TYPE INEQUALITIES FOR PRIMES

M. NAIR
Department of Mathematics, University of Glasgow, Glasgow, Scotland

Proof. For $1 \leqslant m \leqslant n$, consider the integral

$$I = I(m,n) = \int_0^1 x^{m-1}(1-x)^{n-m}\,dx = \sum_{r=0}^{n-m}(-1)^r\binom{n-m}{r}\frac{1}{m+r}. \tag{7}$$

Clearly, $\mathrm{I}d_n \in \mathbb{N}$. On the other hand, repeated integration by parts yields

$$I = 1/m\binom{n}{m}. \tag{8}$$

In conclusion, it is perhaps appropriate to point out that Theorem 3 can also be proved by the standard methods of proof. The interest here lies essentially in the rather curious nature of this proof. It is unexpected to use (i) to prove (ii), and it certainly is strange that there is no mention of primes in the proof of Theorem 3. It also seems worthwhile to point out that there are different ways to prove the identity implied by equations (7) and (8), for example, by expressing $1/x(x+1)\cdots(x+m)$ in partial fractions or by using the difference operator.

"Clearly" the integral I, multiplied by $d_n = \texttt{LCM}\,[1\ldots n]$, is an integer.

The Journey

Math about LCM

- Google: “LCM lower bound”
- Google: “LCM identity”

Math about LCM

- Google: “LCM lower bound”
- Google: “LCM identity”
  - Found this question on Math Stack Exchange:

Is there a direct proof of this lcm identity?

26

7

The identity

$$(n+1)\operatorname{lcm}\left(\binom{n}{0},\binom{n}{1},\ldots\binom{n}{n}\right)=\operatorname{lcm}(1,2,\ldots n+1)$$

is probably not well-known. The only way I know how to prove it is by using Kummer's theorem that the power of p dividing $\binom{a+b}{a}$ is the number of carries needed to add a and b in base p. Is there a more direct proof, e.g. by showing that each side divides the other?

(number-theory) (binomial-coefficients)

share cite improve this question edited Aug 3 '10 at 8:04 asked Aug 3 '10 at 4:18

Math Stack Exchange

3 Answers

active oldest **votes**

19

Consider *Leibniz harmonic triangle* — a table that is like «Pascal triangle reversed»: on it's sides lie numbers $\frac{1}{n}$ and each number is the sum of two beneath it (see the picture).

One can easily proove by induction that m-th number in n-th row of Leibniz triangle is $\frac{1}{(n+1)\binom{n}{m}}$.

So LHS of our identity is just lcd of fractions in n-th row of the triangle.

But it's not hard to see that any such number is an integer linear combination of fractions on triangle's sides (i.e. $1/1, 1/2, \ldots, 1/n$) — and vice versa. So LHS is equal to $lcd(1/1, \ldots, 1/n)$ — and that is exactly RHS.

share cite improve this answer

edited Aug 20 '11 at 13:23
t.b.
54k ● 6 ■ 162 ▲ 237

answered Aug 5 '10 at 19:44
Grigory M
11.6k ● 3 ■ 41 ▲ 90

Math Stack Exchange

3 Answers

active oldest **votes**

19

Consider *Leibniz harmonic triangle* — a table that is like «Pascal triangle reversed»: on it's sides lie numbers $\frac{1}{n}$ and each number is the sum of two beneath it (see the picture).

One can easily proove by induction that m-th number in n-th row of Leibniz triangle is $\frac{1}{(n+1)\binom{n}{m}}$.

So LHS of our identity is just lcd of fractions in n-th row of the triangle.

But it's not hard to see that any such number is an integer linear combination of fractions on triangle's sides (i.e. $1/1, 1/2, \ldots, 1/n$) — and vice versa. So LHS is equal to $lcd(1/1, \ldots, 1/n)$ — and that is exactly RHS.

share cite improve this answer

edited Aug 20 '11 at 13:23
t.b.
54k 6 162 237

answered Aug 5 '10 at 19:44
Grigory M
11.6k 3 41 90

Cryptic “difference operator” means Leibniz’s Harmonic Triangle!

Leibniz triplet: $\begin{pmatrix} a \\ bc \end{pmatrix}$: $\quad \frac{1}{a} = \frac{1}{b} + \frac{1}{c}, \quad$ or $\quad \frac{1}{c} = \frac{1}{a} - \frac{1}{b}.$

Induction Pattern

```
(* LCM Lemma

(n+1) lcm (C(n,0) to C(n,n)) = lcm (1 to (n+1))

m-th number in the n-th row of Leibniz triangle is:  1/ (n+1)C(n,m)

So LHS = lcd (1/1, 1/2, 1/3, ..., 1/n) = RHS = lcm (1,2,3, ..., (n+1)).

0-th row:                 1
1-st row:             1/2  1/2
2-nd row:          1/3  1/6  1/3
3-rd row:      1/4  1/12  1/12  1/4
4-th row: 1/5  1/20  1/30  1/20  1/5

4-th row: 1/5 C(4,m), C(4,m) = 1 4 6 4 1, hence 1/5 1/20 1/30 1/20 1/5
  lcd (1/5 1/20 1/30 1/20 1/5)
= lcm (5, 20, 30, 20, 5)
= lcm (5 C(4,0), 5 C(4,1), 5 C(4,2), 5 C(4,3), 5 C(4,4))
= 5 lcm (C(4,0), C(4,1), C(4,2), C(4,3), C(4,4))

