Lower Semicontinuous Functions

By Bogdan Grechuk

September 19, 2015

Abstract

We define the notions of lower and upper semicontinuity for functions from a metric space to the extended real line. We prove that a function is both lower and upper semicontinuous if and only if it is continuous. We also give several equivalent characterizations of lower semicontinuity. In particular, we prove that a function is lower semicontinuous if and only if its epigraph is a closed set. Also, we introduce the notion of the lower semicontinuous hull of an arbitrary function and prove its basic properties.

Contents

1 Lower semicontinuous functions 1
   1.1 Relative interior in one dimension ......................... 1
   1.2 Lower and upper semicontinuity ............................ 2
   1.3 Epigraphs ............................................... 12
   1.4 Convex Functions ......................................... 21

1 Lower semicontinuous functions

theory Lower-Semicontinuous
imports Multivariate-Analysis
begin

1.1 Relative interior in one dimension

lemma rel-interior-ereal-semiline:
  fixes a :: ereal
  shows rel-interior { y. a ≤ ereal y } = { y. a < ereal y }
proof (cases a)
  case (real r) then show ?thesis
    using rel-interior-real-semiline[of r]
    by (simp add: atLeast-def greaterThan-def)
  next case PInf thus ?thesis using rel-interior-empty by auto
  next case MInf thus ?thesis using rel-interior-uneq2 by auto
qed
lemma closed-ereal-semiline:
  fixes a :: ereal
  shows closed \{ y. a \leq ereal y \}
proof (cases a)
  case (real r)
  then show ?thesis using closed-real-atLeast unfolding atLeast_def by simp
qed auto

lemma ereal-semiline-unique:
  fixes a b :: ereal
  shows \{ y. a \leq ereal y \} = \{ y. b \leq ereal y \} \iff a = b
by (metis mem-Collect-eq ereal-le-real order-antisym)

1.2 Lower and upper semicontinuity

definition lsc-at :: 'a \Rightarrow ('a::topological-space \Rightarrow 'b::order-topology) \Rightarrow bool where
lsc-at x0 f \iff (\forall X. X \supset x0 \land (f \circ X) \supset l \rightarrow f x0 \leq l)
definition usc-at :: 'a \Rightarrow ('a::topological-space \Rightarrow 'b::order-topology) \Rightarrow bool where
usc-at x0 f \iff (\forall X. X \supset x0 \land (f \circ X) \supset l \rightarrow l \leq f x0)

lemma lsc-at-mem:
  assumes lsc-at x0 f
  assumes x \supset x0
  assumes (f o x) \supset A
  shows f x0 = A
  using assms lsc-at-def[of x0 f] by blast

lemma usc-at-mem:
  assumes usc-at x0 f
  assumes x \supset x0
  assumes (f o x) \supset A
  shows f x0 >= A
  using assms usc-at-def[of x0 f] by blast

lemma lsc-at-open:
  fixes f :: 'a::first-countable-topology \Rightarrow 'b::{complete-linorder, linorder-topology}
  shows lsc-at x0 f \iff
  (\forall S. open S \land f x0 \in S \rightarrow (\exists T. open T \land x0 \in T \land (\forall x'\in T. f x' \leq f x0 \rightarrow f x' \in S)))
(is ?lhs \iff ?rhs)
proof
  { assume ~?rhs
from this obtain S where S-def:
  open S \& f x0 : S \& (ALL T. (open T \& x0 : T) \rightarrow (EX x'.T. f x' <= f

2
\( x_0 \land f x' \approx S \) by \texttt{metis}

\texttt{def X} \(= \{x'. f x' \leq f x_0 \land f x' \approx S\} \) hence \(x_0\) \texttt{islimpt} \(X\) unfolding

\texttt{islimpt-def using S-def by auto}

\texttt{from this obtain} \(x\) where \texttt{x-def}: \(\text{ALL} n. x n : X \land x \dashrightarrow \to x_0\)

\texttt{using \texttt{islimpt-sequential[of x0 X]} by auto}

hence \(\sim (f o x) \dashrightarrow \to (f x_0)\) \texttt{unfolding tendsto-explicit using X-def}

\texttt{S-def by auto}

\texttt{from compact-complete-linorder[of o x] obtain l r where r-def: subseq r \& (f o x o r) \dashrightarrow \to l by auto}

\{ \texttt{assume} \(l : S\) \texttt{hence} \(\text{EX} N. \text{ALL} n \geq N. f(x(r n)) : S\)

\texttt{using r-def tendsto-explicit[of f o o o r]\ S-def by auto}

\texttt{hence False using x-def X-def by auto}\}

\texttt{hence l-prop: \(l \sim: S \& \text{l} < f x_0\)

\texttt{using r-def x-def X-def Lim-bounded-creal[of f o o r]\ by auto}

\{ \texttt{assume} \(f x_0 \leq l\) \texttt{hence} \(f x_0 = l\) \texttt{using l-prop by auto}

\texttt{hence False using l-prop S-def by auto}\}

\texttt{hence EX x l. x \dashrightarrow \to x_0 \land (f o x) \dashrightarrow \to l \& \sim (f x_0 \leq l)\}

\texttt{apply(rule-tac \text{exI} x=x o r in \text{exI}) apply(rule-tac \text{x=I} x \leq l in \text{exI})

using \texttt{r-def x-def by (auto simp add: o-assoc lim-subseq)}

\texttt{hence \sim \text{lhs} unfolding lsc-at-def by blast} \}

\texttt{moreover}

\texttt{assume \texttt{?rhs}}

\{ \texttt{fix} \(x A\) \texttt{assume x-def: x \dashrightarrow \to x_0 (f o x) \dashrightarrow \to A\)

\texttt{assume} \(A \sim: f x_0\)

\texttt{from this obtain} \(S V\) \texttt{where SV-def: open S \& open V \& f x_0 : S \& A : V\)

\texttt{& S Int V = {}\}

\texttt{using hausdorff[of f x0 A] by auto}

\texttt{from this obtain} \(T\) \texttt{where T-def: open T \& x_0 : T \& (\text{ALL} x:T. (f x' \leq f x_0 \dashrightarrow \to f x': S))\)

\texttt{using (\text{?rhs} by metis}

\texttt{from this obtain} \(N1\) \texttt{where ALL n \geq N1. x n : T using x-def tendsto-explicit[of x x0] by auto}

\texttt{hence \*: ALL n \geq N1. (f (x n) \leq f x_0 \dashrightarrow \to f(x n) : S) using T-def by auto}

\texttt{from SV-def obtain N2 where ALL n \geq N2. f(x n) : V}

\texttt{using tendsto-explicit[of f o A x-def by auto}

\texttt{hence ALL n \geq (\text{max} N1 N2). \sim (f(x n) \leq f x_0) using SV-def \* by auto}

\texttt{hence ALL n \geq (\text{max} N1 N2). f(x n) \geq f x_0 by auto}

\texttt{hence f x_0 \leq A using Lim-bounded2-ereal[of f o x A max N1 N2 f x0] x-def by auto}

\texttt{hence f x_0 \leq A by auto}

\{ \texttt{hence \?lhs unfolding lsc-at-def by blast} \}

\texttt{ultimately show \?thesis by blast}

\texttt{qed}
lemma lsc-at-open-mem:
  fixes f :: 'a::first-countable-topology ⇒ 'b::{complete-linorder, linorder-topology}
  assumes lsc-at x0 f
  assumes open S & f x0 : S
  obtains T where open T & x0 : T & (ALL x':T. (f x' <= f x0 --> f x': S))
using assms lsc-at-open[of x0 f] by blast

lemma lsc-at-MInfy:
  fixes f :: 'a::topological-space ⇒ ereal
  assumes f x0 = -∞
  shows lsc-at x0 f
unfolding lsc-at-def using assms by auto

lemma lsc-at-PInfy:
  fixes f :: 'a::metric-space ⇒ ereal
  assumes f x0 = ∞
  shows lsc-at x0 f < − continuous (at x0) f
unfolding lsc-at-open continuous-at-open using assms by auto

lemma lsc-at-real:
  fixes f :: 'a::metric-space ⇒ ereal
  assumes |f x0| ≠ ∞
  shows lsc-at x0 f < − (EX T. open T & x0 : T & (∀ y : T. f y > f x0 - e)))
(is ?lhs < − ?rhs)
proof -
  obtain m where m-def: f x0 = ereal m using assms by (cases f x0) auto
  { assume lsc: lsc-at x0 f
    { fix e assume e-def: (e :: ereal)>0
      hence *: f x0 : {f x0 - e <..< f x0 + e} using assms ereal-between by auto
        from this obtain T where T-def: open T & x0 : T & (ALL x':T. f x' <= f x0 --> f x': {f x0 - e <..< f x0 + e})
          apply (subst lsc-at-open-mem[of x0 f {f x0 - e <..< f x0 + e}]) using lsc e-def by auto
        { fix y assume y:T
          { assume f y <= f x0 hence f y > f x0 - e using T-def (y:T) by auto } 
          moreover
          { assume f y > f x0 hence ...>f x0 - e using * by auto }
            ultimately have f y > f x0 - e using not-le by blast
          } hence EX T. open T & x0 : T & (∀ y : T. f y > f x0 - e) using T-def by auto
        } hence ?rhs by auto 
    }
  moreover
  { assume ?rhs
    { fix S assume S-def: open S & f x0 : S
    }
from this obtain \( e \) where \( e \text{-def}: e > 0 \& \{ f x0 - e < f x0 + e \} \subseteq S \)
apply \( \{ \text{subst ereal-open-cont-interval[of f x0]} \} \) using assms by auto
from this obtain \( T \) where \( T \text{-def}: \text{open T} \& x0 : T \& (!y : T. f y > f x0 - e) \)
using \( \langle \text{rhs<rhs>[rule-format, of ereal C]} \rangle \) by auto
\{ fix \( y \) assume \( y \in T \text{ f y <= f x0 } \) hence \( f y < f x0 + e \)
using assms e-def ereal-between[of f x0 e] by auto
hence \( f y : S \) using e-def T-def \( y \in T \) by auto \}
hence \( \text{EX T. open T} \& x0 : T \& (\text{ALL y : T. (f y <= f x0 --> f y : S)}) \)
using T-def by auto
\} hence lsc-at \( x0 \) f using lsc-at-open by auto
\}
ultimately show \( \text{thesis by auto} \)
\}
qed

lemma lsc-at-ereal:
fixes \( f \) :: '\a::metric-space => ereal
shows lsc-at \( x0 \) f \( --> \) \( (\text{ALL C < f(x0)}. \text{EX T. open T} \& x0 : T \& (!y : T. f y > C)) \)
(is ?lhs \( --> \) ?rhs)
proof
\{ assume \( f x0 = -\infty \) hence \( \text{thesis using lsc-at-MInf by auto} \) \}
moreover
\{ assume pinf \( f \) x0 = \( \infty \)
\{ assume lsc: lsc-at \( x0 \) f
\{ fix \( C \) assume \( C < f x0 \)
\text{hence open} \( \{C<..\} \& f x0 : \{C<..\} \) by auto
from this obtain \( T \) where \( T \text{-def}: \text{open T} \& x0 : T \& (\text{ALL y : T. f y : } \{C<..\}) \)
using \( \langle \text{rhs<rhs>[rule-format, of ereal C]} \rangle \) by auto
\}
hence \( \text{thesis by auto} \) \}
moreover
\{ assume \( \text{rhs} \)
\{ fix \( S \) assume \( S \text{-def: open S} \& f x0 : S \)
\text{then obtain} \( C \) where \( C \text{-def: ereal C < f x0 \& \{ereal C<..\} \subseteq S \) using pinf open-PInf by auto
\text{then obtain} \( T \) where \( T \text{-def: open T} \& x0 : T \& (!y : T. f y : S) \)
using \( \langle \text{rhs<rhs>[rule-format, of ereal C]} \rangle \) by auto
\text{hence EX T. open T} \& x0 : T \& (\text{ALL y : T. (f y <= f x0 --> f y : S)}) \)
using T-def by auto
\} hence lsc-at \( x0 \) f using lsc-at-open by auto
\} ultimately have \( \text{thesis by blast} \) \}
moreover
\{ assume \( \text{fin: f x0 ~ = -}\infty \) f x0 ~ = \( \infty \)
\{ assume lsc: lsc-at \( x0 \) f
\{ fix \( C \) assume \( C < f x0 \)
hence \( f x_0 - C > 0 \) using \( \text{fin real-less-minus-iff} \) by \( \text{auto} \)
from this obtain \( T \) where \( T\text{-def}: \text{open} T \land x_0 : T \land (\forall y : T. f x_0 - (f x_0 - C) < f y) \)
\( \text{using lsc-at-real[of } f x_0 \text{] lsc fin by auto} \)
moreover have \( f x_0 - (f x_0 - C) = C \) using \( \text{fin apply (cases } f x_0, \text{ cases } C) \) by \( \text{auto} \)
ultimately have \( \exists T. \text{open} T \land x_0 : T \land (\forall y : T. f x_0 - C < f y) \) by \( \text{auto} \)
moreover have \( ?\text{rhs} \) by \( \text{auto} \)
\} hence \( \text{thesis by blast} \)
ultimately show \( \text{thesis by blast} \)
\text{qed}

lemma \( \text{lst-at-ball} \):
\begin{align*}
\text{fixes } f :: 'a::metric-space \Rightarrow \text{ereal} \\
\text{shows } \text{lsc-at } x_0 f \iff (\forall C < f(x_0). \exists d > 0. \forall y : (\text{ball } x_0 d). C < f(y)) \\
(\text{is } ?\text{lhs} \iff ?\text{rhs}) \\
\end{align*}
\text{proof} -
\begin{align*}
\{ \& \text{assume } lsc : \text{lsc-at } x_0 f \\
\{ \& \text{fix } C :: \text{ereal assume } C < f x_0 \\
\quad \text{from this obtain } T \text{ where } \text{open} T \land x_0 : T \land (\forall y : T. C < f y) \\\n\quad \text{using lsc lsc-at-ereal[of } x_0 f \text{] by auto} \\
\quad \text{hence } \exists d. d > 0 \land (\forall y : (\text{ball } x_0 d). C < f y) \text{ by (force simp add: open-contains-ball)} \\
\} \}
\}
moreover \{ \& \text{assume } ?\text{rhs} \\
\{ \& \text{fix } C :: \text{ereal assume } C < f x_0 \\
\quad \text{from this obtain } d \text{ where } d > 0 \land (\forall y : (\text{ball } x_0 d). C < f y) \text{ using } (?\text{rhs}) \text{ by auto} \\
\quad \text{hence } \exists d \text{ in } ex \exists (\text{simp add: centre-in-ball}) \text{ done} \\
\} \& \text{hence } \text{lsc-at } x_0 f \text{ using } \text{assms lsc-at-ereal[of } x_0 f \text{] by auto} \\
\}
\text{ultimately show } \text{thesis by auto} \\
\text{qed}

lemma \( \text{lst-at-delta} \):
fixes \( f :: 'a::metric-space \Rightarrow \text{ereal} \)
shows \( \text{lsc-at } x_0 \ f \iff (\text{ALL } C < f(x_0)). \text{EX } d > 0. \ \text{!y. dist } x_0 \ y < d \implies C < f(y) \)
(is ?lhs \iff ?rhs)
proof –
have ?rhs \iff (\text{ALL } C < f(x_0)). \text{ALL } y : (\text{ball } x_0 \ d). \ C < f(y)
unfolding \text{ball-def} by \text{auto}
thus ?thesis using \text{lst-at-ball[of } x_0 \ f \text{]} by \text{auto}
qed

