Cauchy's Mean Theorem and the Cauchy-Schwarz Inequality

Benjamin Porter

November 27, 2007

Contents

1 Cauchy's Mean Theorem **3**

1.1 Abstract 3

1.2 Formal proof 4

1.2.1 Collection sum and product 4

1.2.2 Auxillary lemma 5

1.2.3 Mean and GMean 6

1.2.4 *list-neq*, *list-eq* 7

1.2.5 Element selection 8

1.2.6 Abstract properties 8

1.2.7 Existence of a new collection 10

1.2.8 Cauchy's Mean Theorem 11

2 The Cauchy-Schwarz Inequality **12**

2.1 Abstract 12

2.2 Formal Proof 12

2.2.1 Vector, Dot and Norm definitions. 12

Abstract

This document presents the mechanised proofs of two popular theorems attributed to Augustin Louis Cauchy - Cauchy's Mean Theorem and the Cauchy-Schwarz Inequality.

Chapter 1

Cauchy's Mean Theorem

theory *CauchysMeanTheorem*
imports *Complex-Main*
begin

1.1 Abstract

The following document presents a proof of Cauchy's Mean theorem formalised in the Isabelle/Isar theorem proving system.

Theorem: For any collection of positive real numbers the geometric mean is always less than or equal to the arithmetic mean. In mathematical terms:

$$\sqrt[n]{x_1 x_2 \dots x_n} \leq \frac{x_1 + \dots + x_n}{n}$$

We will use the term *mean* to denote the arithmetic mean and *gmean* to denote the geometric mean.

Informal Proof:

This proof is based on the proof presented in [1]. First we need an auxillary lemma (the proof of which is presented formally below) that states:

Given two pairs of numbers of equal sum, the pair with the greater product is the pair with the least difference. Using this lemma we now present the proof -

Given any collection C of positive numbers with mean M and product P and with some element not equal to M we can choose two elements from the collection, a and b where $a > M$ and $b < M$. Remove these elements from the collection and replace them with two new elements, a' and b' such that $a' = M$ and $a' + b' = a + b$. This new collection C' now has a greater product P' but equal mean with respect to C. We can continue in this fashion until we have a collection C_n such that $P_n > P$ and $M_n = M$, but C_n has all its elements equal to M and thus $P_n = M^n$. Using the definition of geometric and arithmetic means above we can see that for any collection of positive

elements E it is always true that gmean E $\leq$ mean E. QED.

[1] Dorrie, H. "100 Great Problems of Elementary Mathematics." 1965, Dover.

1.2 Formal proof

1.2.1 Collection sum and product

The finite collections of numbers will be modelled as lists. We then define sum and product operations over these lists.

Sum and product definitions

definition
listsum :: (*real list*) ⇒ *real* ($\sum$:- [*999*] *1000*) **where**
listsum xs = *foldr op+ xs 0*

definition
listprod :: (*real list*) ⇒ *real* ($\prod$:- [*999*] *1000*) **where**
listprod xs = *foldr op∗ xs 1*

lemma *listsum-empty* [*simp*]: $\sum$:[] = *0*
⟨*proof*⟩

lemma *listsum-cons* [*simp*]: $\sum$:(*a*#*b*) = *a* + $\sum$:*b*
⟨*proof*⟩

lemma *listprod-empty* [*simp*]: $\prod$:[] = *1*
⟨*proof*⟩

lemma *listprod-cons* [*simp*]: $\prod$:(*a*#*b*) = *a* ∗ $\prod$:*b*
⟨*proof*⟩

Properties of sum and product

We now present some useful properties of sum and product over collections.

