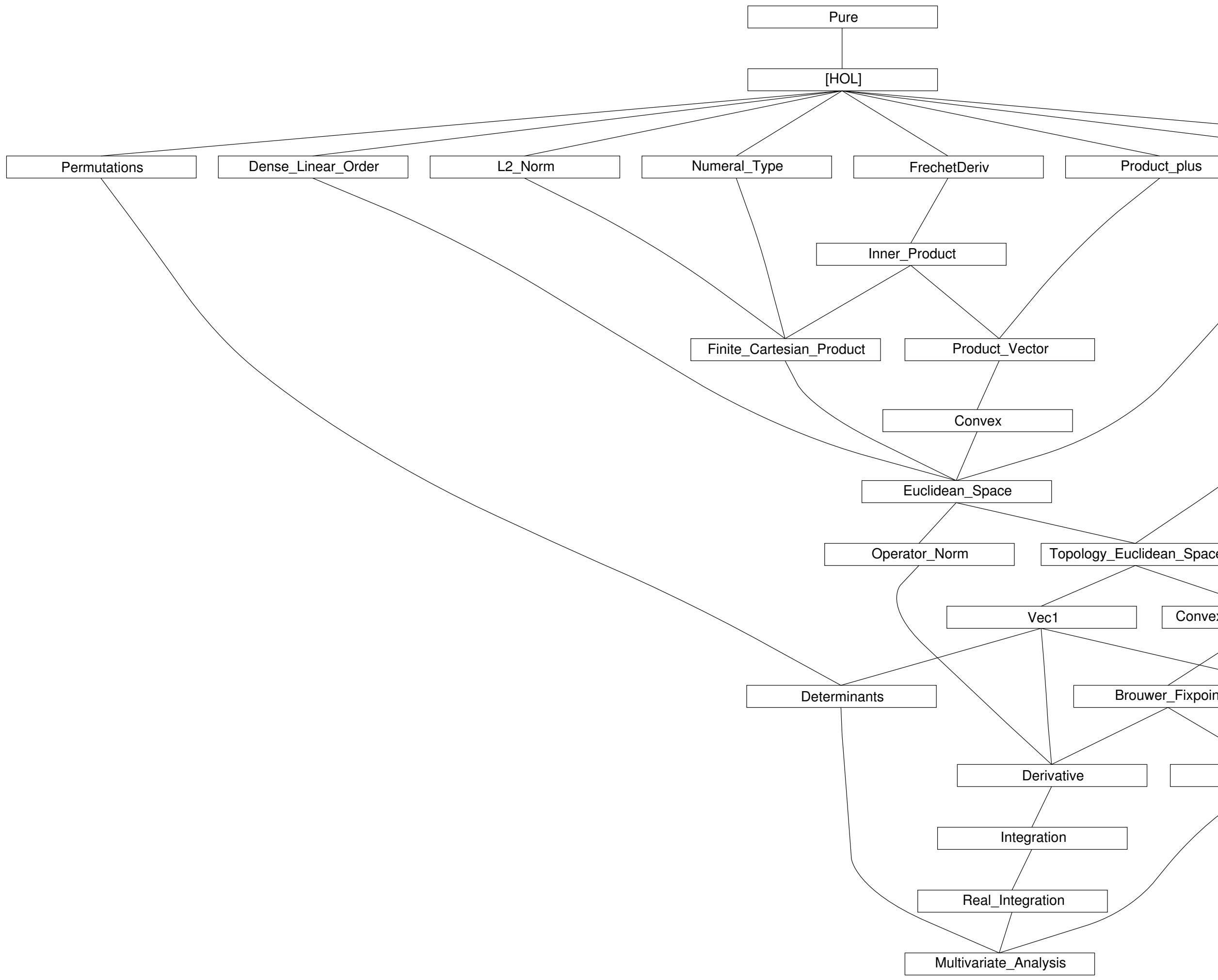
Multivariate Analysis

June 21, 2010

Contents



1 Dense-Linear-Order: Dense linear order without endpoints and a quantifier elimination procedure in Ferrante and Rackoff style

theory *Dense-Linear-Order*
imports *Main*
uses
langford-data.ML
ferrante-rackoff-data.ML
(*langford.ML*)
(*ferrante-rackoff.ML*)
begin

⟨*ML*⟩

context *linorder*
begin

lemma *less-not-permute*[*no-atp*]: $\neg (x < y \wedge y < x)$ ⟨*proof*⟩

lemma *gather-simps*[*no-atp*]:
shows
$(\exists x. (\forall y \in L. y < x) \wedge (\forall y \in U. x < y) \wedge x < u \wedge P\ x) \longleftrightarrow (\exists x. (\forall y \in L. y < x) \wedge (\forall y \in (\mathit{insert}\ u\ U). x < y) \wedge P\ x)$
and $(\exists x. (\forall y \in L. y < x) \wedge (\forall y \in U. x < y) \wedge l < x \wedge P\ x) \longleftrightarrow (\exists x. (\forall y \in (\mathit{insert}\ l\ L). y < x) \wedge (\forall y \in U. x < y) \wedge P\ x)$
$(\exists x. (\forall y \in L. y < x) \wedge (\forall y \in U. x < y) \wedge x < u) \longleftrightarrow (\exists x. (\forall y \in L. y < x) \wedge (\forall y \in (\mathit{insert}\ u\ U). x < y))$
and $(\exists x. (\forall y \in L. y < x) \wedge (\forall y \in U. x < y) \wedge l < x) \longleftrightarrow (\exists x. (\forall y \in (\mathit{insert}\ l\ L). y < x) \wedge (\forall y \in U. x < y))$ ⟨*proof*⟩

lemma
gather-start[*no-atp*]: $(\exists x. P\ x) \equiv (\exists x. (\forall y \in \{\}. y < x) \wedge (\forall y \in \{\}. x < y) \wedge P\ x)$
⟨*proof*⟩