```

Induction Pattern

```
(* LCM Lemma

(n+1) lcm (C(n,0) to C(n,n)) = lcm (1 to (n+1))

m-th number in the n-th row of Leibniz triangle is:  1/ (n+1)C(n,m)

So LHS = lcd (1/1, 1/2, 1/3, ..., 1/n) = RHS = lcm (1,2,3, ..., (n+1)).

0-th row:              1
1-st row:          1/2   1/2
2-nd row:       1/3   1/6   1/3
3-rd row:    1/4   1/12   1/12   1/4
4-th row: 1/5   1/20   1/30   1/20   1/5

4-th row: 1/5 C(4,m), C(4,m) = 1 4 6 4 1, hence 1/5 1/20 1/30 1/20 1/5
  lcd (1/5 1/20 1/30 1/20 1/5)
= lcm (5, 20, 30, 20, 5)
= lcm (5 C(4,0), 5 C(4,1), 5 C(4,2), 5 C(4,3), 5 C(4,4))
= 5 lcm (C(4,0), C(4,1), C(4,2), C(4,3), C(4,4))

```

- How to prove the identity by induction? Need a pattern.

Induction Pattern

```
(* LCM Lemma

(n+1) lcm (C(n,0) to C(n,n)) = lcm (1 to (n+1))

m-th number in the n-th row of Leibniz triangle is:  1/ (n+1)C(n,m)

So LHS = lcd (1/1, 1/2, 1/3, ..., 1/n) = RHS = lcm (1,2,3, ..., (n+1)).

0-th row:              1
1-st row:           1/2  1/2
2-nd row:        1/3  1/6  1/3
3-rd row:     1/4  1/12  1/12  1/4
4-th row: 1/5  1/20  1/30  1/20  1/5

4-th row: 1/5 C(4,m), C(4,m) = 1 4 6 4 1, hence 1/5 1/20 1/30 1/20 1/5
  lcd (1/5 1/20 1/30 1/20 1/5)
= lcm (5, 20, 30, 20, 5)
= lcm (5 C(4,0), 5 C(4,1), 5 C(4,2), 5 C(4,3), 5 C(4,4))
= 5 lcm (C(4,0), C(4,1), C(4,2), C(4,3), C(4,4))
```

- How to prove the identity by induction? Need a pattern.
- Assuming the identity, does it lead to the lower bound?

Finding Pattern

```
Theorem: In the Multiples Triangle, the vertical-lcm = horizontal-lcm.
i.e.     lcm (1, 2, 3) = lcm (3, 6, 3) = 6
         lcm (1, 2, 3, 4) = lcm (4, 12, 12, 4) = 12
         lcm (1, 2, 3, 4, 5) = lcm (5, 20, 30, 20, 5) = 60
         lcm (1, 2, 3, 4, 5, 6) = lcm (6, 30, 60, 60, 30, 6) = 60
Proof: With reference to Leibniz's Triangle, note: term = left-up - left
  lcm (5, 20, 30, 20, 5)
= lcm (5, 20, 30)                     by reduce repetition
= lcm (5, d(1/20), d(1/30))           by denominator of fraction
= lcm (5, d(1/4 - 1/5), d(1/30))      by term = left-up - left
= lcm (5, lcm(4, 5), d(1/12 - 1/20))      by denominator of fraction subtraction
= lcm (5, 4, lcm(12, 20))                 by lcm (a, lcm (a, b)) = lcm (a, b)
= lcm (5, 4, lcm(d(1/12), d(1/20)))       to fraction again
= lcm (5, 4, lcm(d(1/3 - 1/4), d(1/4 - 1/5)))   by Leibniz's Triangle
= lcm (5, 4, lcm(lcm(3,4),     lcm(4,5)))     by fraction subtraction denominator
= lcm (5, 4, lcm(3, 4, 5))                    by lcm merge
= lcm (5, 4, 3)                               merge again
= lcm (5, 4, 3, 2)                            by lcm include factor (!!!)
= lcm (5, 4, 3, 2, 1)                         by lcm include 1
```

A sample of my investigation, by examples.