These lemmas just state that if all the elements in a collection C are less (greater than) than some value m, then the sum will less than (greater than) $m * length(C)$.

lemma *listsum-mono-lt* [*rule-format*]:
fixes *xs*::*real list*
shows *xs* ≠ [] ∧ (∀ *x*∈ *set xs*. *x* < *m*)
⟶ (($\sum$:*xs*) < (*m*∗(*real* (*length xs*))))
⟨*proof*⟩

lemma *listsum-mono-gt* [*rule-format*]:
fixes *xs*::*real list*
shows $xs \neq [] \wedge (\forall x \in set\ xs.\ x > m)$
$\longrightarrow ((\sum{:}xs) > (m{*}(real\ (length\ xs))))$⟨*proof*⟩

If a is in C then the sum of the collection D where D is C with a removed is the sum of C minus a.

lemma *listsum-rmv1*:
$a \in set\ xs \Longrightarrow \sum{:}(remove1\ a\ xs) = \sum{:}xs - a$
⟨*proof*⟩

A handy addition and division distribution law over collection sums.

lemma *list-sum-distrib-aux*:
shows $(\sum{:}xs/n + \sum{:}xs) = (1 + (1/n)) * \sum{:}xs$
⟨*proof*⟩

lemma *remove1-retains-prod*:
fixes *a*::*real* **and** *xs*::*real list*
shows $a : set\ xs \longrightarrow \prod{:}xs = \prod{:}(remove1\ a\ xs) * a$
(**is** *?P xs*)
⟨*proof*⟩

The final lemma of this section states that if all elements are positive and non-zero then the product of these elements is also positive and non-zero.

lemma *el-gt0-imp-prod-gt0* [*rule-format*]:
fixes *xs*::*real list*
shows $\forall y.\ y : set\ xs \longrightarrow y > 0 \Longrightarrow \prod{:}xs > 0$
⟨*proof*⟩

1.2.2 Auxillary lemma

This section presents a proof of the auxillary lemma required for this theorem.

lemma *prod-exp*:
fixes *x*::*real*
shows $4{*}(x{*}y) = (x{+}y)\hat{}2 - (x{-}y)\hat{}2$
⟨*proof*⟩

lemma *abs-less-imp-sq-less* [*rule-format*]:
fixes *x*::*real* **and** *y*::*real* **and** *z*::*real* **and** *w*::*real*
assumes *diff*: $abs\ (x{-}y) < abs\ (z{-}w)$
shows $(x{-}y)\hat{}2 < (z{-}w)\hat{}2$
⟨*proof*⟩

The required lemma (phrased slightly differently than in the informal proof.) Here we show that for any two pairs of numbers with equal sums the pair with the least difference has the greater product.

lemma *le-diff-imp-gt-prod* [*rule-format*]:
  fixes x::*real* **and** y::*real* **and** z::*real* **and** w::*real*
  assumes *diff*: *abs* $(x-y) <$ *abs* $(z-w)$ **and** *sum*: $x+y = z+w$
  shows $x*y > z*w$
⟨*proof*⟩

1.2.3 Mean and GMean

Now we introduce definitions and properties of arithmetic and geometric means over collections of real numbers.

Definitions

Arithmetic mean

definition
  mean :: (*real list*)⇒*real* **where**
  mean s = ($\sum$:*s* / *real* (*length s*))

Geometric mean

definition
  gmean :: (*real list*)⇒*real* **where**
  gmean s = *root* (*length s*) ($\prod$:*s*)

Properties

Here we present some trival properties of *mean* and *gmean*.

lemma *list-sum-mean*:
  fixes *xs*::*real list*
  shows $\sum$:*xs* = ((*mean xs*) $*$ (*real* (*length xs*)))
⟨*proof*⟩

lemma *list-mean-eq-iff*:
  fixes *one*::*real list* **and** *two*::*real list*
  assumes
    se: ($\sum$:*one* = $\sum$:*two*) **and**
    le: (*length one* = *length two*)
  shows (*mean one* = *mean two*)
⟨*proof*⟩

lemma *list-gmean-gt-iff*:
  fixes *one*::*real list* **and** *two*::*real list*
  assumes
    gz1: $\prod$:*one* > *0* **and** *gz2*: $\prod$:*two* > *0* **and**
    ne1: *one* ≠ [] **and** *ne2*: *two* ≠ [] **and**
    pe: ($\prod$:*one* > $\prod$:*two*) **and**
    le: (*length one* = *length two*)
  shows (*gmean one* > *gmean two*)