Theorems for $\exists z. \forall x. x < z \longrightarrow (P\ x \longleftrightarrow P_{-\infty})$

lemma *minf-lt*[*no-atp*]: $\exists z . \forall x. x < z \longrightarrow (x < t \longleftrightarrow \mathit{True})$ ⟨*proof*⟩
lemma *minf-gt*[*no-atp*]: $\exists z . \forall x. x < z \longrightarrow (t < x \longleftrightarrow \mathit{False})$
⟨*proof*⟩

lemma *minf-le*[*no-atp*]: $\exists z. \forall x. x < z \longrightarrow (x \leq t \longleftrightarrow \mathit{True})$ ⟨*proof*⟩
lemma *minf-ge*[*no-atp*]: $\exists z. \forall x. x < z \longrightarrow (t \leq x \longleftrightarrow \mathit{False})$
⟨*proof*⟩
lemma *minf-eq*[*no-atp*]: $\exists z. \forall x. x < z \longrightarrow (x = t \longleftrightarrow \mathit{False})$ ⟨*proof*⟩
lemma *minf-neq*[*no-atp*]: $\exists z. \forall x. x < z \longrightarrow (x \neq t \longleftrightarrow \mathit{True})$ ⟨*proof*⟩
lemma *minf-P*[*no-atp*]: $\exists z. \forall x. x < z \longrightarrow (P \longleftrightarrow P)$ ⟨*proof*⟩

Theorems for $\exists z. \forall x. x < z \longrightarrow (P\ x \longleftrightarrow P_{+\infty})$

lemma *pinf-gt*[*no-atp*]: $\exists z . \forall x. z < x \longrightarrow (t < x \longleftrightarrow \mathit{True})$ ⟨*proof*⟩
lemma *pinf-lt*[*no-atp*]: $\exists z . \forall x. z < x \longrightarrow (x < t \longleftrightarrow \mathit{False})$
⟨*proof*⟩

lemma *pinf-ge*[*no-atp*]: $\exists z. \forall x. z < x \longrightarrow (t \leq x \longleftrightarrow \mathit{True})$ ⟨*proof*⟩
lemma *pinf-le*[*no-atp*]: $\exists z. \forall x. z < x \longrightarrow (x \leq t \longleftrightarrow \mathit{False})$
⟨*proof*⟩
lemma *pinf-eq*[*no-atp*]: $\exists z. \forall x. z < x \longrightarrow (x = t \longleftrightarrow \mathit{False})$ ⟨*proof*⟩
lemma *pinf-neq*[*no-atp*]: $\exists z. \forall x. z < x \longrightarrow (x \neq t \longleftrightarrow \mathit{True})$ ⟨*proof*⟩
lemma *pinf-P*[*no-atp*]: $\exists z. \forall x. z < x \longrightarrow (P \longleftrightarrow P)$ ⟨*proof*⟩

lemma *nmi-lt*[*no-atp*]: $t \in U \Longrightarrow \forall x. \neg \mathit{True} \wedge x < t \longrightarrow (\exists u \in U. u \leq x)$
⟨*proof*⟩

lemma *nmi-gt*[*no-atp*]: $t \in U \Longrightarrow \forall x. \neg False \wedge t < x \longrightarrow (\exists u \in U. u \leq x)$
⟨*proof*⟩
lemma *nmi-le*[*no-atp*]: $t \in U \Longrightarrow \forall x. \neg True \wedge x \leq t \longrightarrow (\exists u \in U. u \leq x)$
⟨*proof*⟩
lemma *nmi-ge*[*no-atp*]: $t \in U \Longrightarrow \forall x. \neg False \wedge t \leq x \longrightarrow (\exists u \in U. u \leq x)$
⟨*proof*⟩
lemma *nmi-eq*[*no-atp*]: $t \in U \Longrightarrow \forall x. \neg False \wedge x = t \longrightarrow (\exists u \in U. u \leq x)$
⟨*proof*⟩
lemma *nmi-neq*[*no-atp*]: $t \in U \Longrightarrow \forall x. \neg True \wedge x \neq t \longrightarrow (\exists u \in U. u \leq x)$
⟨*proof*⟩
lemma *nmi-P*[*no-atp*]: $\forall x. \sim P \wedge P \longrightarrow (\exists u \in U. u \leq x)$ ⟨*proof*⟩
lemma *nmi-conj*[*no-atp*]: $[\![\forall x. \neg P1' \wedge P1\ x \longrightarrow (\exists u \in U. u \leq x)$;
$\forall x. \neg P2' \wedge P2\ x \longrightarrow (\exists u \in U. u \leq x)]\!] \Longrightarrow$
$\forall x. \neg(P1' \wedge P2') \wedge (P1\ x \wedge P2\ x) \longrightarrow (\exists u \in U. u \leq x)$ ⟨*proof*⟩
lemma *nmi-disj*[*no-atp*]: $[\![\forall x. \neg P1' \wedge P1\ x \longrightarrow (\exists u \in U. u \leq x)$;
$\forall x. \neg P2' \wedge P2\ x \longrightarrow (\exists u \in U. u \leq x)]\!] \Longrightarrow$
$\forall x. \neg(P1' \vee P2') \wedge (P1\ x \vee P2\ x) \longrightarrow (\exists u \in U. u \leq x)$ ⟨*proof*⟩

