The Best Proof is Combinatorial

...and now Reed-Dawson has one

Gara Pruesse

Department of Computing Science

Vancouver Island University

Nanaimo, BC

Combinatorial Proofs

Two types:

1. Two ways of counting one object
2. A bijection proves two objects equally numerous

Combinatorial Proof: Type 1

$$\sum_k \binom{n}{k} = 2^n$$

Combinatorial Proof: Example (Type

Proof: $\sum_k \binom{n}{k} =$ number of subsets of n-set $= 2^n$

Combinatorial Proof: Generally

Given an identity

$$F(n, k) = G(n, k)$$

...find a combinatorial object
show that it is counted by $F(n, k)$
show that it is counted by $G(n, k)$

Combinatorial Proof: Slightly harder

Lemma:

$$\sum_{k \geq 0} \binom{n}{2k} \binom{2k}{k} 2^{n-2k} = \binom{2n}{n}$$

$$\sum_{k\geq 0}\binom{n}{2k}\binom{2k}{k}2^{n-2k}=\binom{2n}{n}$$

Try: The number of words of length n over alphabet $\Gamma = \{a, b, c, d\}$ where $\#a$'s = $\#b$'s.

Proof, cont'd

Clearly $\sum_{k\geq 0}\binom{n}{2k}\binom{2k}{k}2^{n-2k} =$ #(length n words over $\{a, b, c, d\}$ where # a = # b)

#(length n words, #a's =# b's)= $\binom{2n}{n}$

- Let $a = 00, b = 11, c = 01, d = 10$.

#(length n words, #a's =# b's)= $\binom{2n}{n}$

- Let $a = 00, b = 11, c = 01, d = 10$.
- Observe: a bitstring encoding of a string over $\{a, b, c, d\}$ with #a's=#b's has #0's = #1's.

#(length n words, #a's =# b's)= $\binom{2n}{n}$

- Let $a = 00, b = 11, c = 01, d = 10$.
- Observe: a bitstring encoding of a string over $\{a, b, c, d\}$ with #a's=#b's has #0's = #1's.
- Indeed: every bitstring with #0's = #1's is an encoding of an a, b, c, d-string with #a's = #b's.
  0 1 1 0 1 0 1 1 1 1 0 1 0 0 0 0

#(length n words, #a's =# b's)= $\binom{2n}{n}$

- Let $a = 00, b = 11, c = 01, d = 10$.
- Observe: a bitstring encoding of a string over $\{a, b, c, d\}$ with #a's=#b's has #0's = #1's.
- Indeed: every bitstring with #0's = #1's is an encoding of an a, b, c, d-string with #a's = #b's.
  0 1 1 0 1 0 1 1 1 1 0 1 0 0 0 0
- 01 10 10 11 11 01 00 00

#(length n words, #a's =# b's)= $\binom{2n}{n}$

- Let $a = 00, b = 11, c = 01, d = 10$.
- Observe: a bitstring encoding of a string over $\{a, b, c, d\}$ with #a's=#b's has #0's = #1's.
- Indeed: every bitstring with #0's = #1's is an encoding of an a, b, c, d-string with #a's = #b's.
  0 1 1 0 1 0 1 1 1 1 0 1 0 0 0 0
- 01 10 10 11 11 01 00 00
- c d b b c a a

#(length n words, #a's =# b's)= $\binom{2n}{n}$

- Let $a = 00, b = 11, c = 01, d = 10$.
- Observe: a bitstring encoding of a string over $\{a, b, c, d\}$ with #a's=#b's has #0's = #1's.
- Indeed: every bitstring with #0's = #1's is an encoding of an a, b, c, d-string with #a's = #b's.
  0 1 1 0 1 0 1 1 1 1 0 1 0 0 0 0
- 01 10 10 11 11 01 00 00
- c d b b c a a
- Bijection! ... and there are $\binom{2n}{n}$ such bitstrings. $\square$

Reed-Dawson Identity

Reed-Dawson:

$$\sum_{k\geq 0}\binom{n}{k}\binom{2k}{k}(-2)^{n-k} = \begin{cases} 0 & \text{if } n \text{ odd} \\ \binom{n}{n/2} & \text{if } n \text{ even.} \end{cases}$$

Reed-Dawson Identity

Reed-Dawson Identity is also known as Knuth's Old Sum.

Reed-Dawson Identity

Reed-Dawson Identity is also known as Knuth's Old Sum.
Proved by Jonassen and Knuth (1978).