\text{lemma lsc-liminf-at:}
fixes \( f :: 'a::metric-space \Rightarrow \text{ereal} \)
shows \( \text{lsc-at } x_0 \ f \iff f(x_0) \leq \text{Liminf (at } x_0 \text{) } f \)
unfolding \text{lst-at-ball le-Liminf-iff eventually-at}
by \text{(intro arg-cong[of } f = \text{All}] \text{ imp-cong refl ext ex-cong)}
\text{(auto simp: dist-commute zero-less-dist-iff)}

\text{lemma lsc-liminf-at-eq:}
fixes \( f :: 'a::metric-space \Rightarrow \text{ereal} \)
shows \( \text{lsc-at } x_0 \ f \iff (f(x_0) = \text{min (f(x_0)) (Liminf (at } x_0 \text{) } f)) \)
by \text{(metis \text{inf-ereal-def le-iff-inf lsc-liminf-at})}

\text{lemma lsc-imp-liminf:}
fixes \( f :: 'a::metric-space \Rightarrow \text{ereal} \)
assumes \( \text{lsc-at } x_0 \ f \)
assumes \( x \dashv \vdash x_0 \)
shows \( f(x_0) = \text{liminf (f o x)} \)
proof (cases \( f(x_0) \))
case \( P \text{Inf} \)
then show ?thesis using \text{assms lsc-at-PInfty[of } x_0 \text{]} \text{lim-imp-Liminf[of - f o x]}\text{continuous-at-sequentially[of } x_0 \ f \text{]} by \text{auto}
next
case \( \text{real } r \)
\{ fix \( e \) assume \( e \text{-def: (e :: ereal)}>0 \)
  from this obtain \( T \) where \( T\text{-def: open } T \& x_0 : T \& (!y : T. f y > f(x_0) - e) \)
  using \text{lsc-at-real[of } f x_0 \text{]} \text{real \text{assms by} auto}
  from this obtain \( N \) where \( N\text{-def: ALL } n >= N. \ x n : T \)
  apply \text{(subst tendsto-obtains-N[of } x_0 \ T}) using \text{assms by} auto
  hence \( \text{ALL } n >= N. f(x_0) - e < (f o x) n \) using \( T\text{-def} \) by auto
  hence \text{liminf (f o x)} >= f(x_0) - e by \text{(intro Liminf-bounded)} \text{(auto simp: eventually-sequentially intro!: ext[of - N])}
  hence \( f(x_0) <= \text{liminf (f o x)} + e \) apply \text{cases e} unfolding \text{ereal-minus-le-iff}
  by \text{auto}
\}
then show ?thesis by \text{(intro \text{ereal-le-epsilon}) auto

7
qed auto

lemma lsc-liminf:
fixes f :: 'a::metric-space => ereal
shows lsc-at x 0 f <--> (\x. x ----> x 0 ---> f x 0 <= liminf (f o x))
(is ?lhs <--> ?rhs)
proof -
{ assume ?rhs
  { fix x A assume x-def: x ----> x 0 (f o x) ----> A
    hence f x 0 <= A using ?rhs; lim-imp-Liminf[of sequentially] by auto
  } hence ?lhs unfolding lsc-at-def by blast
} from this show ?thesis using lsc-imp-liminf by auto
qed

lemma lsc-sequentially:
fixes f :: 'a::metric-space => ereal
shows lsc-at x 0 f <--> (ALL x c. x ----> x 0 & (ALL n. f(x n)<=c) --->
  f(x 0)<=c)
(is ?lhs <--> ?rhs)
proof -
{ assume ?rhs
  { fix x l assume x ----> x 0 (f o x) ----> l
    { assume l = -\infty hence f x 0 <= l by auto }
  moreover
    { assume l = -\infty
      { fix B :: real obtain N where N-def: ALL n>=N. f(x n) <= ereal B
          using Lim-MInf[\of f o x] (\f o x) ----> l by \infty by auto
          def g == (\%n. if n>=N then x n else x N)
          hence g ----> x 0
                      by (intro filterlim-cong[THEN iffD1, OF refl refl - (x ----> x 0:)])
                      (auto simp: eventually-sequentially)
          moreover have ALL n. f(g n) <= ereal B using g-def N-def by auto
          ultimately have f x 0 <= ereal B using ?rhs by auto
      } hence f x 0 = -\infty using ereal-bot by auto
      hence f x 0 <= l by auto }
  moreover
    { assume fin: \| l \| \infty
      { fix e assume e-def: (e :: ereal)>0
          from this obtain N where N-def: ALL n>=N. f(x n) : \{l - e <..< l + e\}
          apply (subst tendsto-obtains-N[\of f o x l \{l - e <..< l + e\}])
          using fin e-def ereal-between (\f o x) ----> l by auto
          def g == (\%n. if n>=N then x n else x N)
          hence g ----> x 0
                      by (intro filterlim-cong[THEN iffD1, OF refl refl - (x ----> x 0:)])
                      (auto simp: eventually-sequentially)
          moreover have ALL n. f(g n) <= l + e using g-def N-def by auto
          ultimately have f x 0 <= l + e using ?rhs by auto
      }
  }
}

8
hence \( f(x_0) < l \) using `ereal-le-epsilon` by `auto`

ultimately have \( f(x_0) < l \) by `blast`

hence `lsc-at x0 f` unfolding `lsc-at-def` by `auto`

moreover

assume `lsc`: `lsc-at x0 f`

fix \( x \) \( c \) assume `xc-def`: \( x \to x_0 \) \& (\( \text{ALL } n. f(x_n) < c \))

hence `liminf (f \circ x) <= c` using `Limsup-bounded[of sequentially f \circ x]` by `auto`

hence `f(x_0) <= c` using `lsc-def lsc-imp-liminf[of x0 f x]` by `auto`

hence `?rhs` by `auto`

ultimately show `?thesis` by `blast`

qed

**Lemma lsc-sequentially-gen:**

fixes \( f :: \alpha :: \text{metric-space} \to \text{ereal} \)

shows `lsc-at x0 f <= (\text{ALL } x \ c \ c_0. x \to x_0 \ & \ c \to c_0 \ & (\text{ALL } n. f(x_n) <= c_n) \to f(x_0) <= c_0)`

(is `?lhs <= ?rhs`)

**Proof**

assume `?rhs`

fix \( x \) \( c \) assume `a`: `x \to x_0` \& (\( \text{ALL } n. f(x_n) <= c_0 \))

def `c` = (%\( n :: \text{nat} \). c_0) hence `c \to c_0` by `auto`

hence `f(x_0) <= c_0` using `?rhs`[rule-format, of \( x \ c \ c_0 \)] using `a c-def` by `auto`

hence `?lhs` using `lsc-sequentially[of x0 f]` by `auto`

moreover

assume `lsc`: `lsc-at x0 f`

fix \( x \) \( c \) \( c_0 \) assume `xc-def`: \( x \to x_0 \) \& \( c \to c_0 \) \& (\( \text{ALL } n. f(x_n) <= c_n \))

hence `liminf (f \circ x) <= c_0` using `Limsup-mono[of sequentially f \circ x \ c]` by `auto`

hence `f(x_0) <= c_0` using `lsc-def lsc-imp-liminf[of x0 f x]` by `auto`

hence `?rhs` by `auto`

ultimately show `?thesis` by `blast`

qed

**Lemma lsc-sequentially-mem:**

fixes \( f :: \alpha :: \text{metric-space} \to \text{ereal} \)

assumes `lsc-at x0 f`

assumes `x \to x_0` \& `c \to c_0`

assumes `\text{ALL } n. f(x_n) <= c_n`

shows `f(x_0) <= c_0` using `lsc-sequentially-gen[of x0 f]` assms by `auto`

**Lemma lsc-uminus:**
fixes \( f :: 'a::metric-space \Rightarrow \text{ereal} \)
shows \( \text{lsc-at} \ x_0 \ (\%x. -f \ x) \leftrightarrow \text{usc-at} \ x_0 \ f \)
proof -
{ assume \( \text{lsc} :: \text{lsc-at} \ x_0 \ (\%x. -f \ x) \)
  { fix \( A \) assume \( x\text{-def}: x \rightarrow x_0 \ (f \circ x) \rightarrow A \)
    hence \( (\%i. -f \ (x \ i)) \rightarrow -A \) using tendsto-uminus-ereal[of \( f \circ x \ A \)] by auto
    hence \( ((\%x. -f \ x) \circ x \ i) \rightarrow -A \) unfolding o-def by auto
    hence \( f \ x_0 \ <= -A \) applying (subst \( \text{lsc-at-mem}[\text{of} \ x_0 \ (\%x. -f \ x) \ x] \)) using
    lsc x-def by auto
    hence \( f \ x_0 \ >= A \) by auto
  }
  hence \( \text{usc-at} \ x_0 \ f \) unfolding usc-at-def by auto
}
moreover 
{ assume \( \text{usc} :: \text{usc-at} \ x_0 \ f \)
  { fix \( x \) assume \( x\text{-def}: x \rightarrow x_0 \ (\%x. -f \ x) \rightarrow A \)
    hence \( (\%i. f \ (x \ i)) \rightarrow A \) unfolding o-def by auto
    hence \( (f \circ x \ i) \rightarrow -A \) unfolding o-def by auto
    hence \( f \ x_0 \ >= A \) applying (subst \( \text{usc-at-mem}[\text{of} \ x_0 \ f \ x] \)) using usc x-def by auto
    hence \( -f \ x_0 \ <= A \) unfolding ereal-uminus-le-reorder by auto
  }
  hence \( \text{lsc-at} \ x_0 \ (\%x. -f \ x) \) unfolding lsc-at-def by auto
}
ultimately show \( \text{thesis by blast} \)
qed

lemma usc-limsup:
fixes \( f :: 'a::metric-space \Rightarrow \text{ereal} \)
shows \( \text{usc-at} \ x_0 \ f \leftrightarrow (\%x. x \rightarrow x_0 \ (\%x. -f \ x) \rightarrow f \ x_0 \ >= \text{limsup} \ (f \circ x)) \)
(is \( \text{lhs} \leftrightarrow \text{rhs} \))
proof -
have \( \text{usc-at} \ x_0 \ f \leftrightarrow (\%x. x \rightarrow x_0 \ (\%x. -f \ x) \rightarrow f \ x_0 \ <= \text{liminf} \ (\%x. -f \ x)) \)
  using lsc-liminf[of \( f \circ x \) \( \text{of} \ x_0 \ (\%x. -f \ x) \)] by auto
moreover 
{ fix \( x \) assume \( x \rightarrow x_0 \)
  have \(-f \ x_0 \ <= \text{limsup} \ (f \circ x) \leftrightarrow -f \ x_0 \ <= \text{liminf} \ (\%x. -f \ x) \circ x \)
    using ereal-liminf-uminus[of \( f \circ x \) \( \text{of} \ x_0 \)] unfolding o-def by auto
  hence \( f \ x_0 \ >= \text{limsup} \ (f \circ x) \leftrightarrow -f \ x_0 \ <= \text{liminf} \ (\%x. -f \ x) \circ x \)
    by auto
}
ultimately show \( \text{thesis by auto} \)
qed

lemma usc-imp-limsup:
fixes \( f :: 'a::metric-space \Rightarrow \text{ereal} \)
assumes \( \text{usc-at} \ x_0 \ f \)
assumes $x \longrightarrow x_0$
show $f x_0 \geq \limsup (f \circ x)$
using assms usc-limsup[of $x_0$ $f$] by auto

lemma usc-limsup-at:
fixes $f :: 'a::{metric-space} \Rightarrow \text{ereal}$
shows $usc\text{-at } x_0 \ f \leq f \ x_0 \geq \limsup (at \ x_0 \ f)$
proof-
  have $usc\text{-at } x_0 \ f \leq lsc\text{-at } x_0 \ (\%x. -(f \ x))$ by (metis lsc-uminus)
  also have $\cdots \leq -(\liminf (at \ x_0 \ (\%x. -(f \ x))))$ by (metis lsc-liminf-at)
  also have $\cdots \leq -((\limsup (at \ x_0 \ f) \ leq f \ x_0)$ by (metis ereal-Liminf-uminus)
  finally show \thesis by auto
qed

lemma continuous-iff-lsc-usc:
fixes $f :: 'a::{metric-space} \Rightarrow \text{ereal}$
shows $\text{continuous \ (at } x_0 \ f \ (lsc\text{-at } x_0 \ f) \& \ (usc\text{-at } x_0 \ f)$
proof-
  assume $a$ \continuous \ (at } x_0 \ f \ f \ x \ x_0 \Longrightarrow f \ x_0 \Longrightarrow \limsup \ (f \ o \ x) \ using \ a \ \text{continuous-imp-tendsto\[of } x_0 \ f \ x \] \ by \ auto
  hence $\liminf \ (f \ o \ x) = f \ x_0 \ & \ limsup \ (f \ o \ x) = f \ x_0 \ using \ lim\text{-imp-Liminf}[of \ sequentially] \ lim\text{-imp-Limsup}[of \ sequentially] \ by \ auto$
  moreover
  \assume \ \assumes \ \assumes \ (lsc\text{-at } x_0 \ f) \ & \ (usc\text{-at } x_0 \ f)$
  \fix \ \fix \ assume \ assume \ x \ \longrightarrow \ x_0 \ f \ x_0 \Longrightarrow \limsup \ (f \ o \ x) \ using \ a \ unfolding \ usc-limsup \ by \ auto
  moreover have $\\cdots \ \leq \liminf \ (f \ o \ x) \ using \ a \ unfolding \ limsup \ by \ auto$
  lsc-limsup \ by \ auto
  ultimately have $\limsup \ (f \ o \ x) = f \ x_0 \ & \ liminf \ (f \ o \ x) = f \ x_0 \ using \ Liminf\le\text{-Limsup}[of \ sequentially] \ f \ o \ x \ by \ auto$
  hence $\liminf \ (f \ o \ x) \ \longrightarrow \ f \ x_0 \ using \ Liminf\text{-eq-Limsup}[of \ sequentially] \ by \ auto$
  hence $\text{continuous \ (at } x_0 \ f \ f \ x_0 \Longrightarrow L \ using \ continuous\text{-at-sequentially}[of } x_0 \ f \] \ by \ auto$
  } \ultimately \ show \ \thesis \ by \ blast
qed