⟨*proof*⟩

This slightly more complicated lemma shows that for every non-empty collection with mean M, adding another element a where $a = M$ results in a new list with the same mean M.

lemma *list-mean-cons* [*rule-format*]:
fixes *xs*::*real list*
shows $xs \neq [] \longrightarrow mean\ ((mean\ xs)\#xs) = mean\ xs$
⟨*proof*⟩

For a non-empty collection with positive mean, if we add a positive number to the collection then the mean remains positive.

lemma *mean-gt-0* [*rule-format*]:
$xs{\neq}[] \wedge 0 < x \wedge 0 < (mean\ xs) \longrightarrow 0 < (mean\ (x\#xs))$
⟨*proof*⟩

1.2.4 *list-neq*, *list-eq*

This section presents a useful formalisation of the act of removing all the elements from a collection that are equal (not equal) to a particular value. We use this to extract all the non-mean elements from a collection as is required by the proof.

Definitions

list-neq and *list-eq* just extract elements from a collection that are not equal (or equal) to some value.

abbreviation
list-neq :: $('a\ list) \Rightarrow 'a \Rightarrow ('a\ list)$ **where**
list-neq xs el == *filter* ($\lambda x.\ x{\neq}el$) *xs*

abbreviation
list-eq :: $('a\ list) \Rightarrow 'a \Rightarrow ('a\ list)$ **where**
list-eq xs el == *filter* ($\lambda x.\ x{=}el$) *xs*

Properties

This lemma just proves a required fact about *list-neq*, *remove1* and *length*.

lemma *list-neq-remove1* [*rule-format*]:
shows $a{\neq}m \wedge a : set\ xs$
$\longrightarrow$ *length* (*list-neq* (*remove1 a xs*) *m*) < *length* (*list-neq xs m*)
(**is** *?A xs* $\longrightarrow$ *?B xs* **is** *?P xs*)
⟨*proof*⟩

We now prove some facts about *list-eq*, *list-neq*, length, sum and product.

lemma *list-eq-sum* [*simp*]:

fixes *xs*::*real list*
shows $\sum$:(*list-eq xs m*) = (*m* ∗ (*real* (*length* (*list-eq xs m*))))
⟨*proof*⟩

lemma *list-eq-prod* [*simp*]:
fixes *xs*::*real list*
shows $\prod$:(*list-eq xs m*) = (*m* ˆ (*length* (*list-eq xs m*)))
⟨*proof*⟩

lemma *listsum-split*:
fixes *xs*::*real list*
shows $\sum$:*xs* = ($\sum$:(*list-neq xs m*) + $\sum$:(*list-eq xs m*))
⟨*proof*⟩

lemma *listprod-split*:
fixes *xs*::*real list*
shows $\prod$:*xs* = ($\prod$:(*list-neq xs m*) ∗ $\prod$:(*list-eq xs m*))
⟨*proof*⟩

lemma *listsum-length-split*:
fixes *xs*::*real list*
shows *length xs* = *length* (*list-neq xs m*) + *length* (*list-eq xs m*)
⟨*proof*⟩

1.2.5 Element selection

We now show that given after extracting all the elements not equal to the mean there exists one that is greater then (or less than) the mean.

lemma *pick-one-gt*:
fixes *xs*::*real list* **and** *m*::*real*
defines *m*: *m* ≡ (*mean xs*) **and** *neq*: *noteq* ≡ *list-neq xs m*
assumes *asum*: *noteq*≠[]
shows ∃ *e*. *e* : *set noteq* ∧ *e* > *m*
⟨*proof*⟩

lemma *pick-one-lt*:
fixes *xs*::*real list* **and** *m*::*real*
defines *m*: *m* ≡ (*mean xs*) **and** *neq*: *noteq* ≡ *list-neq xs m*
assumes *asum*: *noteq*≠[]
shows ∃ *e*. *e* : *set noteq* ∧ *e* < *m*
⟨*proof*⟩

1.2.6 Abstract properties

In order to maintain some comprehension of the following proofs we now introduce some properties of collections.