lemma *npi-lt*[*no-atp*]: $t \in U \Longrightarrow \forall x. \neg False \wedge x < t \longrightarrow (\exists u \in U. x \leq u)$
⟨*proof*⟩
lemma *npi-gt*[*no-atp*]: $t \in U \Longrightarrow \forall x. \neg True \wedge t < x \longrightarrow (\exists u \in U. x \leq u)$
⟨*proof*⟩
lemma *npi-le*[*no-atp*]: $t \in U \Longrightarrow \forall x. \neg False \wedge x \leq t \longrightarrow (\exists u \in U. x \leq u)$
⟨*proof*⟩
lemma *npi-ge*[*no-atp*]: $t \in U \Longrightarrow \forall x. \neg True \wedge t \leq x \longrightarrow (\exists u \in U. x \leq u)$
⟨*proof*⟩
lemma *npi-eq*[*no-atp*]: $t \in U \Longrightarrow \forall x. \neg False \wedge x = t \longrightarrow (\exists u \in U. x \leq u)$
⟨*proof*⟩
lemma *npi-neq*[*no-atp*]: $t \in U \Longrightarrow \forall x. \neg True \wedge x \neq t \longrightarrow (\exists u \in U. x \leq u)$
⟨*proof*⟩
lemma *npi-P*[*no-atp*]: $\forall x. \sim P \wedge P \longrightarrow (\exists u \in U. x \leq u)$ ⟨*proof*⟩
lemma *npi-conj*[*no-atp*]: $[\![\forall x. \neg P1' \wedge P1\ x \longrightarrow (\exists u \in U. x \leq u)$; $\forall x. \neg P2'$
$\wedge P2\ x \longrightarrow (\exists u \in U. x \leq u)]\!]$
$\Longrightarrow \forall x. \neg(P1' \wedge P2') \wedge (P1\ x \wedge P2\ x) \longrightarrow (\exists u \in U. x \leq u)$ ⟨*proof*⟩
lemma *npi-disj*[*no-atp*]: $[\![\forall x. \neg P1' \wedge P1\ x \longrightarrow (\exists u \in U. x \leq u)$; $\forall x. \neg P2' \wedge$
$P2\ x \longrightarrow (\exists u \in U. x \leq u)]\!]$
$\Longrightarrow \forall x. \neg(P1' \vee P2') \wedge (P1\ x \vee P2\ x) \longrightarrow (\exists u \in U. x \leq u)$ ⟨*proof*⟩

lemma *lin-dense-lt*[*no-atp*]: $t \in U \Longrightarrow \forall x\ l\ u. (\forall t. l < t \wedge t < u \longrightarrow t \notin U) \wedge$
$l < x \wedge x < u \wedge x < t \longrightarrow (\forall y. l < y \wedge y < u \longrightarrow y < t)$
⟨*proof*⟩

lemma *lin-dense-gt*[*no-atp*]: $t \in U \Longrightarrow \forall x\ l\ u. (\forall t. l < t \wedge t < u \longrightarrow t \notin U) \wedge$
$l < x \wedge x < u \wedge t < x \longrightarrow (\forall y. l < y \wedge y < u \longrightarrow t < y)$
⟨*proof*⟩

lemma *lin-dense-le*[*no-atp*]: $t \in U \Longrightarrow \forall x\ l\ u. (\forall t. l < t \wedge t < u \longrightarrow t \notin U) \wedge$
$l < x \wedge x < u \wedge x \leq t \longrightarrow (\forall y. l < y \wedge y < u \longrightarrow y \leq t)$
⟨*proof*⟩

lemma *lin-dense-ge*[*no-atp*]: $t \in U \Longrightarrow \forall x\ l\ u.\ (\forall\ t.\ l < t \wedge t< u \longrightarrow t \notin U) \wedge l< x \wedge x < u \wedge t \leq x \longrightarrow (\forall\ y.\ l < y \wedge y < u \longrightarrow t \leq y)$
⟨*proof*⟩
lemma *lin-dense-eq*[*no-atp*]: $t \in U \Longrightarrow \forall x\ l\ u.\ (\forall\ t.\ l < t \wedge t< u \longrightarrow t \notin U) \wedge l< x \wedge x < u \wedge x = t \quad \longrightarrow (\forall\ y.\ l < y \wedge y < u \longrightarrow y= t)$ ⟨*proof*⟩
lemma *lin-dense-neq*[*no-atp*]: $t \in U \Longrightarrow \forall x\ l\ u.\ (\forall\ t.\ l < t \wedge t< u \longrightarrow t \notin U) \wedge l< x \wedge x < u \wedge x \neq t \quad \longrightarrow (\forall\ y.\ l < y \wedge y < u \longrightarrow y\neq t)$ ⟨*proof*⟩
lemma *lin-dense-P*[*no-atp*]: $\forall x\ l\ u.\ (\forall\ t.\ l < t \wedge t< u \longrightarrow t \notin U) \wedge l< x \wedge x < u \wedge P \quad \longrightarrow (\forall\ y.\ l < y \wedge y < u \longrightarrow P)$ ⟨*proof*⟩

lemma *lin-dense-conj*[*no-atp*]:
$[\![\forall x\ l\ u.\ (\forall\ t.\ l < t \wedge t< u \longrightarrow t \notin U) \wedge l< x \wedge x < u \wedge P1\ x$
$\longrightarrow (\forall\ y.\ l < y \wedge y < u \longrightarrow P1\ y)$;
$\forall x\ l\ u.\ (\forall\ t.\ l < t \wedge t< u \longrightarrow t \notin U) \wedge l< x \wedge x < u \wedge P2\ x$
$\longrightarrow (\forall\ y.\ l < y \wedge y < u \longrightarrow P2\ y)]\!] \Longrightarrow$
$\forall x\ l\ u.\ (\forall\ t.\ l < t \wedge t< u \longrightarrow t \notin U) \wedge l< x \wedge x < u \wedge (P1\ x \wedge P2\ x)$
$\longrightarrow (\forall\ y.\ l < y \wedge y < u \longrightarrow (P1\ y \wedge P2\ y))$
⟨*proof*⟩
lemma *lin-dense-disj*[*no-atp*]:
$[\![\forall x\ l\ u.\ (\forall\ t.\ l < t \wedge t< u \longrightarrow t \notin U) \wedge l< x \wedge x < u \wedge P1\ x$
$\longrightarrow (\forall\ y.\ l < y \wedge y < u \longrightarrow P1\ y)$;
$\forall x\ l\ u.\ (\forall\ t.\ l < t \wedge t< u \longrightarrow t \notin U) \wedge l< x \wedge x < u \wedge P2\ x$
$\longrightarrow (\forall\ y.\ l < y \wedge y < u \longrightarrow P2\ y)]\!] \Longrightarrow$
$\forall x\ l\ u.\ (\forall\ t.\ l < t \wedge t< u \longrightarrow t \notin U) \wedge l< x \wedge x < u \wedge (P1\ x \vee P2\ x)$
$\longrightarrow (\forall\ y.\ l < y \wedge y < u \longrightarrow (P1\ y \vee P2\ y))$
⟨*proof*⟩