Promising Result

```
  lcm (1 to 5)                 = 1x2x3x4x5/2 = 60
= 5 lcm (1 4 6 4 1)            = 5 x 12
=  lcm (1 4 6 4 1)             --> unfold 5x to add 5 times
 + lcm (1 4 6 4 1)
 + lcm (1 4 6 4 1)
 + lcm (1 4 6 4 1)
 + lcm (1 4 6 4 1)
>= 1 + 4 + 6 + 4 + 1           --> pick one of each 5 C(n,m), i.e. diagonal
= (1 + 1)^4                    --> fold back binomial
= 2^4                          = 16

Actually, can take 5 lcm (1 4 6 4 1) >= 5 x 6 = 30,
but this will need estimation of C(n, n/2), or C(2n,n), i.e. Stirling's formula.

Theorem: lcm (x y z) >= x  or lcm (x y z) >= y  or lcm (x y z) >= z
```

Figure out that the LCM identity leads to the desired lower bound.

Hit an Idea

```
(* The Idea:
                                                         b
Actually, lcm a b = lcm b c = lcm c a     for   a c  in Leibniz Triangle.
The only relationship is: c = ab/(a - b), or ab = c(a - b).

Is this a theorem:  ab = c(a - b)  ==> lcm a b = lcm b c = lcm c a
Or in fractions,   1/c = 1/b - 1/a ==> lcm a b = lcm b c = lcm c a ?

  lcm a b
= a b / (gcd a b)
= c(a - b) / (gcd a (a - b))
= ac(a - b) / gcd a (a-b) / a
= lcm (a (a-b)) c / a
= lcm (ca c(a-b)) / a
= lcm (ca ab) / a
= lcm b c

  lcm b c
= b c / gcd b c
= a b c / gcd a*b a*c
= a b c / gcd c*(a-b) c*a
= a b / gcd (a-b) a
= a b / gcd b a
= lcm (a b)
= lcm a b
```

Focus on a triplet . . .

Hit an Idea

```
(* The Idea:
                                                           b
Actually, lcm a b = lcm b c = lcm c a      for    a c  in Leibniz Triangle.
The only relationship is: c = ab/(a - b), or ab = c(a - b).

Is this a theorem:  ab = c(a - b)  ==> lcm a b = lcm b c = lcm c a
Or in fractions,   1/c = 1/b - 1/a ==> lcm a b = lcm b c = lcm c a ?

  lcm a b
= a b / (gcd a b)
= c(a - b) / (gcd a (a - b))
= ac(a - b) / gcd a (a-b) / a
= lcm (a (a-b)) c / a
= lcm (ca c(a-b)) / a
= lcm (ca ab) / a
= lcm b c

  lcm b c
= b c / gcd b c
= a b c / gcd a*b a*c
= a b c / gcd c*(a-b) c*a
= a b / gcd (a-b) a
= a b / gcd b a
= lcm (a b)
= lcm a b
```

Focus on a triplet … hope: $\texttt{lcm}\ a\ b = \texttt{lcm}\ b\ c = \texttt{lcm}\ c\ a$.

Voliá

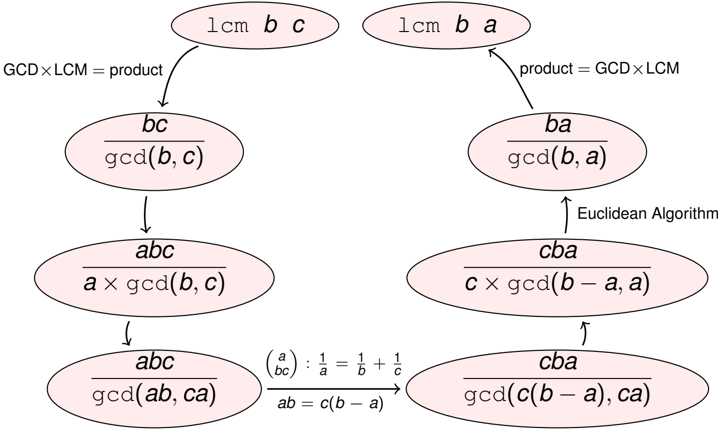
```
(* The Idea:
                                                b
Actually, lcm a b = lcm b c = lcm c a      for   a c  in Leibniz Triangle.
The only relationship is: c = ab/(a - b), or ab = c(a - b).

Is this a theorem:  ab = c(a - b)  ==> lcm a b = lcm b c = lcm c a
Or in fractions,   1/c = 1/b - 1/a ==> lcm a b = lcm b c = lcm c a ?


  lcm a c
= a c / gcd a c
= a b c / gcd b*a b*c
= a b c / gcd c*(a-b) b*c
= a b / gcd (a-b) b
= a b / gcd a b
= lcm a b

Yes!

This is now in LCM_EXCHANGE:
val it = |- !a b c. (a * b = c * (a - b)) ==> (lcm a b = lcm a c): thm
*)
```

Success!

Key Property

Theorem (LCM Exchange)

For a Leibniz triplet $\binom{a}{bc}$, $\texttt{lcm}\ b\ c = \texttt{lcm}\ b\ a$.