lemma continuous-lsc-compose:
\assumes \ \assumes \ lsc\text{-at} \ (g \ x_0) \ f \ \text{continuous \ (at } x_0) \ g \ f \ x_0 \Longrightarrow \ L \ applying \ subst \ lsc\text{-at-mem}[of \ g \ x_0 \ f \ g \ o \ x \ L]$
using assms continuous-imp-tendsto[of x0 g x] unfolding o-def by auto
}
from this show ?thesis unfolding lsc-at-def by auto
qed

lemma continuous-isCont:
continuous (at x0) f <−> isCont f x0
by (metis continuous-at isCont-def)

lemma isCont-iff-lsc-usc:
fixes f :: 'a::metric-space => ereal
shows isCont f x0 <−> (lsc-at x0 f) & (usc-at x0 f)
by (metis continuous-iff-lsc-usc continuous-isCont)

definition
lsc :: ('a::topological-space => 'b::order-topology) => bool where
lsc f <−> (!x. lsc-at x f)

definition
usc :: ('a::topological-space => 'b::order-topology) => bool where
usc f <−> (!x. usc-at x f)

lemma continuous-UNIV-iff-lsc-usc:
fixes f :: 'a::metric-space => ereal
shows (ALL x. continuous (at x) f) <−> (lsc f) & (usc f)
by (metis continuous-iff-lsc-usc lsc-def usc-def)

1.3 Epigraphs

definition Epigraph S f :: (- => ereal) = \{ xy. fst xy : S & f (fst xy) <= ereal (snd xy)\}

lemma mem-Epigraph: (x, y) : Epigraph S f <−> x : S & f x <= ereal y unfolding Epigraph-def by auto

lemma ereal-closed-levels:
fixes f :: 'a::metric-space => ereal
shows (ALL y. closed {x. f(x)<=y}) <−> (ALL r. closed {x. f(x)<=ereal r})
(is ?lhs <−> ?rhs)
proof —
{ assume ?rhs
  { fix y :: ereal
    { assume y ~ = \infty & y ~ = -\infty hence closed {x. f(x)<=y} using ?rhs by (cases y) auto }
  }
}

12
moreover
{ assume \( y=\infty \) hence \( \text{closed} \{ x. f(x)\leq y \} \) by auto }
moreover
{ assume \( y=-\infty \)
     hence \( \{ x. f(x)\leq y \} = \text{Inter} \{ \{ x. f(x)\leq \text{ereal} \ r \} | r. r : \text{UNIV} \} \) using
ereal-bot by auto
     hence \( \text{closed} \{ x. f(x)\leq y \} \) using closed-Inter ( ?rhs ) by auto
} ultimately have \( \text{closed} \{ x. f(x)\leq y \} \) by blast
} hence \(?lhs\) by auto
} from this show \(?thesis\) by auto
qed

lemma lsc-iff:
fixes \( f :: 'a::\text{metric-space} => \text{ereal} \)
shows \( (\text{lsc} \ f \ <-> (\forall y. \text{closed} \{ x. f(x)\leq y \}) \) & \( (\text{lsc} \ f \ <-> \text{closed} \ (\text{Epigraph UNIV} \ f)) \)\)
proof –
{ assume \( \text{lsc} \ f \)
{ fix \( z \neq z0 \) assume \( a : \text{ALL} \ n. \ z \ n = (x_n, c_n) \) & \( f(x_n)\leq \text{ereal} \ c_n \)
     by auto
     hence \( \text{EX} \ x \ n \ c. \ z \ n = (x_n, c_n) \) & \( f(x_n)\leq \text{ereal} \ c_n \)
     apply (rule-tac \( x=\text{fst} \ (z \ n) \) in \( \text{exI} \)) apply (rule-tac \( x=\text{snd} \ (z \ n) \) in \( \text{exI} \))
     by auto
} from this obtain \( x \ c \) where \( \text{xc-def} : \text{ALL} \ n. \ z \ n = (x_n, c_n) \) & \( f(x_n)\leq \text{ereal} \ c_n \)
apply (rule-tac \( x=\text{fst} \ (z \ n) \) in \( \text{exI} \)) apply (rule-tac \( x=\text{snd} \ (z \ n) \) in \( \text{exI} \))
by auto
} from this obtain \( x \ n \ c \) where \( \text{xc0-def} : z_0 = (x_0, c_0) \) & \( x ----> x_0 \) & \( c ----> c_0 \)
apply (rule-tac \( x=\text{fst} \ (z_0) \) in \( \text{exI} \)) apply (rule-tac \( x=\text{snd} \ (z_0) \) in \( \text{exI} \))
using tendsto-fst[of \( z \ z0 \)] tendsto-snd[of \( z \ z0 \)] by auto
} from this obtain \( x \ n \ c \) where \( \text{xc0-def} : z_0 = (x_0, c_0) \) & \( x ----> x_0 \) & \( c ----> c_0 \)
by auto
hence \( f(x_0)\leq \text{ereal} \ c_0 \) apply (subst lsc-sequentially-mem[of \( z_0, f \ x \ \text{ereal} \ o \ c \ \text{ereal} \ c_0 \)])
using \( (\text{lsc} \ f) \) xc-def unfolding lsc-def unfolding o-def by auto
hence \( \text{closed} : (\text{Epigraph UNIV} \ f) \) unfolding Epigraph-def using xc0-def by auto
} hence \( \text{closed} \ (\text{Epigraph UNIV} \ f) \) by (simp add: closed-sequential-limits)
}
moreover
{ assume \( \text{closed} \ (\text{Epigraph UNIV} \ f) \)
     hence \*: \( \text{ALL} \ x. l. (\text{ALL} \ n. \ f(\text{fst} \ (x_n))) \leq \text{ereal} \ (\text{snd} \ (x_n))) \) & \( x ----> l \)
-->
\( f(\text{fst} \ l) \leq \text{ereal} \ (\text{snd} \ l) \) unfolding Epigraph-def closed-sequential-limits by auto
{ fix \( r :: \text{real} \)
{ fix \( z \neq z0 \) assume \( a : \text{ALL} \ n. \ f(z_n)\leq \text{ereal} \ r \) & \( z ----> z_0 \)
     hence \( f(z_0)\leq \text{ereal} \ r \) using *[rule-format, of (%n. (z_n, r))](z_0, r)]
     tendsto-Pair[of \( z \ z0 \)] by auto
}
} hence \( \text{closed} \{ x. f(x) \leq \text{ereal} \} \) by (simp add: closed-sequential-limits)
} hence \( \text{ALL} \ y. \text{closed} \{ x. f(x) \leq y \} \) using ereal-closed-levels by auto
}

moreover
{ assume \( a: \text{ALL} \ y. \text{closed} \{ x. f(x) \leq y \} \}
{ fix x0

\{ fix x l assume \( x \longrightarrow \longrightarrow x0 \) (f o x) \( \longrightarrow \longrightarrow l \)
\{ assume \( l = \infty \) hence \( f x0 \leq l \) by auto \}
moreover
{ assume \( mi: l = -\infty \)
{ fix \( B :: \text{real} \)
 obtain \( N \) where \( N\text{-def}: \text{ALL} \ n \geq N. \ f(x n) \leq \text{ereal} \ B \)
 using \( mi \langle f o x \rangle \longrightarrow \longrightarrow l \)Lim-MInfty[of f o x] by auto
{ fix \( d \) assume \( (d :: \text{real}) > 0 \)
 from this obtain \( N1 \) where \( N1\text{-def}: \text{ALL} \ n \geq N1. \ \text{dist} (x n) x0 < d \)
 using \( (x \longrightarrow \longrightarrow x0) \) unfolding lim-sequentially by auto
 hence \( \text{EX} \ y. \ \text{dist} y x0 < d \) & \( \& \) \( \{ x. f(x) \leq \text{ereal} \ B \} \)
 apply (rule-tac \( x = x \) (max N N1) in exI) using \( N\text{-def} \) by auto
}

hence \( x0 : \text{closure} \{ x. f(x) \leq \text{ereal} \ B \} \) apply (subst closure-approachable) by auto

hence \( f x0 \leq \text{ereal} \ B \) using \( a \) by auto
}
} hence \( f x0 \leq l \) using ereal-bot[of f x0] by auto
}

moreover
{ assume \( \text{fin: } |l| \neq \infty \)
{ fix \( e \) assume \( e\text{-def}: (e :: \text{ereal}) > 0 \)
 from this obtain \( N \) where \( N\text{-def}: \text{ALL} \ n \geq N. \ f(x n) : \{ l - e \leq \ldots \leq l \}
 + e \} \)
 apply (subst tendsto-approaches-N[of f o x l \{ l - e \leq \ldots \leq l + e \}])
 using \( \text{fin} \ e\text{-def} \)ereal-between \( \langle f o x \rangle \longrightarrow \longrightarrow l \) by auto
 hence \( *: \text{ALL} \ n \geq N. \ \text{x n} : \{ x. f(x) \leq l + e \} \) using \( N\text{-def} \) by auto
{ fix \( d \) assume \( (d :: \text{real}) > 0 \)
 from this obtain \( N1 \) where \( N1\text{-def}: \text{ALL} \ n \geq N1. \ \text{dist} (x n) x0 < d \)
 using \( (x \longrightarrow \longrightarrow x0) \) unfolding lim-sequentially by auto
 hence \( \text{EX} \ y. \ \text{dist} y x0 < d \) & \( \& \) \( \{ x. f(x) \leq l + e \} \)
 apply (rule-tac \( x = x \) (max N N1) in exI) using \( * \) by auto
}

hence \( x0 : \text{closure} \{ x. f(x) \leq l + e \} \) apply (subst closure-approachable) by auto

hence \( f x0 \leq l + e \) using \( a \) by auto
}
} hence \( f x0 \leq l \) using ereal-le-epsilon by auto

} ultimately have \( f x0 \leq l \) by blast

} hence \( \text{lsc-at} x0 f \) unfolding \( \text{lsc-at-def} \) by auto
}
} hence \( \text{lsc} f \) unfolding \( \text{lsc-def} \) by auto

ultimately show \( \text{thesis} \) by auto

qed
definition lsc-hull :: ('a::metric-space => ereal) => ('a::metric-space => ereal)
where
  lsc-hull f = (SOME g. Epigraph UNIV g = closure(Epigraph UNIV f))

lemma epigraph-mono:
  fixes f :: 'a::metric-space => ereal
  shows (x,y):Epigraph UNIV f & y<=z -->(x,z):Epigraph UNIV f
unfolding Epigraph-def apply auto
using ereal-less-eq[of y z] order-trans-rules(23) by blast

lemma closed-epigraph-lines:
  fixes S :: ('a::metric-space * 'b::metric-space) set
  assumes closed S
  shows closed {z. (x, z) : S}
proof
  { fix z assume z-def: z : closure {z. (x, z) : S}
    { fix e :: real assume e>0
      from this obtain y where y-def: (x,y) : S & dist y z < e
      using closure-approachable[of z {z. (x, z) : S}] z-def by auto
      moreover have dist (x,y) (x,z) = dist y z unfolding dist-prod-def by auto
      ultimately have EX s. s : S & dist s (x,z) < e apply(rule-tac x=(x,y) in exf) by auto
      } hence (x,z):S using closed-approachable[of S (x,z)] assms by auto
    } hence closure {z. (x, z) : S} <= {z. (x, z) : S} by blast
from this show ?thesis using closure-subset-eq by auto
qed

lemma mono-epigraph:
  fixes S :: ('a::metric-space * real) set
  assumes mono: ALL x y z. (x,y):S & y<=z -->(x,z):S
  assumes closed S
  shows EX g. ((Epigraph UNIV g) = S)
proof
  { fix x
    have closed {z. (x, z) : S} using (closed S) closed-epigraph-lines by auto
    hence EX a. {z. (x, z) : S} = {z. a <= ereal z} apply (subst mono-closed-ereal)
    using mono by auto
  } from this obtain g where g-def: ALL x. {z. (x, z) : S} = {z. g x <= ereal z}
    by metis
  { fix s
    have s:S <-> (fst s, snd s) : S by auto
    also have ... <-> g(fst s) <= ereal (snd s) using g-def[rule-format, of fst s]
    by blast
    finally have s:S <-> g(fst s) <= ereal (snd s) by auto
  }
hence \((\text{Epigraph UNIV } g) = S\) unfolding \text{Epigraph-def} by auto
from this show \?thesis by auto
qed