lemma *npmibnd*[*no-atp*]: $[\![\forall x.\ \neg\ MP \wedge P\ x \longrightarrow (\exists\ u\in\ U.\ u \leq x); \forall x.\ \neg PP \wedge P\ x \longrightarrow (\exists\ u\in\ U.\ x \leq u)]\!]$
$\Longrightarrow \forall x.\ \neg\ MP \wedge \neg PP \wedge P\ x \longrightarrow (\exists\ u\in\ U.\ \exists\ u' \in U.\ u \leq x \wedge x \leq u')$
⟨*proof*⟩

lemma *finite-set-intervals*[*no-atp*]:
assumes *px*: $P\ x$ **and** *lx*: $l \leq x$ **and** *xu*: $x \leq u$ **and** *linS*: $l\in S$
and *uinS*: $u \in S$ **and** *fS*:*finite* S **and** *lS*: $\forall\ x\in S.\ l \leq x$ **and** *Su*: $\forall\ x\in S.\ x \leq u$
shows $\exists\ a \in S.\ \exists\ b \in S.\ (\forall\ y.\ a < y \wedge y < b \longrightarrow y \notin S) \wedge a \leq x \wedge x \leq b \wedge P\ x$
⟨*proof*⟩

lemma *finite-set-intervals2*[*no-atp*]:
assumes *px*: $P\ x$ **and** *lx*: $l \leq x$ **and** *xu*: $x \leq u$ **and** *linS*: $l\in S$
and *uinS*: $u \in S$ **and** *fS*:*finite* S **and** *lS*: $\forall\ x\in S.\ l \leq x$ **and** *Su*: $\forall\ x\in S.\ x \leq u$
shows $(\exists\ s\in S.\ P\ s) \vee (\exists\ a \in S.\ \exists\ b \in S.\ (\forall\ y.\ a < y \wedge y < b \longrightarrow y \notin S) \wedge a < x \wedge x < b \wedge P\ x)$
⟨*proof*⟩

end

2 The classical QE after Langford for dense linear orders

context *dense-linorder*
begin

lemma *interval-empty-iff*:
$\{y.\ x < y \wedge y < z\} = \{\} \longleftrightarrow \neg\ x < z$
⟨*proof*⟩

lemma *dlo-qe-bnds*[*no-atp*]:
assumes *ne*: $L \neq \{\}$ **and** *neU*: $U \neq \{\}$ **and** *fL*: *finite* L **and** *fU*: *finite* U
shows $(\exists\, x.\ (\forall\, y \in L.\ y < x) \wedge (\forall\, y \in U.\ x < y)) \equiv (\forall\ l \in L.\ \forall\, u \in U.\ l < u)$
⟨*proof*⟩

lemma *dlo-qe-noub*[*no-atp*]:
assumes *ne*: $L \neq \{\}$ **and** *fL*: *finite* L
shows $(\exists\, x.\ (\forall\, y \in L.\ y < x) \wedge (\forall\, y \in \{\}.\ x < y)) \equiv$ *True*
⟨*proof*⟩

lemma *dlo-qe-nolb*[*no-atp*]:
assumes *ne*: $U \neq \{\}$ **and** *fU*: *finite* U
shows $(\exists\, x.\ (\forall\, y \in \{\}.\ y < x) \wedge (\forall\, y \in U.\ x < y)) \equiv$ *True*
⟨*proof*⟩

lemma *exists-neq*[*no-atp*]: $\exists\,(x{::}'a).\ x \neq t$ $\exists\,(x{::}'a).\ t \neq x$
⟨*proof*⟩

lemmas *dlo-simps*[*no-atp*] = *order-refl less-irrefl not-less not-le exists-neq le-less neq-iff linear less-not-permute*

lemma *axiom*[*no-atp*]: *class.dense-linorder* $(op\ \leq)$ $(op\ <)$ ⟨*proof*⟩
lemma *atoms*[*no-atp*]:
shows *TERM* (*less* :: $'a \Rightarrow$ -)
and *TERM* (*less-eq* :: $'a \Rightarrow$ -)
and *TERM* ($op =$ :: $'a \Rightarrow$ -) ⟨*proof*⟩

declare *axiom*[*langford qe*: *dlo-qe-bnds dlo-qe-nolb dlo-qe-noub gather*: *gather-start gather-simps atoms*: *atoms*]
declare *dlo-simps*[*langfordsimp*]

end

lemma *dnf*[*no-atp*]:
$(P\ \&\ (Q\ |\ R)) = ((P \& Q)\ |\ (P \& R))$

$((Q \mid R)\ \&\ P) = ((Q \& P) \mid (R \& P))$
⟨*proof*⟩

lemmas *weak-dnf-simps*[*no-atp*] = *simp-thms dnf*

lemma *nnf-simps*[*no-atp*]:
$(\neg(P \wedge Q)) = (\neg P \vee \neg Q)$ $(\neg(P \vee Q)) = (\neg P \wedge \neg Q)$ $(P \longrightarrow Q) = (\neg P \vee Q)$
$(P = Q) = ((P \wedge Q) \vee (\neg P \wedge \neg\ Q))$ $(\neg\ \neg(P)) = P$
⟨*proof*⟩

lemma *ex-distrib*[*no-atp*]: $(\exists\, x.\ P\ x \vee Q\ x) \longleftrightarrow ((\exists\, x.\ P\ x) \vee (\exists\, x.\ Q\ x))$ ⟨*proof*⟩

lemmas *dnf-simps*[*no-atp*] = *weak-dnf-simps nnf-simps ex-distrib*

⟨*ML*⟩

3 Contructive dense linear orders yield QE for linear arithmetic over ordered Fields

Linear order without upper bounds

locale *linorder-stupid-syntax* = *linorder*
begin
notation
less-eq (*op* ⊑) **and**
less-eq ((-/ ⊑ -) [*51*, *51*] *50*) **and**
less (*op* ⊏) **and**
less ((-/ ⊏ -) [*51*, *51*] *50*)

end

locale *linorder-no-ub* = *linorder-stupid-syntax* +
assumes *gt-ex*: $\exists\, y.\ less\ x\ y$
begin
lemma *ge-ex*[*no-atp*]: $\exists\, y.\ x \sqsubseteq y$ ⟨*proof*⟩