Triplets in Triangle

1					
2	2				
3	6	3			
4	12	12	4		
5	20	30	20	5	
6	30	60	60	30	6

Triplets are the building blocks of Leibniz's triangle:

Triplets in Triangle

1					
2	2				
3	6	3			
4	12	12	4		
5	20	30	20	5	
6	30	60	60	30	6

Triplets are the building blocks of Leibniz's triangle:

Triplets in Triangle

1					
2	2				
3	6	3			
4	12	12	4		
5	20	30	20	5	
6	30	60	60	30	6

Triplets are the building blocks of Leibniz's triangle:

- Each triplet has a vertical pair and a horizontal pair.

Triplets in Triangle

1					
2	2				
3	6	3			
4	12	12	4		
5	20	30	20	5	
6	30	60	60	30	6

Triplets are the building blocks of Leibniz's triangle:

- Each triplet has a vertical pair and a horizontal pair.
- We have: $\texttt{LCM}$ (vertical pair) = $\texttt{LCM}$ (horizontal pair)

Triplets in Triangle

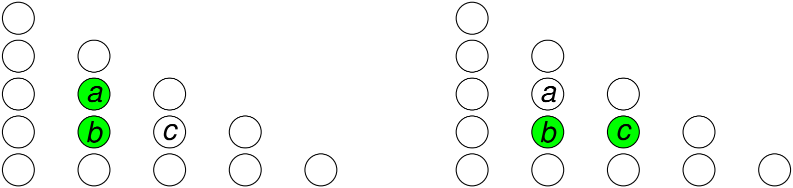
1					
2	2				
3	6	3			
4	12	12	4		
5	20	30	30	5	
6	30	60	60	60	6

Triplets are the building blocks of Leibniz's triangle:

- Each triplet has a vertical pair and a horizontal pair.
- We have: $\texttt{LCM}$ (vertical pair) $=$ $\texttt{LCM}$ (horizontal pair)

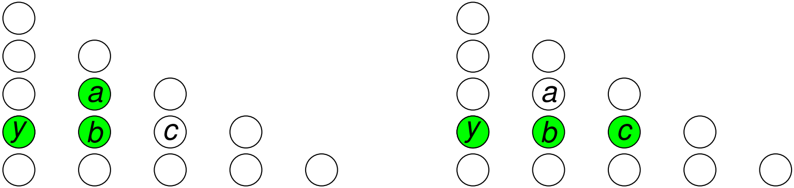
This is LCM exchange for small-L.

Zig-zag Paths



Before $\leftarrow$ identical LCM $\rightarrow$ After

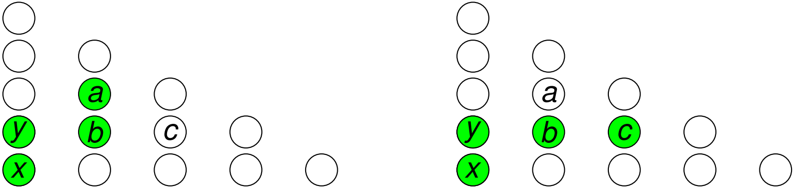
Zig-zag Paths



Before $\leftarrow$ identical LCM $\rightarrow$ After

- Extending a Leibniz triplet, keeping the overall LCM.

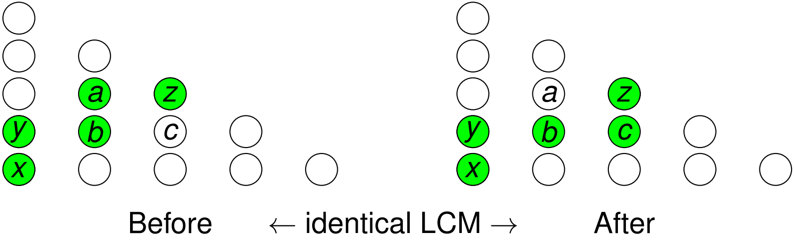
Zig-zag Paths



Before $\leftarrow$ identical LCM $\rightarrow$ After

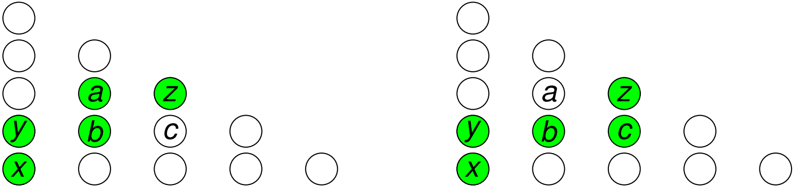
- Extending a Leibniz triplet, keeping the overall LCM.

Zig-zag Paths



- Extending a Leibniz triplet, keeping the overall LCM.
- A path can zig-zag to another by a suitable Leibniz triplet.

Zig-zag Paths



Before $\leftarrow$ identical LCM $\rightarrow$ After

- Extending a Leibniz triplet, keeping the overall LCM.
- A path can zig-zag to another by a suitable Leibniz triplet.