lemma \text{lsc-hull-exists}:
\begin{align*}
& \text{fixes } f :: a::\text{metric-space} =\Rightarrow \text{ereal} \\
& \text{shows } \forall g. \text{Epigraph UNIV } g = \text{closure (Epigraph UNIV } f) \\
& \text{proof-} \\
& \{ \text{fix } x \ y \ z \text{ assume } xy: (x,y) : \text{closure (Epigraph UNIV } f) \ & \text{& } y <= z \\
& \{ \text{fix } e :: \text{real assume } e > 0 \\
& \text{hence } EX \ y a : \text{Epigraph UNIV } f. \text{ dist } y a (x, y) < e \\
& \text{using } xy \text{ closure-approachable[of } (x,y) \text{ Epigraph UNIV } f\text{] by auto} \\
& \text{from this obtain } a \ b \text{ where } ab: (a,b) : \text{Epigraph UNIV } f \ & \text{& dist } (a,b) (x,y) < e \text{ by auto} \\
& \text{moreover have } \text{dist } (a,b) (x,y) = \text{dist } (a,b+(z-y)) (x,z) \\
& \text{unfolding } \text{dist-prod-def dist-norm by (simp add: algebra-simps)} \\
& \text{moreover have } (a,b+(z-y)) : \text{Epigraph UNIV } f \text{ apply (subst epigraph-mono[of - b]) using ab xy by auto} \\
& \text{ultimately have } EX \ w : \text{Epigraph UNIV } f. \text{ dist } w (x, z) < e \text{ by auto} \\
& \} \text{ hence } (x,z) : \text{closure (Epigraph UNIV } f) \text{ using } \text{closure-approachable by auto} \\
& \} \text{ hence } \forall x \ y \ z. (x,y) : \text{closure (Epigraph UNIV } f) \ & \text{& } y <= z \Rightarrow (x,z) : \text{closure (Epigraph UNIV } f) \text{ by auto} \\
& \text{from this show } \?thesis \text{ using mono-epigraph[of closure (Epigraph UNIV } f)] by auto} \\
& \text{qed} \\
\end{align*}

lemma \text{epigraph-invertible}:
\begin{align*}
& \text{assumes } \text{Epigraph UNIV } f = \text{Epigraph UNIV } g \\
& \text{shows } f = g \\
& \text{proof-} \\
& \{ \text{fix } x \\
& \{ \text{fix } C \text{ have } f \ x <= \text{ereal } C \Rightarrow (x,C) : \text{Epigraph UNIV } f \text{ unfolding Epigraph-def by auto} \\
& \text{also have } ... \Rightarrow (x,C) : \text{Epigraph UNIV } g \text{ using assms by auto} \\
& \text{also have } ... \Rightarrow g \ x <= \text{ereal } C \text{ unfolding Epigraph-def by auto} \\
& \text{finally have } f \ x <= \text{ereal } C \Rightarrow g \ x <= \text{ereal } C \text{ by auto} \\
& \} \text{ hence } g \ x = f \ x \text{ using ereal-le-real[of } g \ x \ f \ x\text{] ereal-le-real[of } f \ x \ g \ x\text{] by auto} \\
& \} \text{ from this show } \?thesis \text{ by (simp add: ext)} \\
& \text{qed} \\
\end{align*}

lemma \text{lsc-hull-ex-unique}:
\begin{align*}
& \text{fixes } f :: a::\text{metric-space} =\Rightarrow \text{ereal} \\
& \text{shows } \forall g. \text{Epigraph UNIV } g = \text{closure (Epigraph UNIV } f) \\
& \text{using \text{lsc-hull-exists epigraph-invertible by metis} } \\
\end{align*}

lemma epigraph-lsc-hull:
  fixes f :: 'a::metric-space => ereal
  shows Epigraph UNIV (lsc-hull f) = closure(Epigraph UNIV f)
proof -
  have EX g. Epigraph UNIV g = closure (Epigraph UNIV f) using lsc-hull-exists
  by auto
  thus ?thesis unfolding lsc-hull-def
    using some-eq-ex[of (%g. Epigraph UNIV g = closure(Epigraph UNIV f))] by auto
qed

lemma lsc-hull-expl:
  (g = lsc-hull f) <-> (Epigraph UNIV g = closure(Epigraph UNIV f))
proof -
  { assume Epigraph UNIV g = closure(Epigraph UNIV f)
    hence lsc-hull f = g unfolding lsc-hull-def apply (subst some1-equality[of - g])
      using lsc-hull-ex-unique by metis+}
  from this show ?thesis using epigraph-lsc-hull by auto
qed

lemma lsc-lsc-hull: lsc (lsc-hull f)
  using epigraph-lsc-hull[of f] lsc-iff[of lsc-hull f] by auto

lemma epigraph-subset-iff:
  fixes f g :: 'a::metric-space => ereal
  shows Epigraph UNIV f <= Epigraph UNIV g <=> (ALL x. g x <= f x)
proof -
  { assume epi: Epigraph UNIV f <= Epigraph UNIV g
    { fix x
      { fix z assume f x <= ereal z
        hence (x,z):Epigraph UNIV f unfolding Epigraph-def by auto
        hence (x,z):Epigraph UNIV g using epi by auto
        hence g x <= ereal z unfolding Epigraph-def by auto
        } hence g x <= f x apply (subst ereal-le-real) by auto
      }
    }
  moreover
  { assume le: ALL x. g x <= f x
    { fix x y assume (x,y):Epigraph UNIV f
      hence f x <= ereal y unfolding Epigraph-def by auto
      moreover have g x <= f x using le by auto
      ultimately have g x <= ereal y by auto
      hence (x,y):Epigraph UNIV g unfolding Epigraph-def by auto
    }
  }

17
ultimately show \texttt{?thesis} by \texttt{auto}
qed

\textbf{lemma} \texttt{lsc-hull-le}: \texttt{(lsc-hull f) x <= f x}
\textbf{using} \texttt{epigraph-lsc-hull[of f]} \texttt{closure-subset epigraph-subset-iff[of f lsc-hull f]} \textbf{by} \texttt{auto}

\textbf{lemma} \texttt{lsc-hull-greatest}:
\textbf{fixes} \texttt{f g :: 'a::metric-space => ereal}
\textbf{assumes} \texttt{lsc g ALL x. g x <= f x}
\textbf{shows} \texttt{ALL x. g x <= (lsc-hull f) x}
\textbf{proof} –
\textbf{have} \texttt{closure(Epigraph UNIV f) <= Epigraph UNIV g}
\textbf{using} \texttt{lsc-iff epigraph-subset-iff assms} \textbf{by} \texttt{(metis closure-minimal)}
\textbf{from} \texttt{this} \textbf{show} \texttt{?thesis} \textbf{using} \texttt{epigraph-subset-iff lsc-hull-expl} \textbf{by} \texttt{metis}
qed

\textbf{lemma} \texttt{lsc-hull-iff-greatest}:
\textbf{fixes} \texttt{f g :: 'a::metric-space => ereal}
\textbf{shows} \texttt{(g = lsc-hull f)} \texttt{<->} \texttt{(ALL x. g x < f x) & (ALL h. lsc h & (ALL x. h x < f x) --> (ALL x. h x < g x))}
\texttt{(is \ ?lhs <-> \ ?rhs)}
\textbf{proof} –
\{ \textbf{assume} \ ?lhs \textbf{hence} \ ?rhs \textbf{using} \texttt{lsc-lsc-hull lsc-hull-le lsc-hull-greatest by metis} \}
\textbf{moreover}
\{ \textbf{assume} \ ?rhs \}
\{ \textbf{fix} \texttt{x} \textbf{have} \texttt{(lsc-hull f) x <= g x} \textbf{using} \texttt{(？rhs) lsc-lsc-hull lsc-hull-le by metis} \textbf{moreover have} \texttt{(lsc-hull f) x >= g x} \textbf{using} \texttt{(？rhs) lsc-hull-greatest by metis} \textbf{ultimately have} \texttt{(lsc-hull f) x = g x by auto} \}
\textbf{hence} \texttt{？lhs by} \texttt{(simp add: ext)}
\} \textbf{ultimately show} \texttt{？thesis by} \texttt{blast}
qed

\textbf{lemma} \texttt{lsc-hull-mono}:
\textbf{fixes} \texttt{f g :: 'a::metric-space => ereal}
\textbf{assumes} \texttt{ALL x. g x <= f x}
\textbf{shows} \texttt{ALL x. (lsc-hull g) x <= (lsc-hull f) x}
\textbf{proof} –
\{ \textbf{fix} \texttt{x} \textbf{have} \texttt{(lsc-hull g) x <= g x} \textbf{using} \texttt{lsc-hull-le[of g x]} \textbf{by} \texttt{auto} \textbf{also have} \texttt{... <= f x} \textbf{using} \texttt{assms by auto} \textbf{finally have} \texttt{(lsc-hull g) x <= f x} \textbf{by} \texttt{auto} \}
lemma lsc-hull-lsc:
\[ \text{lsc } f \iff (f = \text{lsc-hull } f) \]
using lsc-hull-lsc-hull-liminf-at using auto

lemma lsc-hull-liminf-at:
\[ (x,z) : \text{Epigraph } UNIV \% x. \min(f x) (\text{Liminf } (at x) f) \]
unfolding Epigraph-def min-Liminf-at by auto
\[ \text{fix } e :: \text{real assume } e > 0 \]
\[ \text{hence } e / \sqrt{2} > 0 \text{ using } (e > 0) \text{ by simp} \]
\[ \text{from this obtain } e_1 \text{ where } e_1 < e / \sqrt{2} \text{ & } e_1 > 0 \text{ using dense by auto} \]
\[ \text{hence } (SUP e : \{0 <..\}. \text{INF } y. \text{ball } x e_1 f y) \]
by (auto intro: SUP-upper)
\[ \text{hence } \text{ereal } z >= (\text{INF } y. \text{ball } x e_1 f y) \text{ using } xx-def \text{ by auto} \]
\[ \text{hence } *: ALL y > \text{ereal } z. \text{EX } t : \text{ball } x e_1 f t \]
by (simp add: Bex-def Inf-le-iff-less)
obtain t where t-def: \( t : \text{ball } x e_1 f t \leq \text{ereal } (z+e_1) \)
using e1-def *[rule-format, of ereal [z+e1]] by auto
\[ \text{hence epri: } (t,z+e_1) : \text{Epigraph } UNIV f \]
unfolding Epigraph-def by auto
\[ \text{have dist } x t < e_1 \text{ using } t-def \text{ unfolding ball-def dist-norm by auto} \]
\[ \text{hence } \text{sqrt } (e_1 ^ 2 + \text{dist } x t ^ 2) < e \]
using e1-def apply (subst real-sqrt-sum-squares-less) by auto
moreover have dist (x,z) (t,z+e1) = sqrt (e_1 ^ 2 + dist x t ^ 2)
unfolding dist-prod-def dist-norm by (simp add: algebra-simps)
ultimately have dist (x,z) (t,z+e1) < e by auto
\[ \text{hence } EX y : \text{Epigraph } UNIV f. \text{dist } y (x,z) < e \]
apply (rule-tac x=(t,z+e1) in bexI) apply (simp add: dist-commute) using
epi by auto
\} hence (x,z) : \text{closure } (\text{Epigraph } UNIV f)
using closure-approachable[of (x,z) \text{Epigraph } UNIV f] by auto
\}

moreover
\{ fix x z assume xx-def: (x,z):\text{closure } (\text{Epigraph } UNIV f)
\{ fix e :: \text{real assume } e > 0 \}
from this obtain y where y-def: y:(\text{Epigraph } UNIV f) & dist y (x,z) < e
using closure-approachable[of (x,z) \text{Epigraph } UNIV f] xx-def by metis
have dist (fst y) x <= \text{sqrt } ((\text{dist } (fst y) x) ^ 2 + (\text{dist } (snd y) z) ^ 2)
by (auto intro: real-sqrt-sum-squares-ge1)
also have ...<\ e using y-def unfolding dist-prod-def by (simp add: algebra-simps)
finally have dist (fst y) x < e by auto
hence h1: fst y :: ball x e unfolding ball-def by (simp add: dist-commute)
have dist (snd y) z <= sqrt ((dist (fst y) x) ^ 2 + (dist (snd y) z) ^ 2)
  by (auto intro: real-sqrt-sum-squares-ge2)
also have ... < e using y-def unfolding dist-prod-def by (simp add: algebra-simps)
finally have h2: dist (snd y) z < e by auto
have (INF y:ball x e. f y) <= f(fst y) using h1 by (simp add: INF-lower)
also have ... <= ereal(snd y) using y-def unfolding Epigraph-def by auto
also have ... < ereal(z+e) using h2 unfolding dist-norm by auto
finally have (INF y:ball x e. f y) < ereal(z+e) by auto
} hence \( \forall e>0 \). (INF y:ball x e. f y) < ereal(z+e) by auto

\{ fix e assume (e::real)>0
\{ fix d assume (d::real)>0
  \{ assume d<e
    have (INF y:ball x d. f y) < ereal(z+d) using * (d>0) by auto
    also have ... < ereal(z+e) using * (d<e) by auto
    finally have (INF y:ball x d. f y) < ereal(z+e) by auto
  } moreover
  \{ assume ~(d<e)
    hence ball x e <= ball x d by auto
    hence (INF y:ball x d. f y) <= (INF y:ball x e. f y) apply (subst INF-mono)
    by auto
    also have ... < ereal(z+e) using * (e>0) by auto
    finally have (INF y:ball x d. f y) < ereal(z+e) by auto
  } ultimately have (INF y:ball x d. f y) < ereal(z+e) by blast
  hence (INF y:ball x d. f y) <= ereal(z+e) by auto
  } hence min (f x) (Liminf (at x) f) <= ereal(z+e) unfolding min-Liminf-at
  by (auto intro: SUP-least)
  } hence min (f x) (Liminf (at x) f) <= ereal z using ereal-le-epsilon2 by auto
  hence (x,z):Epigraph UNIV (%x. min (f x) (Liminf (at x) f)) unfolding Epigraph-full by auto
\}
ultimately have Epigraph UNIV (%x. min (f x) (Liminf (at x) f))= closure (Epigraph UNIV f) by auto
hence (%x. min (f x) (Liminf (at x) f)) = lsc-hull f
  using epigraph-invertible epigraph-lsc-hull[of f] by blast
from this show ?thesis by metis
qed

lemma lsc-hull-same-inf:
  fixes f :: 'a::metric-space => ereal
  shows (INF x. lsc-hull f x) = (INF x. f x)
proof-
  \{ fix x
    have (INF x. f x) <= (INF y:ball x 1. f y) apply (subst INF-mono) by auto
    also have ... <= min (f x) (Liminf (at x) f) unfolding min-Liminf-at by (auto

also have ..=(lsc-hull f) x using lsc-hull-liminf[of f] by auto
finally have \((\text{INF } x. f x) \leq (\text{lsc-hull } f) x\) by auto
} hence \((\text{INF } x. f x) \leq (\text{INF } x. \text{lsc-hull } f x)\) apply (subst \(\text{INF-greatest}\)) by auto
moreover have \((\text{INF } x. \text{lsc-hull } f x) \leq (\text{INF } x. f x)\) apply (subst \(\text{INF-mono}\)) using lsc-hull-le by auto
ultimately show \(?\text{thesis}\) by auto
qed