Reed-Dawson Identity

Reed-Dawson Identity is also known as Knuth’s Old Sum.
Proved by Jonassen and Knuth (1978).
Proved by Ira Gessel (reported in Greene and Knuth 1981)

Reed-Dawson Identity

Reed-Dawson Identity is also known as Knuth's Old Sum.

Proved by Jonassen and Knuth (1978).

Proved by Ira Gessel (reported in Greene and Knuth 1981)

Proved by C.C. Rousseau (reported in Greene and Knuth 1981)

Reed-Dawson Identity

Reed-Dawson Identity is also known as Knuth's Old Sum.

Proved by Jonassen and Knuth (1978).

Proved by Ira Gessel (reported in Greene and Knuth 1981)

Proved by C.C. Rousseau (reported in Greene and Knuth 1981)

Proved using an Euler transformation by Prodinger (1994)

Reed-Dawson Identity

Reed-Dawson Identity is also known as Knuth's Old Sum.
Proved by Jonassen and Knuth (1978).
Proved by Ira Gessel (reported in Greene and Knuth 1981)
Proved by C.C. Rousseau (reported in Greene and Knuth 1981)
Proved using an Euler transformation by Prodinger (1994)
Proved in Riordan's book by solving a recurrance (1968)

Reed-Dawson Identity

Reed-Dawson Identity is also known as Knuth’s Old Sum.
Proved by Jonassen and Knuth (1978).
Proved by Ira Gessel (reported in Greene and Knuth 1981)
Proved by C.C. Rousseau (reported in Greene and Knuth 1981)
Proved using an Euler transformation by Prodinger (1994)
Proved in Riordan’s book by solving a recurrance (1968)
Proved Concrete Mathematics using hypergeometric series (Grahan, Knuth and

Reed-Dawson Identity

Reed-Dawson Identity is also known as Knuth’s Old Sum.
Proved by Jonassen and Knuth (1978).
Proved by Ira Gessel (reported in Greene and Knuth 1981)
Proved by C.C. Rousseau (reported in Greene and Knuth 1981)
Proved using an Euler transformation by Prodinger (1994)
Proved in Riordan’s book by solving a recurrance (1968)
Proved Concrete Mathematics using hypergeometric series (Grahan, Knuth and

Reed-Dawson Identity

Reed-Dawson Identity is also known as Knuth's Old Sum.
Proved by Jonassen and Knuth (1978).
Proved by Ira Gessel (reported in Greene and Knuth 1981)
Proved by C.C. Rousseau (reported in Greene and Knuth 1981)
Proved using an Euler transformation by Prodinger (1994)
Proved in Riordan's book by solving a recurrance (1968)
Proved Concrete Mathematics using hypergeometric series (Grahan, Knuth and

Reed-Dawson Identity

Reed-Dawson Identity is also known as Knuth's Old Sum.

Proved by Jonassen and Knuth (1978).

Proved by Ira Gessel (reported in Greene and Knuth 1981)

Proved by C.C. Rousseau (reported in Greene and Knuth 1981)

Proved using an Euler transformation by Prodinger (1994)

Proved in Riordan's book by solving a recurrance (1968)

Proved Concrete Mathematics using hypergeometric series (Grahan, Knuth and

The Combinatorial Proof

$$\sum_{k\geq 0}\binom{n}{k}\binom{2k}{k}(-2)^{n-k} = \begin{cases} 0 & \text{if } n \text{ odd} \\ \binom{n}{n/2} & \text{if } n \text{ even.} \end{cases}$$

The Combinatorial Proof

$$\sum_{k\geq 0}\binom{n}{k}\binom{2k}{k}(-2)^{n-k} = \begin{cases} 0 & \text{if } n \text{ odd} \\ \binom{n}{n/2} & \text{if } n \text{ even.} \end{cases}$$

$$\Gamma = \{a, b, c, d, C, D\}$$

The Combinatorial Proof

$$\sum_{k\geq 0}\binom{n}{k}\binom{2k}{k}(-2)^{n-k}=\begin{cases}0 & \text{if } n \text{ odd}\\ \binom{n}{n/2} & \text{if } n \text{ even.}\end{cases}$$

$\Gamma = \{a, b, c, d, C, D\}$

$L = \{$ words over Γ that have #a's = #b's$\}$

The Combinatorial Proof

$$\sum_{k\geq 0}\binom{n}{k}\binom{2k}{k}(-2)^{n-k} = \begin{cases} 0 & \text{if } n \text{ odd} \\ \binom{n}{n/2} & \text{if } n \text{ even.} \end{cases}$$

$\Gamma = \{a, b, c, d, C, D\}$
$L = \{$ words over Γ that have #a's = #b's$\}$
A word in L is *even* if it has an even number of lower-case letters, *odd* otherwise.