By Leibniz triplet property,

Theorem

$\vdash p_1 \rightsquigarrow p_2 \Rightarrow \mathit{LCM}\ p_1 = \mathit{LCM}\ p_2$

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.
- A path can wriggle to another by successive zig-zags.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.
- A path can wriggle to another by successive zig-zags.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.
- A path can wriggle to another by successive zig-zags.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.
- A path can wriggle to another by successive zig-zags.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.
- A path can wriggle to another by successive zig-zags.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.
- A path can wriggle to another by successive zig-zags.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.
- A path can wriggle to another by successive zig-zags.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.
- A path can wriggle to another by successive zig-zags.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.
- A path can wriggle to another by successive zig-zags.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.
- A path can wriggle to another by successive zig-zags.

Wriggle Paths

1				
2	2			
3	6	3		
4	12	12	4	
5	20	30	20	5

- Transform by successive zig-zags, keeping the overall LCM.
- A path can wriggle to another by successive zig-zags.

By Leibniz triplet property,

Theorem

$\vdash p_1 \rightsquigarrow^* p_2 \Rightarrow \mathit{LCM}\ p_1 = \mathit{LCM}\ p_2$

Polishing

Done and Dusted

- Once the key is proved (SourceTree #1200), back to AKS work.
- Did not bother to formulate path transform: zig-zag and wriggle.
- Establish the LCM lower bound by brute-force induction (#1211).

Done and Dusted

- Once the key is proved (SourceTree #1200), back to AKS work.
- Did not bother to formulate path transform: zig-zag and wriggle.
- Establish the LCM lower bound by brute-force induction (#1211).

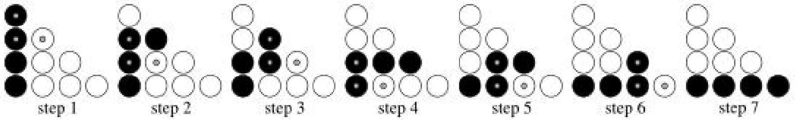
```
   Transform from Vertical LCM to Horizontal LCM:
   leibniz_lcm_shift_one      |- !n k. k <= n ==>
                                 (lcm (list_lcm (TAKE (SUC k) (leibniz_horizontal (SUC n))))
                                      (list_lcm (DROP k (leibniz_horizontal n))) =
                                  lcm (list_lcm (TAKE (SUC (SUC k)) (leibniz_horizontal (SUC n))))
                                      (list_lcm (DROP (SUC k) (leibniz_horizontal n))))
   leibniz_lcm_shift          |- !n k. k <= SUC n ==>
                                 (lcm (list_lcm (TAKE (SUC k) (leibniz_horizontal (SUC n))))
                                      (list_lcm (DROP k (leibniz_horizontal n))) =
                                  lcm (SUC (SUC n)) (list_lcm (leibniz_horizontal n)))
   leibniz_horizontal_lcm     |- !n. list_lcm (leibniz_horizontal (SUC n)) =
                                     lcm (SUC (SUC n)) (list_lcm (leibniz_horizontal n))
   leibniz_lcm_property       |- !n. list_lcm (leibniz_vertical n) = list_lcm (leibniz_horizontal n)

   Binomial Horizontal List:
   binomial_horizontal_def    |- !n. binomial_horizontal n = GENLIST (binomial n) (SUC n)
   binomial_horizontal_0      |- binomial_horizontal 0 = [1]
   binomial_horizontal_len    |- !n. LENGTH (binomial_horizontal n) = n + 1
   binomial_horizontal_pos    |- !n. EVERY (\x. 0 < x) (binomial_horizontal n)
   binomial_horizontal_sum    |- !n. SUM (binomial_horizontal n) = 2 ** n

   Lower Bound of Leibniz LCM:
   leibniz_alt                |- !n. leibniz n = (\k. (n + 1) * k) o binomial n
   leibniz_horizontal_alt     |- !n. leibniz_horizontal n = MAP (\k. (n + 1) * k) (binomial_horizontal n)
   leibniz_horizontal_lcm_alt           |- !n. list_lcm (leibniz_horizontal n) =
                                               (n + 1) * list_lcm (binomial_horizontal n)
   leibniz_horizontal_lcm_lower_bound  |- !n. 2 ** n <= list_lcm (leibniz_horizontal n)
   leibniz_vertical_lcm_lower_bound    |- !n. 2 ** n <= list_lcm (leibniz_vertical n)
*)
```

Back in Spotlight

- Our AKS work, Part 1, was published in ITP2015.
- Plan to submit a paper to ITP2016: on AKS work, Part 2.
- A fortnight before deadline, still working on proof scripts.

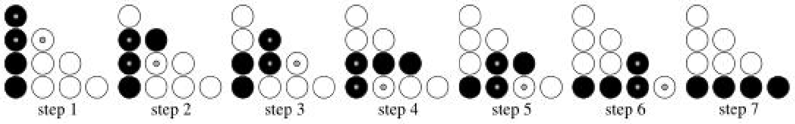
Back in Spotlight

- Our AKS work, Part 1, was published in ITP2015.
- Plan to submit a paper to ITP2016: on AKS work, Part 2.
- A fortnight before deadline, still working on proof scripts.
- Pick this LCM result for the category “Proof Pearl”.
- Explain the proof by drawing this picture:



Back in Spotlight

- Our AKS work, Part 1, was published in ITP2015.
- Plan to submit a paper to ITP2016: on AKS work, Part 2.
- A fortnight before deadline, still working on proof scripts.
- Pick this LCM result for the category “Proof Pearl”.
- Explain the proof by drawing this picture:



Good picture, but the proof script is bad — heaps of induction.