1.4 Convex Functions

definition convex-on :: \(\text{'a::real-vector set => ('}\text{a =>ereal} )\) => bool where
convex-on \(s f \leq\)
\((\text{ALL } x:s. \text{ALL } y:s. \text{ALL } u>=0. \text{ALL } v>=0. u + v = 1\)
\(\longrightarrow f (u * R x + v * R y) \leq \text{ereal } u * f x + \text{ereal } v * f y\)

lemma convex-on-ereal-mem:
assumes \(\text{convex-on } s f\)
assumes \(x:s\)
shows \(f (u * R x + v * R y) \leq \text{ereal } u * f x + \text{ereal } v * f y\)
using assms unfolding convex-on-def by auto

lemma convex-on-oreal-subset: convex-on \(t f \leq\) \(s \leq\) \(t \leq\) convex-on \(s f\)
unfolding convex-on-def by auto

lemma convex-on-oreal-univ: convex-on \(\text{UNIV } f \leq\) \((\text{ALL } S. \text{convex-on } S f)\)
using convex-on-oreal-subset by auto

lemma ereal-pos-setsum-right-distrib:
fakes \(f :: \text{'a =>ereal}\)
assumes \(r>=0 \ r ~\leq \infty\)
shows \(r * \text{setsum } f A = \text{setsum } \%n. r * f n) A\)
proof (cases \(\text{finite } A\))
case True
thus \(?\text{thesis}\)
proof induct
  case empty thus \(?\text{case by simp}\)
next
case (insert \(x A)\) thus \(?\text{case using assms by (simp add: ereal-pos-distrib)}\)
qed
next
case False thus \(?\text{thesis by simp}\)
qed
lemma convex-ereal-add:
fixes f g :: 'a::real-vector => ereal
assumes convex-on s f convex-on s g
shows convex-on s (%x. f x + g x)
proof -
{ fix x y assume x:s y:s moreover
  fix u v ::real assume uv: 0 <= u 0 <= v u + v = 1
  ultimately have f (u *R x + v *R y) <= (ereal u * f x +ereal v * f y) + (ereal u * g x +ereal v * g y)
  using assms unfolding convex-on-def by (auto simp add: ereal-add-mono)
  also have ... = (ereal u * f x +ereal u * g x) + (ereal v * f y +ereal v * g y)
  by (simp add: algebra-simps)
  also have ... = ereal u * (f x + g x) + ereal v * (f y + g y) by auto }
thus thesis unfolding convex-on-def by auto
qed

lemma convex-ereal-cmul:
assumes 0 <= (c::ereal) convex-on s f
shows convex-on s (%x. c * f x)
proof -
{ fix x y assume x:s y:s moreover
  fix u v ::real assume uv: 0 <= u 0 <= v u + v = 1
  ultimately have f (u *R x + v *R y) <= (ereal u * f x +ereal v * f y)
  using assms unfolding convex-on-def by auto
  hence c * f (u *R x + v *R y) <= c * (ereal u * f x +ereal v * f y)
  using assms by (intro ereal-mult-left-mono) auto
  also have ... <= c * (ereal u * f x) + c * (ereal v * f y)
  using assms by (simp add: ereal-le-distrib)
  also have ... = ereal u * (c * f x) + ereal v * (c * f y) by (simp add: algebra-simps)
  finally have c * f (u *R x + v *R y)
  <= ereal u * (c * f x) + ereal v * (c * f y) by auto }
thus thesis unfolding convex-on-def by auto
qed

lemma convex-ereal-max:
fixes f g :: 'a::real-vector => ereal
assumes convex-on s f convex-on s g
shows convex-on s (%x. max (f x) (g x))
proof -
{ fix x y assume x:s y:s moreover
  fix u v ::real assume uv: 0 <= u 0 <= v u + v = 1
  ultimately have max (f (u *R x + v *R y)) (g (u *R x + v *R y))
  <= max (ereal u * f x +ereal v * f y) (ereal u * g x +ereal v * g y)
apply (subst ereal-max-mono) using assms unfolding convex-on-def by auto
also have \ldots \leq \text{ereal}\, u + \text{max}\, (f\, x)\, (g\, x) + \text{ereal}\, v + \text{max}\, (f\, y)\, (g\, y)
apply (subst ereal-max-least)
apply (subst ereal-add-mono) prefer 4 apply (subst ereal-add-mono)
by (subst ereal-mult-left-mono, auto simp add: uv)+
finally have \text{max}\, (f\, (u \ast_R x + v \ast_R y))\, (g\, (u \ast_R x + v \ast_R y)) <= \text{ereal}\, u + \text{max}\, (f\, x)\, (g\, x) + \text{ereal}\, v + \text{max}\, (f\, y)\, (g\, y) by auto 
thus \text{?thesis}\ unfolding convex-on-def by auto
qed

**Lemma** convex-on-ereal-alt:

fixes C :: 'a::real-vector set
assumes convex C
shows convex-on C f =
(\text{ALL } x, y : C. \text{ALL } m : \text{real. } m \geq 0 \& m \leq 1
\rightarrow f\, (m \ast_R x + (1 - m) \ast_R y) \leq (\text{ereal}\, m) \ast f\, x + (1 - (\text{ereal}\, m)) \ast f\, y)
proof safe
fix x y fix m :: real
have[simp]: \text{ereal}\, (1 - m) = (1 - \text{ereal}\, m)
using \text{ereal-minus}(1)\,[\text{of } 1\, m] by (auto simp: one-ereal-def)
assume asms: convex-on C f x : C y : C 0 <= m m <= 1
from this[unfolded convex-on-def, rule-format]
have \text{ALL} u v. ((0 <= u \& \& 0 <= v \& u + v = 1) \rightarrow
f\, (u \ast_R x + v \ast_R y) \leq (\text{ereal}\, u) \ast f\, x + (\text{ereal}\, v) \ast f\, y) by auto
from this[rule-format, of m 1 - m, simplified] asms
show f\, (m \ast_R x + (1 - m) \ast_R y)
\leq (\text{ereal}\, m) \ast f\, x + (1 - \text{ereal}\, m) \ast f\, y by auto
next
assume asm: \text{ALL } x : C. \text{ALL } y : C. \text{ALL } m. 0 <= m \& m <= 1
\rightarrow f\, (m \ast_R x + (1 - m) \ast_R y) \leq (\text{ereal}\, m) \ast f\, x + (1 - \text{ereal}\, m) \ast f\, y
{ fix x y fix u v :: real
assume lasm: x : C y : C u >= 0 v >= 0 u + v = 1
hence[simp]: 1 - u = v \& \text{ereal}\, u = \text{ereal}\, v
using \text{ereal-minus}(1)\,[\text{of } 1\, m] by (auto simp: one-ereal-def)
from asm[rule-format, of x y u]
have f\, (u \ast_R x + v \ast_R y) \leq (\text{ereal}\, u) \ast f\, x + (\text{ereal}\, v) \ast f\, y
using lasm by auto }
thus convex-on C f unfolding convex-on-def by auto
qed

**Lemma** convex-on-ereal-alt-mem:

fixes C :: 'a::real-vector set
assumes convex C
assumes convex-on C f
assumes x : C y : C
assumes (m::real) >= 0 m <= 1

23
shows \( f (m \cdot_R x + (1 - m) \cdot_R y) \leq (\text{ereal } m) \cdot f x + (1 - (\text{ereal } m)) \cdot f y \)

by \( \text{(metis assms convex-on-ereal-alt)} \)

lemma \text{ereal-add-right-mono}: \((a::\text{ereal}) \leq b \Longrightarrow a + c \leq b + c \)

by \( \text{(metis \text{ereal-add-mono order-refl})} \)

lemma \text{convex-on-ereal-setsum-aux}:
assumes \( 1 - a > 0 \)
shows \((1 - \text{ereal } a) \cdot (\text{ereal} \ (c / (1 - a)) \cdot b) = (\text{ereal } c) \cdot b \)
by \( \text{(metis mult.assoc mult.commute eq-divide-eq \text{ereal-minus(1)} \text{assms one-ereal-def less-le times-ereal.simps(1))}} \)

lemma \text{convex-on-ereal-setsum}:
fixes \( a :: 'a \Rightarrow \text{real} \)
fixes \( y :: 'a \Rightarrow 'b::\text{real-vector} \)
fixes \( f :: 'b \Rightarrow \text{ereal} \)
assumes \( \text{finite } s \cdot s \sim = \{\} \)
assumes \( \text{convex-on } C \ f \)
assumes \( \text{convex } C \)
assumes \( (\text{SUM } i : s. \ a i) = 1 \)
assumes \( \text{ALL } i. i : s \Longrightarrow a i >= 0 \)
assumes \( \text{ALL } i. i : s \Longrightarrow y i : C \)
shows \( f (\text{SUM } i : s. \ a i \cdot_R y i) \leq (\text{SUM } i : s. \ \text{ereal } (a i) \cdot f (y i)) \)
using \( \text{assms(1,2,5-)} \)
proof (induct s arbitrary:a rule:finite-ne-induct)
case (singleton i)
hence \( a i: a i = 1 \) by auto
thus case by \( \text{(auto simp: one-ereal-def[symmetric])} \)
next
case (insert i s)
from \( \text{convex-on } C \ f \)
have \( \text{conv: ALL } x \ y \ m. ((x : C & y : C & 0 \leq m & m \leq 1) \Longrightarrow f (m \cdot_R x + (1 - m) \cdot_R y) \leq (\text{ereal } m) \cdot f x + (1 - \text{ereal } m) \cdot f y) \)
using \( \text{convex-on-ereal-alt[of } C \ f \text{]} \langle \text{convex } C \rangle \text{ by auto} \)
\{ assume \( a i = 1 \)
hence \( (\text{SUM } j : s. \ a j) = 0 \)
using insert by auto
hence \( \text{ALL } j. (j : s \Longrightarrow a j = 0) \)
using \( \text{setsum-nonneg-0[where 'b=real] insert by fastforce} \)
hence \( ?\text{case using insert.hyps(1-3) } (a i = 1) \)
by \( \text{(simp add: zero-ereal-def[symmetric] one-ereal-def[symmetric])} \) \}
moreover
\{ assume \( \text{asm: } a i \sim = 1 \)
from insert have \( \text{gai: } y i : C \ a i >= 0 \) by auto
have \( \text{fis: finite (insert i s) using insert by auto} \)
hence \( \text{ail: } a i <= 1 \) using \( \text{setsum-nonneg-leq-bound[of insert i s a] insert by simp} \)
hence $a_i < 1$ using $asm$ by auto

hence $i0: 1 - a_i > 0$ by auto

hence $i1: 1 - $ereal $a_i > 0$ using $ereal-minus(1)$[of $1$ $a_i$]

  by (simp add: zero-ereal-def[symmetric] one-ereal-def[symmetric])

have $i2: 1 - $ereal $a_i ~\leq \infty$ using $ereal-minus(1)$[of $1$]

  by (simp add: zero-ereal-def[symmetric] one-ereal-def[symmetric])

let $?a_j = a_j / (1 - a_i)$

have $a$-nonneg: $\forall j. j \in s \implies 0 \leq a_j / (1 - a_i)$

  using $i0$ insert

    by (metis insert-iff divide-nonneg-pos)

have $(SUM j : insert i s. a j) = 1$ using $insert$ by auto

hence $(SUM j : s. a j) = 1 - a_i$ using $setsum.insert$ insert by fastforce

hence $(SUM j : s. a j) / (1 - a_i) = 1$ using $i0$ by auto

hence $a1: (SUM j : s. ?a_j) = 1$ unfolding $setsum-divide-distrib$ by simp

have $asm: (SUM j : s. ?a_j *R y j) : C$

  using $insert$ convex-setsum[OF $finite$ $s$]

  (convex $C$) $a1$ a-nonneg by auto

have $asm-le: f (SUM j : s. ?a_j *R y j) \leq (SUM j : s. $ereal $($?a_j$) * f (y j))$

  using a-nonneg $a1$ insert by blast

have $f (SUM j : insert i s. a j *R y j) = f ((SUM j : s. a j *R y j) + a_i *R y i)$

  using setsum.insert[of $s$ $i$ $?j$. $a_j$ *R $y j$, $OF$ $finite$ $s$] ($i$ $\sim$: $s$)

    by (auto simp only: add.commute)

  also have $a_i *R y_i$

  using $i0$ by auto

  also have $a_i *R (SUM j : s. (a j * inverse (1 - a_i)) *R (SUM y_i)$

    by (auto simp: algebra-simps)

  also have $f ((1 - a_i) *R (SUM j : s. ?a_j *R y_j) + a_i *R y_i)$

    by (auto simp: divide-inverse)

  also have $<= (1 - $ereal $a_i) * f ((SUM j : s. ?a_j *R y_j)) + ($ereal $a_i) * f (y_i)$

    using $conv$[rule-format], $OF$ $y_i$ ($SUM j : s. ?a_j *R y_j$) $a_i$

    using $gai(1)$ $asm$ $gai(2)$ $a1$ by (auto simp: add.commute)

  also have $<= (1 - $ereal $a_i) * ($SUM j : s. $ereal $($?a_j$) * f (y_j)) + ($ereal $a_i) * f (y_i)$

    using $ereal-add-right-mono[OF $ereal-mult-left-mono$[of $- 1$ $ereal (a_i)$],

      $OF$ $asm-le$ less-imp-le[OF $i1$]], $OF$ ($ereal (a_i) * f (y_i)$)

    by simp

  also have $<= (SUM j : s. ($ereal (a_j) * f ($y_j)) + ($ereal (a_i) * f ($y_i))$

    unfolding $ereal-pos-setsum-right-distrib$[of $1$ $ereal (a_i)$ $?j$. ($ereal (?a_j)$)