Theorems for $\exists\, z.\ \forall\, x.\ z \sqsubset x \longrightarrow (P\ x \longleftrightarrow P_{+\infty})$

lemma *pinf-conj*[*no-atp*]:
assumes *ex1*: $\exists\, z1.\ \forall\, x.\ z1 \sqsubset x \longrightarrow (P1\ x \longleftrightarrow P1')$
and *ex2*: $\exists\, z2.\ \forall\, x.\ z2 \sqsubset x \longrightarrow (P2\ x \longleftrightarrow P2')$
shows $\exists\, z.\ \forall\, x.\ z \sqsubset x \longrightarrow ((P1\ x \wedge P2\ x) \longleftrightarrow (P1' \wedge P2'))$
⟨*proof*⟩

lemma *pinf-disj*[*no-atp*]:
assumes *ex1*: $\exists\, z1.\ \forall\, x.\ z1 \sqsubset x \longrightarrow (P1\ x \longleftrightarrow P1')$
and *ex2*: $\exists\, z2.\ \forall\, x.\ z2 \sqsubset x \longrightarrow (P2\ x \longleftrightarrow P2')$
shows $\exists\, z.\ \forall\, x.\ z \sqsubset x \longrightarrow ((P1\ x \vee P2\ x) \longleftrightarrow (P1' \vee P2'))$
⟨*proof*⟩

lemma *pinf-ex*[*no-atp*]: **assumes** *ex*:$\exists z.\ \forall x.\ z \sqsubset x \longrightarrow (P\ x \longleftrightarrow P1)$ **and** *p1*: $P1$ **shows** $\exists\ x.\ P\ x$
⟨*proof*⟩

end

Linear order without upper bounds

locale *linorder-no-lb* = *linorder-stupid-syntax* +
  assumes *lt-ex*: $\exists y.\ less\ y\ x$
begin
lemma *le-ex*[*no-atp*]: $\exists y.\ y \sqsubseteq x$ ⟨*proof*⟩

Theorems for $\exists z.\ \forall x.\ x \sqsubset z \longrightarrow (P\ x \longleftrightarrow P_{-\infty})$

lemma *minf-conj*[*no-atp*]:
  assumes *ex1*: $\exists z1.\ \forall x.\ x \sqsubset z1 \longrightarrow (P1\ x \longleftrightarrow P1')$
  and *ex2*: $\exists z2.\ \forall x.\ x \sqsubset z2 \longrightarrow (P2\ x \longleftrightarrow P2')$
  shows $\exists z.\ \forall x.\ x \sqsubset\ z \longrightarrow ((P1\ x \wedge P2\ x) \longleftrightarrow (P1' \wedge P2'))$
⟨*proof*⟩

lemma *minf-disj*[*no-atp*]:
  assumes *ex1*: $\exists z1.\ \forall x.\ x \sqsubset z1 \longrightarrow (P1\ x \longleftrightarrow P1')$
  and *ex2*: $\exists z2.\ \forall x.\ x \sqsubset z2 \longrightarrow (P2\ x \longleftrightarrow P2')$
  shows $\exists z.\ \forall x.\ x \sqsubset\ z \longrightarrow ((P1\ x \vee P2\ x) \longleftrightarrow (P1' \vee P2'))$
⟨*proof*⟩

lemma *minf-ex*[*no-atp*]: **assumes** *ex*:$\exists z.\ \forall x.\ x \sqsubset z \longrightarrow (P\ x \longleftrightarrow P1)$ **and** *p1*: $P1$ **shows** $\exists\ x.\ P\ x$
⟨*proof*⟩

end

locale *constr-dense-linorder* = *linorder-no-lb* + *linorder-no-ub* +
  fixes *between*
  assumes *between-less*: $less\ x\ y \Longrightarrow less\ x\ (between\ x\ y) \wedge less\ (between\ x\ y)\ y$
    and *between-same*: $between\ x\ x = x$

sublocale *constr-dense-linorder* < *dense-linorder*
  ⟨*proof*⟩

context *constr-dense-linorder*
begin

lemma *rinf-U*[*no-atp*]:
  assumes *fU*: *finite U*
  and *lin-dense*: $\forall x\ l\ u.\ (\forall\ t.\ l \sqsubset t \wedge t \sqsubset u \longrightarrow t \notin U) \wedge l \sqsubset x \wedge x \sqsubset u \wedge P\ x \longrightarrow (\forall\ y.\ l \sqsubset y \wedge y \sqsubset u \longrightarrow P\ y\ )$
  and *nmpiU*: $\forall x.\ \neg\ MP \wedge \neg PP \wedge P\ x \longrightarrow (\exists\ u \in U.\ \exists\ u' \in U.\ u \sqsubseteq x \wedge x \sqsubseteq$

u')
and *nmi*: $\neg$ *MP* **and** *npi*: $\neg$ *PP* **and** *ex*: $\exists\ x.\ \ P\ x$
shows $\exists\ u{\in}\ U.\ \exists\ u' \in\ U.\ P$ (*between u u'*)
⟨*proof*⟩
term *linorder.Min less-eq*
⟨*proof*⟩

theorem *fr-eq*[*no-atp*]:
assumes *fU*: *finite U*
and *lin-dense*: $\forall x\ l\ u.\ (\forall\ \ t.\ l \sqsubset t \wedge t{\sqsubset}\ u \longrightarrow t \notin U) \wedge l{\sqsubset}\ x \wedge x \sqsubset u \wedge P\ x$
$\longrightarrow (\forall\ \ y.\ l \sqsubset y \wedge y \sqsubset u \longrightarrow P\ y\ )$
and *nmibnd*: $\forall x.\ \neg\ MP \wedge P\ x \longrightarrow (\exists\ \ u{\in}\ U.\ u \sqsubseteq x)$
and *npibnd*: $\forall x.\ \neg PP \wedge P\ x \longrightarrow (\exists\ \ u{\in}\ U.\ x \sqsubseteq u)$
and *mi*: $\exists z.\ \forall x.\ x \sqsubset z \longrightarrow (P\ x = MP)$ **and** *pi*: $\exists z.\ \forall x.\ z \sqsubset x \longrightarrow (P\ x = PP)$
shows $(\exists\ \ x.\ P\ x) \equiv (MP \vee PP \vee (\exists\ \ u \in U.\ \exists\ \ u'{\in}\ U.\ P$ (*between u u'*)))
(**is** - $\equiv$ (- $\vee$ - $\vee$ *?F*) **is** *?E* $\equiv$ *?D*)
⟨*proof*⟩