Major Changes

- Formalize in HOL4: path transform, zig-zag and wriggle.
- Reformulate the theorems based on such concepts (#1531).

Major Changes

- Formalize in HOL4: path transform, zig-zag and wriggle.
- Reformulate the theorems based on such concepts (#1531).

```

   Using Triplet and Paths:
   leibniz_zigzag_def    |- !path1 path2. path1 zigzag path2 <=>
                            ?n k part1 part2. (path1 = part1 ++ [tri_b] ++ [tri_a] ++ part2) /\
                                              (path2 = part1 ++ [tri_b] ++ [tri_c] ++ part2)
   leibniz_wriggle_def   |- !path1 path2. path1 wriggle path2 <=>
                            ?m f. (path1 = f 0) /\ (path2 = f m) /\ !k. k < m ==> f k zigzag f (SUC k)
   leibniz_lcm_triple    |- !n k. lcm tri_b tri_a = lcm tri_b tri_c
   list_lcm_zigzag       |- !path1 path2. path1 zigzag path2 ==> (list_lcm path1 = list_lcm path2)
   list_lcm_wriggle      |- !path1 path2. path1 wriggle path2 ==> (list_lcm path1 = list_lcm path2)
   leibniz_zigzag_wriggle   |- !path1 path2. path1 zigzag path2 ==> path1 wriggle path2
   leibniz_zigzag_tail      |- !path1 path2. path1 zigzag path2 ==> !x. [x] ++ path1 zigzag [x] ++ path2
   leibniz_wriggle_tail     |- !path1 path2. path1 wriggle path2 ==> !x. [x] ++ path1 wriggle [x] ++ path2
   leibniz_horizontal_wriggle
                         |- !n. [SUC (SUC n)] ++ leibniz_horizontal n wriggle leibniz_horizontal (SUC n)

   leibniz_up_0          |- leibniz_up 0 = [1]
   leibniz_up_len        |- !n. LENGTH (leibniz_up n) = SUC n
   leibniz_up_cons       |- !n. leibniz_up (SUC n) = SUC (SUC n)::leibniz_up n
   leibniz_triplet_0     |- leibniz_up 1 zigzag leibniz_horizontal 1
   leibniz_up_wriggle_horizontal  |- !n. 0 < n ==> leibniz_up n wriggle leibniz_horizontal n
   leibniz_lcm_property  |- !n. list_lcm (leibniz_vertical n) = list_lcm (leibniz_horizontal n)
*)
```

Major Changes

- Formalize in HOL4: path transform, zig-zag and wriggle.
- Reformulate the theorems based on such concepts (#1531).

```

Using Triplet and Paths:
leibniz_zigzag_def    |- !path1 path2. path1 zigzag path2 <=>
                          ?n k part1 part2. (path1 = part1 ++ [tri_b] ++ [tri_a] ++ part2) /\
                                            (path2 = part1 ++ [tri_b] ++ [tri_c] ++ part2)
leibniz_wriggle_def   |- !path1 path2. path1 wriggle path2 <=>
                          ?m f. (path1 = f 0) /\ (path2 = f m) /\ !k. k < m ==> f k zigzag f (SUC k)
leibniz_lcm_triple    |- !n k. lcm tri_b tri_a = lcm tri_b tri_c
list_lcm_zigzag       |- !path1 path2. path1 zigzag path2 ==> (list_lcm path1 = list_lcm path2)
list_lcm_wriggle      |- !path1 path2. path1 wriggle path2 ==> (list_lcm path1 = list_lcm path2)
leibniz_zigzag_wriggle   |- !path1 path2. path1 zigzag path2 ==> path1 wriggle path2
leibniz_zigzag_tail      |- !path1 path2. path1 zigzag path2 ==> !x. [x] ++ path1 zigzag [x] ++ path2
leibniz_wriggle_tail     |- !path1 path2. path1 wriggle path2 ==> !x. [x] ++ path1 wriggle [x] ++ path2
leibniz_horizontal_wriggle
                      |- !n. [SUC (SUC n)] ++ leibniz_horizontal n wriggle leibniz_horizontal (SUC n)

leibniz_up_0          |- leibniz_up 0 = [1]
leibniz_up_len        |- !n. LENGTH (leibniz_up n) = SUC n
leibniz_up_cons       |- !n. leibniz_up (SUC n) = SUC (SUC n)::leibniz_up n
leibniz_triplet_0     |- leibniz_up 1 zigzag leibniz_horizontal 1
leibniz_up_wriggle_horizontal   |- !n. 0 < n ==> leibniz_up n wriggle leibniz_horizontal n
leibniz_lcm_property  |- !n. list_lcm (leibniz_vertical n) = list_lcm (leibniz_horizontal n)
*)
```

A 12-page draft, with wonderful diagrams, tables and proofs.