      * f ($y_j$), $OF$ less-imp-le[OF $i1$[if]] $i2$] by auto

  also have $<= (SUM j : s. ($ereal (a_j) * f ($y_j)) + ($ereal (a_i) * f ($y_j))$

    using $i0$ convex-on-ereal-setsum-aux by auto

  also have $<= ($ereal (a_i) * f ($y_i) + (SUM j : s. ($ereal (a_j) * f ($y_j))$)

    by (simp add: add.commute)
also have \( \ldots = \text{(SUM } j : \text{insert } i \text{ s. } \text{ereal } (a \ j)) \ast f (y \ j) \) using insert by auto

finally have \( f (\text{SUM } j : \text{insert } i \text{ s. } a \ j \ast_R y \ j) \leq (\text{SUM } j : \text{insert } i \text{ s. } \text{ereal } (a \ j)) \ast f (y \ j) \) by simp 

ultimately show \( ?\text{case} \) by auto

qed

lemma setsum-2: setsum u \{1::nat..2\} = (u 1)+(u 2)

proof
  have \{1::nat..2\} = \{1::nat,2\} by auto
  thus ?thesis by auto

qed

lemma convex-on-ereal-iff :
  assumes convex s
  shows convex-on s f \iff (ALL k u x. (ALL i:{1..k::nat}. 0 \leq u i & x i : s) & setsum u \{1..k\} = 1 --> f (setsum (%i. u i \ast_R x i) \{1..k\}) \leq setsum (%i. (ereal (u i)) \ast f(x i)) \{1..k\} )
  (is ?rhs \iff ?lhs)

proof
  { assume ?rhs
    \{ fix k u x
      have zero: \sim(setsum u \{1..0::nat\} = (1::real)) by auto
      assume (ALL i:{1..k::nat}. (0::real) \leq u i & x i : s)
      moreover assume setsum u \{1..k\} = 1
      moreover hence k \sim= 0 using zero by metis
      ultimately have f (setsum (%i. u i \ast_R x i) \{1..k\}) \leq setsum (%i. (ereal (u i)) \ast f(x i)) \{1..k\}
        using convex-on-ereal-setsum[of \{1..k\} s f u x] using assms (?rhs) by auto
    \}
  
  moreover
  \{ assume ?lhs
    \{ fix x y u v
      assume xy: x:s y:s
      assume uv: u\geq0 v\geq0 u + v = (1::real)
      def xy = (%i::nat. if i=1 then x else y)
      def uv = (%i::nat. if i=1 then u else v)
      have ALL i:{1..2::nat}. (0 \leq uv i) \& (xy i : s) unfolding xy-def uv-def
        using u v by auto
      moreover have setsum uv \{1..2\} = 1 using setsum-2[of uv] unfolding uv-def
        using u v by auto
      moreover have (SUM i = 1..2. uv i \ast_R xy i) = u \ast_R x + v \ast_R y
        using setsum-2[of (%i. uv i \ast_R xy i)] unfolding xy-def uv-def using xy uv by auto
      moreover have (SUM i = 1..2.ereal (uv i) \ast f (xy i)) = ereal u \ast f x + ereal
    \}
  
  qed
\(v \ast f y\)

\begin{verbatim}
using setsum-2[of (%i.ereal (uv i)) f (xy i)] unfolding xy-def uv-def
using xyz uv auto
ultimately have \(\langle u \ast_R x + v \ast_R y \rangle \leq\)ereal \(u \ast f x +\)ereal \(v \ast f y\)
using \(?\text{rhs}[\text{rule-format}, \text{of 2 uv xy}]\) by auto
\}
hence \(?\text{rhs}\) unfolding convex-on-def by auto
\}
ultimately show \(?\text{thesis}\) by blast
qed
\end{verbatim}

lemma convex-Epigraph:
assumes convex \(S\)
shows convex(Epigraph \(S f\)) \(\iff\) convex-on \(S f\)
proof
\{ assume \(\text{rhs}:\) convex(Epigraph \(S f\))
\{
fix \(x, y\) assume \(xy: x:S y:S\)
fix \(u, v::\text{real}\) assume \(uv: 0 \leq u \leq v u + v = 1\)
have \(\langle u \ast_R x + v \ast_R y \rangle \leq\)ereal \(u \ast f x +\)ereal \(v \ast f y\)
proof
\{ assume \(u=0 \mid v=0\) hence \(?\text{thesis}\) using \(uv\) by (auto simp: zero-ereal-def[symmetric]
one-ereal-def[symmetric]) \}
moreover
\{ assume \(f x = \infty \mid f y = \infty\) hence \(?\text{thesis}\) using \(uv\) by (auto simp:
zero-ereal-def[symmetric] one-ereal-def[symmetric]) \}
moreover
\{ assume \(a: f x = -\infty \& f y = -\infty\) \& \(u=0\)
from this obtain \(z\) where \(f y \leq\)ereal \(z\) apply (cases \(f y\)) by auto
hence \(yz: (y, z) :\)Epigraph \(S f\) unfolding Epigraph-def using \(xy\) by auto
\{ fix \(C::\text{real}\)
\have \(\langle x, (1/u)\ast(C - v \ast z)\rangle :\)Epigraph \(S f\) unfolding Epigraph-def
using \(a\) \(xy\) by auto
hence \(\langle u \ast_R x + v \ast_R y, C\rangle :\)Epigraph \(S f\)
using \(\text{rhs}\) convexD[of Epigraph \(S f\) \(\langle x, (1/u)\ast(C - v \ast z)\rangle\) \((y, z)\) \(u v\) \(uv\)
\(yz\) \(a\) by auto
hence \(\langle f (u \ast_R x + v \ast_R y) \leq\)ereal \(C\)\) unfolding Epigraph-def by
\(auto\)
\}
hence \(\langle u \ast_R x + v \ast_R y \rangle = -\infty\) using \(\text{ereal-bot}\) by \(auto\)
hence \(?\text{thesis}\) by \(auto\) \}
moreover
\{ assume \(a: (f x \sim = \infty) \& f y = -\infty\) \& \(v=0\)
from this obtain \(z\) where \(f x \leq\)ereal \(z\) apply (cases \(f x\)) by auto
hence \(xz: (x, z) :\)Epigraph \(S f\) unfolding Epigraph-def using \(xy\) by auto
\{ fix \(C::\text{real}\)
\have \(\langle y, (1/v)\ast(C - u \ast z)\rangle :\)Epigraph \(S f\) unfolding Epigraph-def
using \(a\) \(xy\) by auto
hence \(\langle u \ast_R x + v \ast_R y, C\rangle :\)Epigraph \(S f\)
using \(\text{rhs}\) convexD[of Epigraph \(S f\) \(\langle x, z\rangle\) \((y, (1/v)\ast(C - u \ast z))\) \(u v\) \(uv\)
\(xz\) \(a\) by auto
hence \(\langle f (u \ast_R x + v \ast_R y) \leq\)ereal \(C\)\) unfolding Epigraph-def by
\end{verbatim}
auto

{ assume u: \text{ereal-bot} by auto
hence \( f(u \cdot R x + v \cdot R y) = -\infty \) by auto
}
hence \( \text{thesis} \) by auto
}

moreover
\{
assume \( a: f x \sim = \infty \) & \( f x \sim = -\infty \) & \( f y \sim = \infty \) & \( f y \sim = -\infty \)

from this obtain \( f x f y \) where \( f x = \text{ereal } f x \) & \( f y = \text{ereal } f y \)

apply (cases f x, cases f y) by auto

hence \( (x, f x): \text{Epigraph } S f \) & \( (y, f y): \text{Epigraph } S f \) unfolding \text{Epigraph-def}
using \( xy \) by auto

hence \( \text{thesis} \) unfolding \text{Epigraph-def using } f x y \) by auto
\}
ultimately have \( \text{thesis} \) by \text{blast}
qed
\}
hence \text{convex-on } S f \text{ unfolding conv-ex-on by auto}
\}

moreover
\{
assume \( \text{lhs}: \text{convex-on } S f \)

\{ fix \( x f x f y \) assume \( xy: (x, f x): \text{Epigraph } S f \) \( (y, f y): \text{Epigraph } S f \)

fix \( u v :: \text{real} \) assume \( uv: 0 < u \iff v < u + v = 1 \)

hence \( le: f x <\text{ereal } f x \) & \( f y <\text{ereal } f y \) using \( xy \) unfolding \text{Epigraph-def}
by auto

have \( x:S \) & \( y:S \) using \( xy \) unfolding \text{Epigraph-def by auto}
moreover hence \( \text{inS: } u \cdot R x + v \cdot R y : S \) using \( \text{assms } uv \text{ convexD[of } S \text{]} \) by auto

ultimately have \( f(u \cdot R x + v \cdot R y) <\text{ereal } u \cdot f x + (\text{ereal } v) \cdot f y \)
using \( \text{lhs convex-on-ereal-mem[of } S f x y u v \text{]} \) \( uv \) by auto

also have \( \ldots <\text{ereal } u \cdot f x + (\text{ereal } v) \cdot f y \)
apply (subst \text{ereal-add-mono}) apply (subst \text{ereal-mult-left-mono})
prefer 4 apply (subst \text{ereal-mult-left-mono}) using \( le uv \) by auto

also have \( \ldots = \text{ereal } (u \cdot f x + v \cdot f y) \) by auto

finally have \( (u \cdot R x + v \cdot R y, u \cdot f x + v \cdot f y): \text{Epigraph } S f \)
unfolding \text{Epigraph-def using } \text{inS by auto}
\}
ultimately have \( \text{thesis} \) by \text{auto}
qed

lemma \text{convex-EpigraphI}:
\[ \text{convex-on } s f \implies \text{convex } s \implies \text{convex( } \text{Epigraph } s f \text{)} \]
unfolding \text{convex-Epigraph by auto}

definition
\[ \text{concave-on } :: \text{'}a::\text{real-vector set} \implies ('a \implies \text{ereal}) \implies \text{bool where} \]
\[ \text{concave-on } S f \iff \text{convex-on } S (\text{\%x. } - f x) \]
definition
finite-on :: 'a::real-vector set => ('a => ereal) => bool where
finite-on S f <-> (ALL x:S. (f x = oo & f x = -oo))

definition
affine-on :: 'a::real-vector set => ('a => ereal) => bool where
affine-on S f <-> (convex-on S f & concave-on S f & finite-on S f)

definition
domain (f::=ereal) = {x. f x < oo}

lemma domain-Epigraph-aux:
assumes x = oo
shows EX r. x = ereal r
using assms by (cases x) auto

lemma domain-Epigraph:
domain f = {x. EX y. (x,y):Epigraph UNIV f}
unfolding domain-def Epigraph-def using domain-Epigraph-aux by auto

lemma domain-Epigraph-fst:
domain f = fst ' (Epigraph UNIV f)

proof
{ fix x assume x:domain f
  from this obtain y where (x,y):Epigraph UNIV f using domain-Epigraph[of f] by auto
  moreover have x = fst (x,y) by auto
  ultimately have x:fst ' (Epigraph UNIV f) unfolding image-def by blast
} from this show ?thesis using domain-Epigraph[of f] by auto
qed

lemma convex-on-domain:
convex-on (domain f) f <-> convex-on UNIV f

proof
{ assume lhs: convex-on (domain f) f
  fix x y
  fix u v ::real assume uv: 0 <= u 0 <= v u + v = 1
  have f (u *R x + v *R y) <= ereal u * f x + ereal v * f y
    proof
      { assume f x = oo | f y = oo hence ?thesis using uv by (auto simp: zero-ereal-def[symmetric] one-ereal-def[symmetric]) } }
    moreover
    { assume ~ (f x = oo | f y = oo)
      hence x: domain f & y : domain f unfolding domain-def by auto
      hence ?thesis using lhs unfolding convex-on-def using uv by auto
    } ultimately show ?thesis by blast
  }
hence convex-on UNIV f unfolding convex-on-def by auto
} from this show thesis using convex-on-ereal-subset by auto
qed

lemma convex-on-domain2:
 convex-on (domain f) f <-> (ALL S. convex-on S f)
by (metis convex-on-domain convex-on-ereal-univ)

lemma convex-domain:
 fixes f :: 'a::euclidean-space => ereal
 assumes convex-on UNIV f
 shows convex (domain f)
proof
 have convex (Epigraph UNIV f) using assms convex-Epigraph by auto
 thus thesis unfolding domain-Epigraph-fst
apply (subst convex-linear-image) using fst-linear linear-conv-bounded-linear by auto
qed

lemma infinite-convex-domain-iff:
 fixes f :: 'a::euclidean-space => ereal
 assumes ALL x. (f x = oo | f x = -oo)
 shows convex-on UNIV f <-> convex (domain f)
proof
{ assume dom: convex (domain f)
  { fix x y assume x:domain f y:domain f moreover
    fix u v ::real assume uv: 0 <= u 0 <= v u + v = 1
    ultimately have u *R x + v *R y : domain f
      using dom unfolding convex-def by auto
    hence f(u *R x + v *R y) = -oo
      using assms unfolding domain-def by auto
  } hence convex-on (domain f) f unfolding convex-on-def by auto
  hence convex-on UNIV f by (metis convex-on-domain2)
} thus thesis by (metis convex-domain)
qed

lemma convex-PInf-overflow:
 fixes f :: 'a::euclidean-space => ereal
 assumes convex-on UNIV f convex S
 shows convex-on UNIV (%x. if x:S then (f x) else oo)
proof
 def g == (%x. if x:S then -oo else oo::ereal)
 hence convex-on UNIV g apply (subst infinite-convex-domain-iff)
  using assms unfolding domain-def by auto
moreover have \((\forall x. \text{if } x:S \text{ then } (f x) \text{ else } \infty) = (\forall x. \max (f x) (g x))\)
apply (subst fun-eq-iff) unfolding g-def apply auto
apply (metis ereal-less-eq(2) max.absorb1)
by (metis ereal-less-eq(1) max.absorb2)
ultimately show \(?thesis\) using convex-ereal-max assms by auto
qed

definition \(\text{proper-on} \colon \mathcal{A} \colon \text{real-vector set} \Rightarrow (\mathcal{A} \Rightarrow \text{ereal}) \Rightarrow \text{bool}\) where
\(\text{proper-on } S f \iff (\forall x:S. f x \sim \leq -\infty) \& (\exists x:S. f x \sim \geq \infty)\)

definition \(\text{proper} \colon (a \colon \text{real-vector} \Rightarrow \text{ereal}) \Rightarrow \text{bool}\) where
\(\text{proper } f \iff \text{proper-on } \text{UNIV} f\)

lemma \(\text{proper-iff}:\)
\(\text{proper } f \iff (\forall x. f x \sim \leq -\infty) \& (\exists x. f x \sim \geq \infty)\)

unfolding proper-def proper-on-def by auto

lemma \(\text{improper-iff}:\)
\(\neg(\text{proper } f) \iff (\exists x. f x = -\infty) \lor (\forall x. f x = \infty)\)
using proper-iff by auto

lemma \(\text{ereal-MInf-plus}[simp]: -\infty + x = (if x = \infty \text{ then } \infty \text{ else } -\infty :: \text{ereal})\)
by (cases x) auto

lemma \(\text{convex-improper}:\)
fixes \(f \colon \mathcal{A} \colon \text{euclidean-space} \Rightarrow \text{ereal}\)
assumes \(\text{convex-on } \text{UNIV} f\)
assumes \(\neg(\text{proper } f)\)
shows \(\forall x. \text{rel-interior}(\text{domain } f). f x = -\infty\)
proof
{ assume \(\text{domain } f = \{\}\) hence \(?thesis\) using rel-interior-empty by auto }
moreover
{ assume \(\text{nemp: domain } f \sim = \{\}\) then obtain \(u\) where \(u\)-def: \(f u = -\infty\) using assms improper-iff[of f] unfolding domain-def by auto
hence \(u\)-dom: \(u\)\text{-domain } f unfolding domain-def by auto
{ fix \(x\) assume \(x\)-rel-interior:\(\text{domain } f\) then obtain \(m\) where \(m\)-def: \(m > 1 \& (1 - m) \ast_R u + m \ast_R x : \text{domain } f\)
using convex-rel-interior-iff[of domain f x] nemp convex-domain[of f] assms unfolding domain-def by auto }
\(\text{def } v = 1/m\) hence \(v01: 0 < v \& v < 1\) using \(m\)-def by auto
\(\text{def } y = (1 - m) \ast_R u + m \ast_R x\)
hence \(x = (1 - v) \ast_R u + v \ast_R y\) unfolding \(v\)-def using \(m\)-def by (simp add:
algebra-simps

hence $f x \leq (1 - \text{ereal } v) \cdot f u + \text{ereal } v \cdot f y$

using convex-on-ereal-alt-mem[of UNIV $f y u v$] assms convex-UNIV v01
by (simp add: convex-UNIV add.commute)

moreover have $f y < \infty$ using m-def y-def unfolding domain-def by auto
moreover have $*: 0 < 1 - \text{ereal } v$ using v01

by (metis diff-less-iff (1)ereal-less (2)ereal-minus (1) one-ereal-def)
moreover hence $(1 - \text{ereal } v) \cdot f u = -\infty$ using u-def by auto
ultimately have $f x = -\infty$ using v01 by simp