lemmas *minf-thms*[*no-atp*] = *minf-conj minf-disj minf-eq minf-neq minf-lt minf-le minf-gt minf-ge minf-P*
lemmas *pinf-thms*[*no-atp*] = *pinf-conj pinf-disj pinf-eq pinf-neq pinf-lt pinf-le pinf-gt pinf-ge pinf-P*

lemmas *nmi-thms*[*no-atp*] = *nmi-conj nmi-disj nmi-eq nmi-neq nmi-lt nmi-le nmi-gt nmi-ge nmi-P*
lemmas *npi-thms*[*no-atp*] = *npi-conj npi-disj npi-eq npi-neq npi-lt npi-le npi-gt npi-ge npi-P*
lemmas *lin-dense-thms*[*no-atp*] = *lin-dense-conj lin-dense-disj lin-dense-eq lin-dense-neq lin-dense-lt lin-dense-le lin-dense-gt lin-dense-ge lin-dense-P*

lemma *ferrack-axiom*[*no-atp*]: *constr-dense-linorder less-eq less between*
⟨*proof*⟩
lemma *atoms*[*no-atp*]:
shows *TERM* (*less* :: $'a \Rightarrow$ -)
and *TERM* (*less-eq* :: $'a \Rightarrow$ -)
and *TERM* (*op* = :: $'a \Rightarrow$ -) ⟨*proof*⟩

declare *ferrack-axiom* [*ferrack minf*: *minf-thms pinf*: *pinf-thms*
nmi: *nmi-thms npi*: *npi-thms lindense*:
lin-dense-thms qe: *fr-eq atoms*: *atoms*]

⟨*ML*⟩

end

⟨*ML*⟩

3.1 Ferrante and Rackoff algorithm over ordered fields

lemma *neg-prod-lt*:$(c::'a::linordered\text{-}field) < 0 \Longrightarrow ((c*x < 0) == (x > 0))$
⟨*proof*⟩

lemma *pos-prod-lt*:$(c::'a::linordered\text{-}field) > 0 \Longrightarrow ((c*x < 0) == (x < 0))$
⟨*proof*⟩

lemma *neg-prod-sum-lt*: $(c::'a::linordered\text{-}field) < 0 \Longrightarrow ((c*x + t< 0) == (x > (- 1/c)*t))$
⟨*proof*⟩

lemma *pos-prod-sum-lt*:$(c::'a::linordered\text{-}field) > 0 \Longrightarrow ((c*x + t < 0) == (x < (- 1/c)*t))$
⟨*proof*⟩

lemma *sum-lt*:$((x::'a::ordered\text{-}ab\text{-}group\text{-}add) + t < 0) == (x < - t)$
⟨*proof*⟩

lemma *neg-prod-le*:$(c::'a::linordered\text{-}field) < 0 \Longrightarrow ((c*x <= 0) == (x >= 0))$
⟨*proof*⟩

lemma *pos-prod-le*:$(c::'a::linordered\text{-}field) > 0 \Longrightarrow ((c*x <= 0) == (x <= 0))$
⟨*proof*⟩

lemma *neg-prod-sum-le*: $(c::'a::linordered\text{-}field) < 0 \Longrightarrow ((c*x + t <= 0) == (x >= (- 1/c)*t))$
⟨*proof*⟩

lemma *pos-prod-sum-le*:$(c::'a::linordered\text{-}field) > 0 \Longrightarrow ((c*x + t <= 0) == (x <= (- 1/c)*t))$
⟨*proof*⟩

lemma *sum-le*:$((x::'a::ordered\text{-}ab\text{-}group\text{-}add) + t <= 0) == (x <= - t)$
⟨*proof*⟩

lemma *nz-prod-eq*:$(c::'a::linordered\text{-}field) \neq 0 \Longrightarrow ((c*x = 0) == (x = 0))$ ⟨*proof*⟩
lemma *nz-prod-sum-eq*: $(c::'a::linordered\text{-}field) \neq 0 \Longrightarrow ((c*x + t = 0) == (x = (- 1/c)*t))$
⟨*proof*⟩
lemma *sum-eq*:$((x::'a::ordered\text{-}ab\text{-}group\text{-}add) + t = 0) == (x = - t)$
⟨*proof*⟩

interpretation *class-dense-linordered-field*: *constr-dense-linorder*
op <= *op* <
λ *x y*. $1/2 * ((x::'a::\{linordered\text{-}field,number\text{-}ring\}) + y)$
⟨*proof*⟩

⟨*ML*⟩

lemma *upper-bound-finite-set*:
assumes *fS*: *finite S*
shows $\exists (a::'a::linorder).\ \forall x \in S.\ f\ x \leq a$
⟨*proof*⟩

lemma *lower-bound-finite-set*:
assumes *fS*: *finite S*
shows $\exists (a::'a::linorder).\ \forall x \in S.\ f\ x \geq a$
⟨*proof*⟩

lemma *bound-finite-set*: **assumes** *f*: *finite S*
shows $\exists\, a.\ \forall x \in S.\ (f\ x{::}\ 'a::linorder) \leq a$
⟨*proof*⟩

end

4 FrechetDeriv: Frechet Derivative

theory *FrechetDeriv*
imports *Lim Complex-Main*
begin

definition
fderiv ::
$['a::real\text{-}normed\text{-}vector \Rightarrow 'b::real\text{-}normed\text{-}vector,\ 'a,\ 'a \Rightarrow 'b] \Rightarrow bool$
— Frechet derivative: D is derivative of function f at x
((*FDERIV* (-)/ (-)/ :> (-)) [*1000*, *1000*, *60*] *60*) **where**
FDERIV f x :> *D* = (*bounded-linear D* ∧
($\lambda h.$ *norm* (*f* (*x* + *h*) − *f x* − *D h*) / *norm h*) −− *0* −−> *0*)

lemma *FDERIV-I*:
⟦*bounded-linear D*; ($\lambda h.$ *norm* (*f* (*x* + *h*) − *f x* − *D h*) / *norm h*) −− *0* −−> *0*⟧
⟹ *FDERIV f x* :> *D*
⟨*proof*⟩