Final Touch (by supervisor)

- Cut away half of the draft, keeping only 3 proofs (so 6 pages).
- Define properly a Leibniz triplet, re-arrange diagrams and tables.
- Wriggle is the reflexive transitive closure (RTC) of zig-zag (#1567).

Final Touch (by supervisor)

- Cut away half of the draft, keeping only 3 proofs (so 6 pages).
- Define properly a Leibniz triplet, re-arrange diagrams and tables.
- Wriggle is the reflexive transitive closure (RTC) of zig-zag (#1567).

```

   Wriggle Paths in Leibniz Triangle (old):
   leibniz_old_wriggle_def     |- !p1 p2. p1 old_wriggle p2 <=>
                                  ?m f. (p1 = f 0) /\ (p2 = f m) /\ !k. k < m ==> f k zigzag f (SUC k)
   list_lcm_old_wriggle        |- !p1 p2. p1 old_wriggle p2 ==> (list_lcm p1 = list_lcm p2)
   leibniz_zigzag_old_wriggle  |- !p1 p2. p1 zigzag p2 ==> p1 old_wriggle p2
   leibniz_old_wriggle_tail    |- !p1 p2. p1 old_wriggle p2 ==> !x. [x] ++ p1 old_wriggle [x] ++ p2
   leibniz_old_wriggle_trans   |- !p1 p2 p3. p1 old_wriggle p2 /\ p2 old_wriggle p3 ==> p1 old_wriggle p3
   leibniz_horizontal_old_wriggle   |- !n. [leibniz (n + 1) 0] ++ leibniz_horizontal n old_wriggle
                                        leibniz_horizontal (n + 1)

   Wriggle Paths in Leibniz Triangle (new):
   list_lcm_wriggle         |- !p1 p2. p1 wriggle p2 ==> (list_lcm p1 = list_lcm p2)
   leibniz_zigzag_wriggle   |- !p1 p2. p1 zigzag p2 ==> p1 wriggle p2
   leibniz_wriggle_tail     |- !p1 p2. p1 wriggle p2 ==> !x. [x] ++ p1 wriggle [x] ++ p2
   leibniz_wriggle_trans    |- !p1 p2 p3. p1 wriggle p2 /\ p2 wriggle p3 ==> p1 wriggle p3

   Back to Milestone Theorem:
   leibniz_triplet_0        |- leibniz_up 1 zigzag leibniz_horizontal 1
   leibniz_up_old_wriggle_horizontal  |- !n. 0 < n ==> leibniz_up n old_wriggle leibniz_horizontal n
   leibniz_lcm_property     |- !n. list_lcm (leibniz_vertical n) = list_lcm (leibniz_horizontal n)

*)
```

Final Touch (by supervisor)

- Cut away half of the draft, keeping only 3 proofs (so 6 pages).
- Define properly a Leibniz triplet, re-arrange diagrams and tables.
- Wriggle is the reflexive transitive closure (RTC) of zig-zag (#1567).

```

   Wriggle Paths in Leibniz Triangle (old):
   leibniz_old_wriggle_def     |- !p1 p2. p1 old_wriggle p2 <=>
                                  ?m f. (p1 = f 0) /\ (p2 = f m) /\ !k. k < m ==> f k zigzag f (SUC k)
   list_lcm_old_wriggle        |- !p1 p2. p1 old_wriggle p2 ==> (list_lcm p1 = list_lcm p2)
   leibniz_zigzag_old_wriggle  |- !p1 p2. p1 zigzag p2 ==> p1 old_wriggle p2
   leibniz_old_wriggle_tail    |- !p1 p2. p1 old_wriggle p2 ==> !x. [x] ++ p1 old_wriggle [x] ++ p2
   leibniz_old_wriggle_trans   |- !p1 p2 p3. p1 old_wriggle p2 /\ p2 old_wriggle p3 ==> p1 old_wriggle p3
   leibniz_horizontal_old_wriggle   |- !n. [leibniz (n + 1) 0] ++ leibniz_horizontal n old_wriggle
                                        leibniz_horizontal (n + 1)

   Wriggle Paths in Leibniz Triangle (new):
   list_lcm_wriggle         |- !p1 p2. p1 wriggle p2 ==> (list_lcm p1 = list_lcm p2)
   leibniz_zigzag_wriggle   |- !p1 p2. p1 zigzag p2 ==> p1 wriggle p2
   leibniz_wriggle_tail     |- !p1 p2. p1 wriggle p2 ==> !x. [x] ++ p1 wriggle [x] ++ p2
   leibniz_wriggle_trans    |- !p1 p2 p3. p1 wriggle p2 /\ p2 wriggle p3 ==> p1 wriggle p3

   Back to Milestone Theorem:
   leibniz_triplet_0        |- leibniz_up 1 zigzag leibniz_horizontal 1
   leibniz_up_old_wriggle_horizontal  |- !n. 0 < n ==> leibniz_up n old_wriggle leibniz_horizontal n
   leibniz_lcm_property     |- !n. list_lcm (leibniz_vertical n) = list_lcm (leibniz_horizontal n)

*)
```

Last day: can't complete the RTC induction for wriggle. Help!