Definitions

het: The heterogeneity of a collection is the number of elements not equal to its mean. A heterogeneity of zero implies the all the elements in the collection are the same (i.e. homogeneous).

definition
  het :: *real list* ⇒ *nat* **where**
  het l = *length* (*list-neq l* (*mean l*))

lemma *het-gt-0-imp-noteq-ne*: *het l* > *0* ⟹ *list-neq l* (*mean l*) ≠ []
  ⟨*proof*⟩

$\gamma-eq$: Two lists are γ-equivalent if and only if they both have the same number of elements and the same arithmetic means.

definition
  γ-*eq* :: ((*real list*)∗(*real list*)) ⇒ *bool* **where**
  γ-*eq a* ⟷ *mean* (*fst a*) = *mean* (*snd a*) ∧ *length* (*fst a*) = *length* (*snd a*)

γ-*eq* is transitive and symmetric.

lemma γ-*eq-sym*: γ-*eq* (*a*,*b*) = γ-*eq* (*b*,*a*)
  ⟨*proof*⟩

lemma γ-*eq-trans*:
  γ-*eq* (*x*,*y*) ⟹ γ-*eq* (*y*,*z*) ⟹ γ-*eq* (*x*,*z*)
  ⟨*proof*⟩

pos: A list is positive if all its elements are greater than 0.

definition
  pos :: *real list* ⇒ *bool* **where**
  pos l ⟷ (*if l*=[] *then False else* ∀ *e*. *e* : *set l* ⟶ *e* > *0*)

lemma *pos-empty* [*simp*]: *pos* [] = *False* ⟨*proof*⟩
lemma *pos-single* [*simp*]: *pos* [*x*] = (*x* > *0*) ⟨*proof*⟩
lemma *pos-imp-ne*: *pos xs* ⟹ *xs*≠[] ⟨*proof*⟩

lemma *pos-cons* [*simp*]:
  xs ≠ [] ⟶ *pos* (*x*#*xs*) =
   (*if* (*x*>*0*) *then pos xs else False*)
  (**is** *?P x xs* **is** *?A xs* ⟶ *?S x xs*)
⟨*proof*⟩

Properties

Here we prove some non-trivial properties of the abstract properties.

Two lemmas regarding *pos*. The first states the removing an element from a positive collection (of more than 1 element) results in a positive collection. The second asserts that the mean of a positive collection is positive.

lemma *pos-imp-rmv-pos*:
assumes (*remove1 a xs*)≠[] *pos xs* **shows** *pos* (*remove1 a xs*)
⟨*proof*⟩

lemma *pos-mean*: *pos xs* ⟹ *mean xs* > *0*
⟨*proof*⟩

We now show that homogeneity of a non-empty collection x implies that its product is equal to (*mean x*) ^(*length x*).

lemma *listprod-het0*:
shows *x*≠[] ∧ *het x* = *0* ⟹ ∏:*x* = (*mean x*) ^ (*length x*)
⟨*proof*⟩

Furthermore we present an important result - that a homogeneous collection has equal geometric and arithmetic means.

lemma *het-base*:
shows *pos x* ∧ *x*≠[] ∧ *het x* = *0* ⟹ *gmean x* = *mean x*
⟨*proof*⟩

1.2.7 Existence of a new collection

We now present the largest and most important proof in this document. Given any positive and non-homogeneous collection of real numbers there exists a new collection that is γ-equivalent, positive, has a strictly lower heterogeneity and a greater geometric mean.

lemma *new-list-gt-gmean*:
fixes *xs*::*real list* **and** *m*::*real*
defines
m: *m* ≡ (*mean xs*) **and**
neq: *noteq* ≡ *list-neq xs m* **and**
eq: *eq* ≡ *list-eq xs m*
assumes *pos-xs*: *pos xs* **and** *het-gt-0*: *het xs* > *0*
shows
∃ *xs'*. *gmean xs'* > *gmean xs* ∧ γ-*eq* (*xs'*,*xs*) ∧
het xs' < *het xs* ∧ *pos xs'*
⟨*proof*⟩