The Combinatorial Proof

t = | #(even words) - #(odd words) | = parity difference

The Combinatorial Proof

$t = |$ #(even words) - #(odd words) $|$ = parity difference

One way to compute t:

sum, for each k, the number of words with k lowercase letters [*(-1) if k odd]

Computing t

Computing t

For a given k, choose from n places for the lowercase letters.

From the Lemma, there are $\binom{2k}{k}$ way to fill those k places so that #a's = #b's.

Computing t

For a given k, choose from n places for the lowercase letters.
From the Lemma, there are $\binom{2k}{k}$ way to fill those k places so that #a's = #b's.
Now select a subset of the $n - k$ remaining place to be filled by C. The remaining places must be filled by D.

Computing t

For a given k, choose from n places for the lowercase letters.
From the Lemma, there are $\binom{2k}{k}$ way to fill those k places so that $\#a$'s = $\#b$'s.
Now select a subset of the $n-k$ remaining place to be filled by C. The remaining places must be filled by D.
Hence

$$t = \sum_{k \geq 0} \binom{n}{k} \binom{2k}{k} 2^{n-k} \times (-1)^k.$$

I.e., t = the right side of the Reed-Dawson

Another way of computing t

$t = |$ #(even words) - #(odd words) $|$.

Another way of computing t

t = $|$ #(even words) - #(odd words) $|$.
Define $\phi(w)$ = change case of the rightmost $c, C, d,$ or D in w.

Another way of computing t

t = | #(even words) - #(odd words) |.
Define $\phi(w)$ = change case of the rightmost $c, C, d,$ or D in w.
E.g. $\phi(abCcdDaCb) =$

Another way of computing t

t = | #(even words) - #(odd words) |.
Define $\phi(w)$ = change case of the rightmost $c, C, d,$ or D in w.
E.g. $\phi(abCcdDaCb) = abCcdDacb$

Another way of computing t

t = | #(even words) - #(odd words) |.
Define $\phi(w)$ = change case of the rightmost $c, C, d,$ or D in w.
E.g. $\phi(abCcdDaCb) = abCcdDacb$
For every word w, w and $\phi(w)$ have different parity

Another way of computing t

t = | #(even words) - #(odd words) |.
Define $\phi(w)$ = change case of the rightmost $c, C, d,$ or D in w.
E.g. $\phi(abCcdDaCb) = abCcdDacb$
For every word w, w and $\phi(w)$ have different parity except when $w = \phi(w)$

Another way of computing t

t = | #(even words) - #(odd words) |.
Define $\phi(w)$ = change case of the rightmost $c, C, d,$ or D in w.
E.g. $\phi(abCcdDaCb) = abCcdDacb$
For every word w, w and $\phi(w)$ have different parity except when $w = \phi(w)$
i.e. when w is only a's and b's.

Another way of computing t

t = | #(even words) - #(odd words) |.
Define $\phi(w)$ = change case of the rightmost $c, C, d,$ or D in w.
E.g. $\phi(abCcdDaCb) = \ abCcdDacb$
For every word w, w and $\phi(w)$ have different parity except when $w = \phi(w)$
i.e. when w is only a's and b's.
$t =$ number of strings over $\{a, b\}$ of length n with #a's = #b's

Another way of computing t

t = | #(even words) - #(odd words) |.
Define $\phi(w)$ = change case of the rightmost $c, C, d,$ or D in w.
E.g. $\phi(abCcdDaCb) = abCcdDacb$
For every word w, w and $\phi(w)$ have different parity except when $w = \phi(w)$
i.e. when w is only a's and b's.
$t =$ number of strings over $\{a, b\}$ of length n with #a's = #b's
$t = 0$ if n is odd, $\binom{n}{n/2}$ otherwise. $\square$

Conclusion

There's no proof so satisfying as a combinatorial proof.

Conclusion

There's no proof so satisfying as a combinatorial proof.

Open Problem:

Number of Spanning Trees of an n-cube $= \prod_{i \geq 0} (2i)^{\binom{n}{i}}$

Conclusion

There's no proof so satisfying as a combinatorial proof.

Open Problem:

Number of Spanning Trees of an n-cube $= \prod_{i \geq 0} (2i)^{\binom{n}{i}}$

Mwaahaahaahaahaaaaaa!