lemma *FDERIV-D*:
FDERIV f x :> *D* ⟹ ($\lambda h.$ *norm* (*f* (*x* + *h*) − *f x* − *D h*) / *norm h*) −− *0* −−> *0*
⟨*proof*⟩

lemma *FDERIV-bounded-linear*: *FDERIV f x* :> *D* ⟹ *bounded-linear D*
⟨*proof*⟩

lemma *bounded-linear-zero*:

bounded-linear (λx::$'a$::*real-normed-vector*. *0*::$'b$::*real-normed-vector*)
⟨*proof*⟩

lemma *FDERIV-const*: *FDERIV* (λx. k) x :> (λh. *0*)
⟨*proof*⟩

lemma *bounded-linear-ident*:
bounded-linear (λx::$'a$::*real-normed-vector*. x)
⟨*proof*⟩

lemma *FDERIV-ident*: *FDERIV* (λx. x) x :> (λh. h)
⟨*proof*⟩

4.1 Addition

lemma *bounded-linear-add*:
assumes *bounded-linear f*
assumes *bounded-linear g*
shows *bounded-linear* (λx. $f\ x + g\ x$)
⟨*proof*⟩

lemma *norm-ratio-ineq*:
fixes $x\ y$ :: $'a$::*real-normed-vector*
fixes h :: $'b$::*real-normed-vector*
shows *norm* $(x + y)$ / *norm* $h \leq$ *norm* x / *norm* h + *norm* y / *norm* h
⟨*proof*⟩

lemma *FDERIV-add*:
assumes f: *FDERIV f x* :> F
assumes g: *FDERIV g x* :> G
shows *FDERIV* (λx. $f\ x + g\ x$) x :> (λh. $F\ h + G\ h$)
⟨*proof*⟩

4.2 Subtraction

lemma *bounded-linear-minus*:
assumes *bounded-linear f*
shows *bounded-linear* (λx. $-\ f\ x$)
⟨*proof*⟩

lemma *FDERIV-minus*:
FDERIV f x :> $F \Longrightarrow$ *FDERIV* (λx. $-\ f\ x$) x :> (λh. $-\ F\ h$)
⟨*proof*⟩

lemma *FDERIV-diff*:
⟦*FDERIV f x* :> F; *FDERIV g x* :> G⟧
$\Longrightarrow$ *FDERIV* (λx. $f\ x - g\ x$) x :> (λh. $F\ h - G\ h$)
⟨*proof*⟩

4.3 Continuity

lemma *FDERIV-isCont*:
  assumes *f*: *FDERIV f x* :> *F*
  shows *isCont f x*
⟨*proof*⟩

4.4 Composition

lemma *real-divide-cancel-lemma*:
  fixes *a b c* :: *real*
  shows $(b = 0 \Longrightarrow a = 0) \Longrightarrow (a\ /\ b) * (b\ /\ c) = a\ /\ c$
⟨*proof*⟩

lemma *bounded-linear-compose*:
  assumes *bounded-linear f*
  assumes *bounded-linear g*
  shows *bounded-linear* $(\lambda x.\ f\ (g\ x))$
⟨*proof*⟩

lemma *FDERIV-compose*:
  fixes *f* :: *'a::real-normed-vector* ⇒ *'b::real-normed-vector*
  fixes *g* :: *'b::real-normed-vector* ⇒ *'c::real-normed-vector*
  assumes *f*: *FDERIV f x* :> *F*
  assumes *g*: *FDERIV g* (*f x*) :> *G*
  shows *FDERIV* $(\lambda x.\ g\ (f\ x))\ x$:> $(\lambda h.\ G\ (F\ h))$
⟨*proof*⟩

4.5 Product Rule

lemma (**in** *bounded-bilinear*) *FDERIV-lemma*:
  $a' ** b' - a ** b - (a ** B + A ** b)$
   $= a ** (b' - b - B) + (a' - a - A) ** b' + A ** (b' - b)$
⟨*proof*⟩

lemma (**in** *bounded-bilinear*) *FDERIV*:
  fixes *x* :: *'d::real-normed-vector*
  assumes *f*: *FDERIV f x* :> *F*
  assumes *g*: *FDERIV g x* :> *G*
  shows *FDERIV* $(\lambda x.\ f\ x ** g\ x)\ x$:> $(\lambda h.\ f\ x ** G\ h + F\ h ** g\ x)$
⟨*proof*⟩

lemmas *FDERIV-mult* = *mult.FDERIV*

lemmas *FDERIV-scaleR* = *scaleR.FDERIV*

4.6 Powers

lemma *FDERIV-power-Suc*:
  fixes *x* :: *'a::{real-normed-algebra,comm-ring-1}*

shows *FDERIV* ($\lambda x.\ x$ ^ *Suc n*) x :> ($\lambda h.$ (*1* + *of-nat n*) $*$ x ^ n $*$ h)
⟨*proof*⟩

lemma *FDERIV-power*:
fixes x :: *'a::{real-normed-algebra,comm-ring-1}*
shows *FDERIV* ($\lambda x.\ x$ ^ n) x :> ($\lambda h.$ *of-nat n* $*$ x ^ (n − *1*) $*$ h)
⟨*proof*⟩

4.7 Inverse

lemmas *bounded-linear-mult-const* =
mult.bounded-linear-left [*THEN bounded-linear-compose*]

lemmas *bounded-linear-const-mult* =
mult.bounded-linear-right [*THEN bounded-linear-compose*]

lemma *FDERIV-inverse*:
fixes x :: *'a::real-normed-div-algebra*
assumes x: $x \neq 0$
shows *FDERIV inverse x* :> ($\lambda h.$ − (*inverse x* $*$ h $*$ *inverse x*))
(**is** *FDERIV ?inv* - :> -)
⟨*proof*⟩

4.8 Alternate definition

lemma *field-fderiv-def*:
fixes x :: *'a::real-normed-field* **shows**
FDERIV f x :> ($\lambda h.\ h * D$) = ($\lambda h.$ (f (x + h) − f x) / h) −− *0* −−> D
⟨*proof*⟩

end

5 Inner-Product: Inner Product Spaces and the Gradient Derivative

theory *Inner-Product*
imports *Complex-Main FrechetDeriv*
begin

5.1 Real inner product spaces

Temporarily relax type constraints for *open*, *dist*, and *norm*.