Furthermore we show that for all non-homogeneous positive collections there exists another collection that is γ-equivalent, positive, has a greater geometric mean *and* is homogeneous.

lemma *existence-of-het0* [*rule-format*]:
shows ∀ *x*. *p* = *het x* ∧ *p* > *0* ∧ *pos x* ⟶
(∃ *y*. *gmean y* > *gmean x* ∧ γ-*eq* (*x*,*y*) ∧ *het y* = *0* ∧ *pos y*)
(**is** *?Q p* **is** ∀ *x*. (*?A x p* ⟶ *?S x*))
⟨*proof*⟩

1.2.8 Cauchy's Mean Theorem

We now present the final proof of the theorem. For any positive collection we show that its geometric mean is less than or equal to its arithmetic mean.

theorem *CauchysMeanTheorem*:
  fixes *z*::*real list*
  assumes *pos z*
  shows *gmean z* $\leq$ *mean z*
⟨*proof*⟩

end

Chapter 2

The Cauchy-Schwarz Inequality

theory *CauchySchwarz*
imports *Complex-Main*
begin
⟨*proof*⟩⟨*proof*⟩⟨*proof*⟩⟨*proof*⟩⟨*proof*⟩⟨*proof*⟩

2.1 Abstract

The following document presents a formalised proof of the Cauchy-Schwarz Inequality for the specific case of R^n. The system used is Isabelle/Isar.

Theorem: Take V to be some vector space possessing a norm and inner product, then for all $a, b \in V$ the following inequality holds: $|a \cdot b| \leq \|a\| * \|b\|$. Specifically, in the Real case, the norm is the Euclidean length and the inner product is the standard dot product.

2.2 Formal Proof

2.2.1 Vector, Dot and Norm definitions.

This section presents definitions for a real vector type, a dot product function and a norm function.

Vector

We now define a vector type to be a tuple of (function, length). Where the function is of type *nat* ⇒ *real*. We also define some accessor functions and appropriate notation.

types *vector* = (*nat*⇒*real*) ∗ *nat*

definition
ith :: *vector* ⇒ *nat* ⇒ *real* (((-)_) [*80*,*100*] *100*) **where**
ith v i = *fst v i*

definition
vlen :: *vector* ⇒ *nat* **where**
vlen v = *snd v*

Now to access the second element of some vector v the syntax is v_2.

Dot and Norm

We now define the dot product and norm operations.

definition
dot :: *vector* ⇒ *vector* ⇒ *real* (**infixr** · *60*) **where**
$dot\ a\ b = (\sum j \in \{1..(vlen\ a)\}.\ a_j * b_j)$

definition
norm :: *vector* ⇒ *real* (‖-‖ *100*) **where**
$norm\ v = sqrt\ (\sum j \in \{1..(vlen\ v)\}.\ v_j\ \hat{}2)$

notation (*HTML* **output**)
norm (||-|| *100*)

Another definition of the norm is $\|v\| = sqrt\ (v \cdot v)$. We show that our definition leads to this one.

lemma *norm-dot*:
$\|v\| = sqrt\ (v \cdot v)$
⟨*proof*⟩

A further important property is that the norm is never negative.

lemma *norm-pos*:
$\|v\| \geq 0$
⟨*proof*⟩

We now prove an intermediary lemma regarding double summation.

lemma *double-sum-aux*:
fixes *f*::*nat* ⇒ *real*
shows
$(\sum k \in \{1..n\}.\ (\sum j \in \{1..n\}.\ f\ k * g\ j)) =$
$(\sum k \in \{1..n\}.\ (\sum j \in \{1..n\}.\ (f\ k * g\ j + f\ j * g\ k)\ /\ 2))$
⟨*proof*⟩

The final theorem can now be proven. It is a simple forward proof that uses properties of double summation and the preceding lemma.

theorem *CauchySchwarzReal*:
fixes *x*::*vector*

assumes *vlen x* = *vlen y*
shows $|x \cdot y| \leq \|x\| * \|y\|$
⟨*proof*⟩

end