ITP2016

Springer Link Search

International Conference on Interactive Theorem Proving
ITP 2016: Interactive Theorem Proving pp 140-150 | Cite as

Proof Pearl: Bounding Least Common Multiples with Triangles

Authors Authors and affiliations

Hing-Lun Chan, Michael Norrish

Conference paper
First Online: 07 August 2016

Part of the Lecture Notes in Computer Science book series (LNCS, volume 9807)

Abstract

We present a proof of the fact that $2^n \leq \text{lcm}\{1, 2, 3, \ldots, (n+1)\}$. This result has a standard proof *via* an integral, but our proof is purely number theoretic, requiring little more than list inductions. The proof is based on manipulations of a variant of Leibniz's Harmonic Triangle, itself a relative of Pascal's better-known Triangle.

Reviews

Review #1, Expertise: high

This paper presents a "proof pearl", a short and clever proof that $2^n \leq lcm(1, ..., n+1)$*. This is not a trivial result: Nair's proof of this fact was published in 1982, and Google search reveals some recent strengthenings and generalizations, but it seems that there is no published elementary proof of this fact.*

Review #1, Expertise: high

This paper presents a "proof pearl", a short and clever proof that $2^n \leq lcm(1, ..., n+1)$*. This is not a trivial result: Nair's proof of this fact was published in 1982, and Google search reveals some recent strengthenings and generalizations, but it seems that there is no published elementary proof of this fact.*

[...], the authors have provided an elegant proof of an interesting result, and have formalized it. It certainly fits the description of a proof pearl.

Review #1, Expertise: high

This paper presents a "proof pearl", a short and clever proof that $2^n \leq lcm(1, ..., n+1)$*. This is not a trivial result: Nair's proof of this fact was published in 1982, and Google search reveals some recent strengthenings and generalizations, but it seems that there is no published elementary proof of this fact.*

[...], the authors have provided an elegant proof of an interesting result, and have formalized it. It certainly fits the description of a proof pearl.

The wording of Theorem 5 *is confusing. [...] How about saying this: [...]*

The reference to the "unrolling" in Section 5 *makes it mysterious, and the proof is needlessly baroque. The argument is simply this: [...]*

Review #2, Expertise: medium

The authors describe a (mechanised) proof of a number-theoretic fact: $2^n \leq lcm(1, ..., n+1)$*. The proof is not new, but the paper is advertised as a pearl.*

Review #2, Expertise: medium

The authors describe a (mechanised) proof of a number-theoretic fact: $2^n \leq lcm(1, ..., n+1)$*. The proof is not new, but the paper is advertised as a pearl.*

In the past I have reviewed several papers that were advertised as pearls, but that in my opinion were not pearls. That is not the case with this paper. I found the text engaging, and easy to follow. The proof is non-trivial, but the authors made it easy to understand for me, and I thought that the mechanisation was presented at a suitable level of detail.

Review #2, Expertise: medium

The authors describe a (mechanised) proof of a number-theoretic fact: $2^n \leq lcm(1, ..., n+1)$*. The proof is not new, but the paper is advertised as a pearl.*

In the past I have reviewed several papers that were advertised as pearls, but that in my opinion were not pearls. That is not the case with this paper. I found the text engaging, and easy to follow. The proof is non-trivial, but the authors made it easy to understand for me, and I thought that the mechanisation was presented at a suitable level of detail.

I strongly recommend the paper for publication.

Review #3, Expertise: medium

This proof pearl shows a lower bound for the least common multiple of the first n integers [...]

Review #3, Expertise: medium

This proof pearl shows a lower bound for the least common multiple of the first n integers [...]

Although the inequality is quite specific, this paper demonstrates that it is worth to search for elegant proofs rather than to apply the golden hammer of a complicated theory. Indeed, the formalised proof is very elementary compared to the published proofs I know of. The authors have done a good job of bringing together the proof ingredients (which have been known) and explaining the proof idea.

Review #3, Expertise: medium

This proof pearl shows a lower bound for the least common multiple of the first n integers [...]

Although the inequality is quite specific, this paper demonstrates that it is worth to search for elegant proofs rather than to apply the golden hammer of a complicated theory. Indeed, the formalised proof is very elementary compared to the published proofs I know of. The authors have done a good job of bringing together the proof ingredients (which have been known) and explaining the proof idea.

In summary, I think that this paper makes a nice proof pearl, and I therefore recommend acceptance.

Epilog

Conclusion

This talk is dedicated to

Michael Norrish,

my supervisor.

- Scripts
  `https://bitbucket.org/jhlchan/hol/src/`
  subfolder: `algebra/lib`.
- Paper
  `https://bitbucket.org/jhlchan/hol/downloads`