} hence $\text{thesis}$ by auto

ultimately show $\text{thesis}$ by blast
qed

lemma convex-improper2:
fixes $f :: 'a::euclidean-space => \text{ereal}$
assumes convex-on UNIV $f$
assumes $\sim$(proper $f$)
shows $f x = \infty \mid f x = -\infty \mid x : \text{rel-frontier (domain } f)$
proof -
{ assume a: $\sim(f x = \infty \mid f x = -\infty)$
  hence $x : \text{domain } f$ unfolding domain-def by auto
  hence $x : \text{closure (domain } f)$ using closure-subset by auto
moreover have $x : \sim: \text{rel-interior (domain } f)$ using assms convex-improper a by auto
ultimately have $x : \text{rel-frontier (domain } f)$ unfolding rel-frontier-def by auto
} thus $\text{thesis}$ by auto
qed

lemma convex-lsc-improper:
fixes $f :: 'a::euclidean-space => \text{ereal}$
assumes convex-on UNIV $f$
assumes $\sim$(proper $f$)
assumes lsc $f$
shows $f x = \infty \mid f x = -\infty$
proof -
{ fix $x$ assume $f x = \infty$
  hence lsc-at $x f$ using assms unfolding lsc-def by auto
  have $x : \text{domain } f$ unfolding domain-def using $f x = \infty$ by auto
  hence $x : \text{closure (domain } f)$ using closure-subset by auto
  hence $x : \text{rel-interior (domain } f)$ by (metis assms (1) convex-closure-rel-interior convex-domain)
  { fix $C$ assume $C < f$
    from this obtain $d$ where d-def: $d > 0 \& (\text{ALL } y. \text{dist } x \ y < d \implies C < f y)$
    using lst-at-delta[of $x f$] lsc-at $x f$ by auto
    from this obtain $y$ where y-def: $y : \text{rel-interior (domain } f)$ & dist $y x < d$
    using $x : \text{rel-interior (domain } f)$[by auto
}
32
hence \( f y = -\infty \) by (metis assms(1) assms(2) convex-improper)
moreover have \( C < f y \) using y-def d-def by (simp add: dist-commute)
ultimately have \( \text{False} \) by auto
\}
hence \( f x = -\infty \) by auto
\}
from this show \(?thesis\) by auto
qed

lemma convex-lsc-hull:
fixes \( f :: 'a::euclidean-space \Rightarrow \text{ereal} \)
assumes convex-on UNIV \( f \)
shows convex-on UNIV \((\text{lsc-hull } f)\)
proof–
\{ have \( \text{convex}(\text{Epigraph UNIV } f) \) by (metis assms convex-EpigraphI convex-UNIV)
hence \( \text{convex}(\text{Epigraph UNIV } (\text{lsc-hull } f)) \) by (metis convex-closure epigraph-lsc-hull)
thus \(?thesis\) by (metis convex-Epigraph convex-UNIV)
\}
qed

lemma improper-lsc-hull:
fixes \( f :: 'a::euclidean-space \Rightarrow \text{ereal} \)
assumes \( \sim(\text{proper } f) \)
shows \( \sim(\text{proper } (\text{lsc-hull } f)) \)
proof–
\{ assume \#: \( \text{ALL } x . f x = \infty \)
then have \( \text{lsc } f \)
  by (metis (full-types) UNIV-I lsc-at-open lsc-def open-UNIV)
  with \# have \( \text{ALL } x . (\text{lsc-hull } f) x = \infty \) by (metis lsc-hull-lsc)
\}
hence \( (\text{ALL } x . f x = \infty) \iff (\text{ALL } x . (\text{lsc-hull } f) x = \infty) \)
  by (metis ereal-infty-less-eq(1) lsc-hull-le)
moreover have \( (\text{EX } x . f x = -\infty) \longrightarrow (\text{EX } x . (\text{lsc-hull } f) x = -\infty) \)
  by (metis ereal-infty-less-eq2(2) lsc-hull-le)
ultimately show \(?thesis\) using assms unfolding improper-iff by auto
qed

lemma lsc-hull-convex-improper:
fixes \( f :: 'a::euclidean-space \Rightarrow \text{ereal} \)
assumes convex-on UNIV \( f \)
assumes \( \sim(\text{proper } f) \)
shows \( \text{ALL } x . \text{rel-interior}(\text{domain } f) . (\text{lsc-hull } f) x = f x \)
by (metis assms convex-improper ereal-infty-less-eq(2) lsc-hull-le)

lemma convex-with-rel-open-domain:
fixes \( f :: 'a::euclidean-space \Rightarrow \text{ereal} \)
assumes convex-on UNIV \( f \)
assumes rel-open (domain f)  
shows \((\text{ALL } x. f x > -\infty) \mid (\text{ALL } x. (f x = \infty \mid f x = -\infty))\)  
proof  
{  
assume \(~(\text{ALL } x. f x > -\infty)\)  
hence \(~(\text{proper } f)\) using proper-iff by auto  
hence \text{ALL } x:\text{rel-interior}(\text{domain } f). f x = -\infty by (metis assms(1) convex-improper)  
hence \text{ALL } x:\text{domain } f. f x = -\infty by (metis assms(2) rel-open-def)  
hence \text{ALL } x. (f x = \infty \mid f x = -\infty) unfolding \text{domain-def} by auto  
} 
thus \(?thesis by auto\)  
qed  

lemma convex-with-UNIV-domain:  
  fixes f :: 'a::euclidean-space =>ereal  
  assumes convex-on UNIV f  
  assumes domain f = UNIV  
  shows \((\text{ALL } x. f x > -\infty) \mid (\text{ALL } x. f x = -\infty)\)  
by (metis assms convex-improper ereal-MInfty-lessI proper-iff rel-interior-univ2 UNIV-I)  

lemma rel-interior-Epigraph:  
  fixes f :: 'a::euclidean-space =>ereal  
  assumes convex-on UNIV f  
  shows \((x, z) : rel-interior (\text{Epigraph UNIV } f) \lt\rightarrow (x : rel-interior (\text{domain } f) \& f x < \text{ereal } z)\)  
apply (subst rel-interior-projection[of - (%y. \{z. (y, z) : \text{Epigraph UNIV } f\})])  
apply (metis assms convex-EpigraphI convex-UNIV convex-on-ereal-univ)  
unfolding domain-Epigraph Epigraph-def using rel-interior-ereal-semiline by auto  

lemma rel-interior-EpigraphI:  
  fixes f :: 'a::euclidean-space =>ereal  
  assumes convex-on UNIV f  
  shows \(\text{rel-interior} (\text{Epigraph UNIV } f) = \{(x,z) \mid z \cdot x : \text{rel-interior} (\text{domain } f) \& f x < \text{ereal } z\}\)  
proof  
{  
fix x z  
  have \((x,z) : \text{rel-interior} (\text{Epigraph UNIV } f) \lt\rightarrow (x : \text{rel-interior} (\text{domain } f) \& f x < \text{ereal } z)\)  
    using rel-interior-Epigraph[of f x z] assms by auto  
} 
thus \(?thesis by auto\)  
qed  

lemma convex-less-ri-domain:  
  fixes f :: 'a::euclidean-space =>ereal  
  assumes convex-on UNIV f
assumes $\exists x. f x < a$

shows $\exists x. \text{rel-interior} (\text{domain } f). f x < a$

proof

  def $A == \{(x :: a :: \text{euclidean-space}, m) | x m. \text{ereal } m < a\}$

  obtain $x$ where $f x < a$ using assms by auto

then obtain $z$ where $z$-def: $f x < \text{ereal } z$ & $\text{ereal } z < a$ using $\text{ereal-dense2}$ by auto

hence $(x, z) : A$ & $(x, z) : \text{Epigraph } \text{UNIV } f$ unfolding $A$-def Epigraph-def by auto

hence $A \setminus (\text{Epigraph } \text{UNIV } f) = \emptyset$ unfolding $A$-def Epigraph-def using assms by auto

moreover have open $A$ proof (cases $a$)

  case real hence $A = \{y. \text{inner } (0 :: a, 1) y < \text{real } a\}$ using $A$-def by auto

  thus $?thesis$ using open-halfspace-lt by auto

next case $\text{PInf}$ thus $?thesis$ using $A$-def by auto

next case $\text{MInf}$ thus $?thesis$ using $A$-def by auto

qed

ultimately have $A \setminus (\text{rel-interior } (\text{domain } f)) \sim \emptyset$

using $\text{rel-interior-Epigraph}[of f]$ assms by auto

thus $?thesis$ apply (rule_tac $x = x0$ in bexI) using $A$-def by auto

qed

lemma rel-interior-eq-between:

fixes $S \ T :: (\text{'m :: euclidean-space}) \text{ set}$

assumes convex $S$ convex $T$

shows $(\text{rel-interior } S = \text{rel-interior } T) \iff (\text{rel-interior } S \subseteq T \& T \subseteq \text{closure } S)$

by (metis assms closure-eq-between convex-closure-rel-interior convex-rel-interior-closure)

lemma convex-less-in-riS:

fixes $f :: 'a :: \text{euclidean-space} \Rightarrow \text{ereal}$

assumes convex-on UNIV $f$

assumes convex $S$ rel-interior $S \subseteq \text{domain } f$

assumes $EX x. \text{closure } S. f x < a$

shows $EX x. \text{rel-interior } S. f x < a$

proof

  def $g == (%x. \text{if } x : \text{closure } S \text{ then } (f x) \text{ else } \infty)$

  hence $EX x. g x < a$ using assms by auto

have $\text{convg} : \text{convex-on UNIV } g$ unfolding $g$-def apply (subst convex-PInfty-outside)

using assms convex-closure by auto

hence $* : EX x. \text{rel-interior } (\text{domain } g). g x < a$ apply (subst convex-less-ri-domain)

using assms $g$-def by auto

have convex (domain $g$) by (metis $\text{convg}$ convex-domain)
moreover have \( \text{rel-interior } S \subseteq \text{domain } g \) & \( \text{domain } g \subseteq \text{closure } S \)
using \( \text{g-def assms rel-interior-subset-closure unfolding domain-def by auto} \)
ultimately have \( \text{rel-interior} (\text{domain } g) = \text{rel-interior } S \)
thus ?thesis
by (metis assms (2) rel-interior-eq-between)

qed

lemma convex-less-inS:
fixes \( f : \mathcal{A} : \text{euclidean-space} \Rightarrow \text{ereal} \)
assumes \( \text{convex-on UNIV } f \)
assumes \( \text{convex } S S \subseteq \text{domain } f \)
assumes \( \exists x : \text{closure } S. f x < a \)
shows \( \exists x : S. f x < a \)
using convex-less-in-riS [of \( f \) \( S \) \( a \)] rel-interior-subset [of \( S \)] assms by auto

lemma convex-finite-geq-on-closure:
fixes \( f : \mathcal{A} : \text{euclidean-space} \Rightarrow \text{ereal} \)
assumes \( \text{convex-on UNIV } f \)
assumes \( \text{convex } S \text{ finite-on } S f \)
assumes \( \forall x : S. f x > a \)
shows \( \forall x : \text{closure } S. f x > a \)
proof
have \( S \subseteq \text{domain } f \) using assms unfolding finite-on-def domain-def by auto
{ assume \( \sim (\forall x : \text{closure } S. f x > a) \)
hence \( \exists x : \text{closure } S. f x < a \) by (simp add: not-le) 
hence False using convex-less-inS [of \( f \) \( S \) \( a \)] assms (\( S \subseteq \text{domain } f \)) by auto }
thus ?thesis by auto
qed

lemma lsc-hull-of-convex-determined:
fixes \( f g : \mathcal{A} : \text{euclidean-space} \Rightarrow \text{ereal} \)
assumes \( \text{convex-on UNIV } f \text{ convex-on UNIV } g \)
assumes \( \text{rel-interior } (\text{domain } f) = \text{rel-interior } (\text{domain } g) \)
assumes \( \forall x : \text{rel-interior } (\text{domain } f). f x = g x \)
shows \( \text{lsc-hull } f \equiv \text{lsc-hull } g \)
proof
have \( \text{rel-interior } (\text{Epigraph } \text{UNIV } f) = \text{rel-interior } (\text{Epigraph } \text{UNIV } g) \)
apply (subst rel-interior-EpigraphI, metis assms)+ using assms by auto
hence \( \text{closure } (\text{Epigraph } \text{UNIV } f) = \text{closure } (\text{Epigraph } \text{UNIV } g) \)
by (metis assms convex-EpigraphI convex-UNIV convex-closure-rel-interior)
thus ?thesis by (metis lsc-hull-expl)
qed

lemma domain-lsc-hull-between:

36
fixes \( f :: 'a::euclidean-space => ereal \)
shows \( \text{domain } f \subseteq \text{domain } (\text{lsc-hull } f) \)
  & \( \text{domain } (\text{lsc-hull } f) \subseteq \text{closure } (\text{domain } f) \)
proof-
  \{ \text{fix } x \text{ assume } x:\text{domain } f \}
  \hence x:\text{domain } (\text{lsc-hull } f) \unfolding \text{domain-def} \text{ using lsc-hull-le[of } f \text{ ] by auto} 
  \}
moreover
  \{ \text{fix } x \text{ assume } x:\text{domain } (\text{lsc-hull } f) \}
  \hence \text{min } (f x) (\text{Liminf } \text{at } x f) < \infty \unfolding \text{domain-def} \text{ using lsc-hull-liminf-at[of } f \text{ ] by auto} 
  \text{then obtain } z \text{ where } z\text{-def: } \text{min } (f x) (\text{Liminf } \text{at } x f) < z \& z < \infty \text{ by } \text{(metis dense)} 
  \}
moreover
  \{ \text{fix } x \text{ assume } x:\text{domain } (\text{lsc-hull } f) \}
  \text{unfolding } \text{closure} \unfolding \text{domain-def} \text{ using lsc-hull-between \text{by auto}} 
  \}
ultimately show \(?\text{thesis by auto} \)
qed

lemma domain-vs-domain-lsc-hull:
  \text{fixes } f :: 'a::euclidean-space => ereal
  \text{assumes } \text{convex-on } UNIV f
  \text{shows } \text{rel-interior } (\text{domain } (\text{lsc-hull } f)) = \text{rel-interior } (\text{domain } f)
  \text{& } \text{closure } (\text{domain } (\text{lsc-hull } f)) = \text{closure } (\text{domain } f)
  \text{& } \text{aff-dim } (\text{domain } (\text{lsc-hull } f)) = \text{aff-dim } (\text{domain } f)
proof-
  \text{have convex } (\text{domain } f) \text{ by } \text{(metis assms convex-domain)}
  \text{moreover have convex } (\text{domain } (\text{lsc-hull } f)) \text{ by } \text{(metis assms convex-domain convex-lsc-hull)}
  \text{moreover have rel-interior } (\text{domain } f) \subseteq \text{domain } (\text{lsc-hull } f)
  \text{& } \text{domain } (\text{lsc-hull } f) \subseteq \text{closure } (\text{domain } f)
  \text{by } \text{(metis domain-lsc-hull-between rel-interior-subset subset-trans)}
  \text{ultimately show ?thesis by } \text{(metis closure-eq-between rel-interior-aff-dim rel-interior-eq-between)}
qed

lemma vertical-line-affine:
  \text{fixes } x :: 'a::euclidean-space
  \text{shows affine } \{(x,m::ereal)|m, m:UNIV\}
unfolding affine-def \text{ by } \text{(auto simp add: pth-8)}
lemma lsc-hull-of-convex-agrees-onRI:
  fixes f :: 'a::euclidean-space =>ereal
  assumes convex-on UNIV f
  shows ALL x:rel-interior (domain f). (f x = (lsc-hull f) x)
proof
  have cEpi: convex (Epigraph UNIV f) by (metis assms convex-EpigraphI convex-UNIV)
  { fix x assume x-def: x : rel-interior (domain f)
    hence f x < ∞ unfolding domain-def by auto
    then obtain r where r-def: (x,r):rel-interior (Epigraph UNIV f)
      using assms x-def rel-interior-Epigraph[of f x] by (metis ereal-dense2)
    def M == \{(x,m::real)| m. m:UNIV\}
    hence affine M using vertical-line-affine by auto
    moreover have rel-interior (Epigraph UNIV f) Int M = {} using r-def M-def by auto
    ultimately have *: closure (Epigraph UNIV f) Int M = closure (Epigraph UNIV f Int M)
      using convex-affine-closure-inter[of Epigraph UNIV f M] cEpi by auto
    have Epigraph UNIV f Int M = {x} <-> {m. f x <=ereal m}
      unfolding Epigraph-def M-def by auto
    moreover have closed({x} <-> {m. f x <=ereal m}) apply (subst closed-Times)
      using closed-ereal-semiline by auto
    ultimately have {x} <-> {m. f x <=ereal m} = closure (Epigraph UNIV f)
      Int M
      by (metis * Int-commute closure-closed)
    also have ...=Epigraph UNIV (lsc-hull f) Int M by (metis Int-commute epigraph-lsc-hull)
    also have ...={x} <-> {m. ((lsc-hull f) x) <=ereal m}
      unfolding Epigraph-def M-def by auto
    finally have {m. f x <=ereal m} = {m. lsc-hull f x <=ereal m} by auto
    hence f x = (lsc-hull f) x usingereal-semiline-unique by auto
  } thus ?thesis by auto
qed

lemma lsc-hull-of-convex-agrees-outside:
  fixes f :: 'a::euclidean-space =>ereal
  assumes convex-on UNIV f
  shows ALL x. x ~: closure (domain f) --> (f x = (lsc-hull f) x)
proof
  { fix x assume x-def: x ~: closure (domain f)
    hence x ~: domain (lsc-hull f) using domain-lsc-hull-between by auto
    hence (lsc-hull f) x = ∞ unfolding domain-def by auto
    hence f x = (lsc-hull f) x using lsc-hull-le[of f x] by auto
  } thus ?thesis by auto
qed

lemma lsc-hull-of-convex-agrees:
  fixes f :: 'a::euclidean-space =>ereal
assumes convex-on UNIV f
shows ALL x. (f x = (lsc-hull f) x | x : rel-frontier (domain f))
by (metis DiffI assms lsc-hull-of-convex-agrees-onRI lsc-hull-of-convex-agrees-outside
rel-frontier-def)

lemma lsc-hull-of-proper-convex-proper:
fixes f :: 'a::euclidean-space => ereal
assumes convex-on UNIV f proper f
shows proper (lsc-hull f)
proof
obtain x where x-def: x : rel-interior (domain f) & f x < ∞
  by (metis assms convex-less-ri-domain ereal-less-PInf ty proper-iff)
hence f x = (lsc-hull f) x using lsc-hull-of-convex-agrees[of f] assms
unfolding rel-frontier-def by auto
moreover have f x > −∞ using assms proper-iff by auto
ultimately have (lsc-hull f) x < ∞ & (lsc-hull f) x > −∞ using x-def by auto
thus ?thesis using convex-lsc-hull lsc-hull-of-proper-convex-proper
lsc-lsc-hull[of f] assms convex-lsc-hull[of f] by auto
qed

lemma lsc-hull-of-proper-convex:
fixes f :: 'a::euclidean-space => ereal
assumes convex-on UNIV f proper f
shows lsc (lsc-hull f) & proper (lsc-hull f) & convex-on UNIV (lsc-hull f) &
(ALL x. (f x = (lsc-hull f) x | x : rel-frontier (domain f)))
by (metis assms convex-lsc-hull lsc-hull-of-convex-agrees lsc-hull-of-proper-convex-proper
lsc-lsc-hull)

lemma affine-no-rel-frontier:
fixes S :: ('n::euclidean-space) set
assumes affine S
shows rel-frontier S = {}
unfolding rel-frontier-def using assms affine-closed[of S]

lemma convex-with-affine-domain-is-lsc:
fixes f :: 'a::euclidean-space => ereal
assumes convex-on UNIV f
assumes affine (domain f)
shows lsc f
by (metis assms affine-no-rel-frontier emptyE lsc-def lsc-hull-liminf-at
lsc-hull-of-convex-agrees lsc-liminf-at-eq)

lemma convex-finite-is-lsc:
fixes f :: 'a::euclidean-space =>ereal
assumes convex-on UNIV f
assumes finite-on UNIV f
shows lsc f

proof
  have affine (domain f)
    using assms affine-UNIV unfolding finite-on-def domain-def by auto
  thus ?thesis by (metis assms(1) convex-with-affine-domain-is-lsc)
qed

lemma always-eventually-within:
  (ALL x:S. P x) ==> eventually P (at x within S)
  unfolding eventually-at-filter by auto

lemmaereal-divide-pos:
  assumes (a::ereal)>0 b>0
  shows a/ereal b>0
  by (metis PInfty-eq-infinity assms ereal.simps(2) ereal-less(2) ereal-less-divide-pos ereal-mult-zero)

lemma real-interval-limpt:
  assumes a<b
  shows (b::real) islimpt {a..<b}
proof
  { fix T assume b:T open T
    then obtain e where e-def: e>0 & cball b e <= T using open-contains-cball[of T] by auto
    hence (b-e):cball b e unfolding cball-def dist-norm by auto
    moreover
    { assume a>=b-e hence a:cball b e unfolding cball-def dist-norm using (a<b)
      by auto }
    ultimately have max a (b-e):cball b e
      by (metis max.absorb1 max.absorb2 linear)
    hence max a (b-e):T using e-def by auto
    moreover have max a (b-e):{a..<b} using e-def (a<b) by auto
    ultimately have EX y:{a..<b}. y : T & y ~ = b by auto
  } thus ?thesis unfolding islimpt-def by auto
qed

lemma lsc-hull-of-convex-aux:
  Limsup (at 1 within {0..<1}) (%m.ereal ((1-m)*a+m*b)) <= ereal b
proof
  have nontr: ~trivial-limit (at 1 within {0..<1::ereal})
    apply (subst trivial-limit-within) using real-interval-limpt by auto
  have ((%m.ereal ((1-m)*a+m*b))) --->(1 - 1) * a + 1 * b) (at 1 within
lemma lsc-hull-of-convex:
fixes $f :: \text{a::euclidean-space} \Rightarrow \text{ereal}$
assumes $x :: \text{rel-interior (domain f)}$
shows $((\%m. f((1-m) * f x + m * m y)) \rightarrow (\text{lsc-hull f}) y) \text{ (at 1 within } \{0..<1\})$
proof
let $?g m = f((1-m) * f x + m * m y)$
{ assume $y=x$ hence $?g = (\%m. f y)$ by (simp add: algebra-simps)
  hence $(?g \rightarrow f y) \text{ (at 1 within } \{0..<1\})$ by (simp add: tendsto-const)
moreover have $(\text{lsc-hull f}) y = f y$ by (metis assms lsc-hull-of-convex-agrees-onRI)
ultimately have $\text{thesis by auto}$
}
moreover
{ assume $y=x$
  have aux: $\text{ALL m. y} = ((1-m) * f x + m * m y) = (1-m) * (y-x)$ by (simp add: algebra-simps)
  have $(\text{lsc-hull f}) y = \text{min} (f y) (\text{Liminf (at y f)})$ by (metis lsc-hull-liminf-at)
  also have $\ldots \Rightarrow \text{Liminf (at 1 within } \{0..<1\})$ ?g unfolding min-Liminf-at
unfolding Liminf-within
  apply (subst SUP-mono) apply (rule tac x=n/norm(y-x) in bexI)
  apply (subst INF-mono) apply (rule tac x=(1-m) * f x + m * m y in bexI)
prefer 2
unfolding ball-def dist-norm by (auto simp add: aux y=x: less-divide-eq)
finally have $*: (\text{lsc-hull f}) y \Rightarrow \text{Liminf (at 1 within } \{0..<1\})$ ?g by auto
  { fix $b$ assume $\text{ereal b} = (\text{lsc-hull f}) y$
    hence $y b: (y,b) \text{: closure (Epigraph UNIV f)}$ by (metis epigraph-lsc-hull_mem-Epigraph UNIV-I)
    have $x : \text{domain f}$ by (metis assms(2) rel-interior-subset set-rev-mp)
    hence $f x < \infty$ unfolding domain-def by auto
    then obtain $a$ where $\text{ereal a} > f x$ by (metis ereal-dense2)
    hence $xa: (x,a) : \text{rel-interior (Epigraph UNIV f)}$ by (metis assms rel-interior-Epigraph)
    { fix $m :: \text{real assume 0} < = m \& m < 1$
      hence $(y, b) = (1-m) * ((y, b) - (x, a)) : \text{rel-interior (Epigraph UNIV f)}$
        apply (subst rel-interior-closure-convex-shrink)
        apply (metis assms(1) convex-Epigraph convex-UNIV convex-on-ereal-univ)
        using $yb$ $xa$ by auto
      hence $f (y = (1-m) * (y-x)) < \text{ereal (b-(1-m)*(b-a))}$
        using assms(1) rel-interior-Epigraph by auto
      hence $?g m <= \text{ereal ((1-m)*a+m*b)}$ by (simp add: algebra-simps)
    }
    hence eventually $(\%m. ?g m <= \text{ereal ((1-m)*a+m*b)})$
  }

{0..<1}}
  unfolding $\text{lim-ereal by (intro tendsto-intros)}$
  from $\text{lim-imp-Limsup}[OF nontr this]$ show $\text{thesis by simp}$
qed
(at 1 within \{0..<1\}) apply (subst always-eventually-within) by auto

hence Limsup (at 1 within \{0..<1\}) \(?g \leq Limsup (at 1 within \{0..<1\})

(%m. ereal ((1-ml)\times a+m\times b))

apply (subst Limsup-mono) by auto

also have ... \(<= ereal b using lsc-hull-of-convex-aux by auto

finally have Limsup (at 1 within \{0..<1\}) \(?g \leq ereal b by auto

hence Limsup (at 1 within \{0..<1\}) \(?g \leq (lsc-hull f) y

using ereal-le-real[of (lsc-hull f) y] by auto

moreover have nontr: ~trivial-limit (at (1::real) within \{0..<1\})

apply (subst trivial-limit-within) using real-interval-limpt by auto

moreover hence Liminf (at 1 within \{0..<1\}) \(?g \leq Limsup (at 1 within \{0..<1\}) \(?g

apply (subst Liminf-le-Limsup) by auto

ultimately have Limsup (at 1 within \{0..<1\}) \(?g = (lsc-hull f) y

& Liminf (at 1 within \{0..<1\}) \(?g = (lsc-hull f) y

using * by auto

hence \(?thesis apply (subst Liminf-eq-Limsup) using nontr by auto

} ultimately show \(?thesis by auto

qed

end