⟨*ML*⟩

class *real-inner* = *real-vector* + *sgn-div-norm* + *dist-norm* + *open-dist* +
fixes *inner* :: *'a* ⇒ *'a* ⇒ *real*
assumes *inner-commute*: *inner x y* = *inner y x*

and *inner-add-left*: *inner* $(x + y)$ z = *inner* x z + *inner* y z
and *inner-scaleR-left* [*simp*]: *inner* (*scaleR* r x) y = r ∗ (*inner* x y)
and *inner-ge-zero* [*simp*]: $0 \leq$ *inner* x x
and *inner-eq-zero-iff* [*simp*]: *inner* x x = $0 \longleftrightarrow x = 0$
and *norm-eq-sqrt-inner*: *norm* x = *sqrt* (*inner* x x)
begin

lemma *inner-zero-left* [*simp*]: *inner* 0 x = 0
⟨*proof*⟩

lemma *inner-minus-left* [*simp*]: *inner* $(- x)$ y = − *inner* x y
⟨*proof*⟩

lemma *inner-diff-left*: *inner* $(x - y)$ z = *inner* x z − *inner* y z
⟨*proof*⟩

Transfer distributivity rules to right argument.

lemma *inner-add-right*: *inner* x $(y + z)$ = *inner* x y + *inner* x z
⟨*proof*⟩

lemma *inner-scaleR-right* [*simp*]: *inner* x (*scaleR* r y) = r ∗ (*inner* x y)
⟨*proof*⟩

lemma *inner-zero-right* [*simp*]: *inner* x 0 = 0
⟨*proof*⟩

lemma *inner-minus-right* [*simp*]: *inner* x $(- y)$ = − *inner* x y
⟨*proof*⟩

lemma *inner-diff-right*: *inner* x $(y - z)$ = *inner* x y − *inner* x z
⟨*proof*⟩

lemmas *inner-add* [*algebra-simps*] = *inner-add-left inner-add-right*
lemmas *inner-diff* [*algebra-simps*] = *inner-diff-left inner-diff-right*
lemmas *inner-scaleR* = *inner-scaleR-left inner-scaleR-right*

Legacy theorem names

lemmas *inner-left-distrib* = *inner-add-left*
lemmas *inner-right-distrib* = *inner-add-right*
lemmas *inner-distrib* = *inner-left-distrib inner-right-distrib*

lemma *inner-gt-zero-iff* [*simp*]: $0 <$ *inner* x x $\longleftrightarrow x \neq 0$
⟨*proof*⟩

lemma *power2-norm-eq-inner*: $(norm\ x)^2$ = *inner* x x
⟨*proof*⟩

lemma *Cauchy-Schwarz-ineq*:
$(inner\ x\ y)^2 \leq$ *inner* x x ∗ *inner* y y

⟨*proof*⟩

lemma *Cauchy-Schwarz-ineq2*:
$|inner\ x\ y| \le norm\ x * norm\ y$
⟨*proof*⟩

subclass *real-normed-vector*
⟨*proof*⟩

end

Re-enable constraints for *open*, *dist*, and *norm*.

⟨*ML*⟩

interpretation *inner*:
bounded-bilinear inner::'a::real-inner ⇒ 'a ⇒ real
⟨*proof*⟩

interpretation *inner-left*:
bounded-linear λx::'a::real-inner. inner x y
⟨*proof*⟩

interpretation *inner-right*:
bounded-linear λy::'a::real-inner. inner x y
⟨*proof*⟩

5.2 Class instances

instantiation *real* :: *real-inner*
begin

definition *inner-real-def* [*simp*]: *inner* = *op* *

instance ⟨*proof*⟩

end

instantiation *complex* :: *real-inner*
begin

definition *inner-complex-def*:
*inner x y = Re x * Re y + Im x * Im y*

instance ⟨*proof*⟩

end

5.3 Gradient derivative

definition

gderiv ::
  $['a::real\text{-}inner \Rightarrow real,\ 'a,\ 'a] \Rightarrow bool$
    $((GDERIV\ (\text{-})/\ (\text{-})/\ :>\ (\text{-}))\ [1000,\ 1000,\ 60]\ 60)$

where
  $GDERIV\ f\ x :> D \longleftrightarrow FDERIV\ f\ x :> (\lambda h.\ inner\ h\ D)$

lemma *deriv-fderiv*: $DERIV\ f\ x :> D \longleftrightarrow FDERIV\ f\ x :> (\lambda h.\ h * D)$
  ⟨*proof*⟩

lemma *gderiv-deriv* [*simp*]: $GDERIV\ f\ x :> D \longleftrightarrow DERIV\ f\ x :> D$
  ⟨*proof*⟩

lemma *GDERIV-DERIV-compose*:
  $[\![GDERIV\ f\ x :> df;\ DERIV\ g\ (f\ x) :> dg]\!]$
  $\Longrightarrow GDERIV\ (\lambda x.\ g\ (f\ x))\ x :> scaleR\ dg\ df$
  ⟨*proof*⟩

lemma *FDERIV-subst*: $[\![FDERIV\ f\ x :> df;\ df = d]\!] \Longrightarrow FDERIV\ f\ x :> d$
  ⟨*proof*⟩

lemma *GDERIV-subst*: $[\![GDERIV\ f\ x :> df;\ df = d]\!] \Longrightarrow GDERIV\ f\ x :> d$
  ⟨*proof*⟩

lemma *GDERIV-const*: $GDERIV\ (\lambda x.\ k)\ x :> 0$
  ⟨*proof*⟩

lemma *GDERIV-add*:
  $[\![GDERIV\ f\ x :> df;\ GDERIV\ g\ x :> dg]\!]$
  $\Longrightarrow GDERIV\ (\lambda x.\ f\ x + g\ x)\ x :> df + dg$
  ⟨*proof*⟩

lemma *GDERIV-minus*:
  $GDERIV\ f\ x :> df \Longrightarrow GDERIV\ (\lambda x.\ -\ f\ x)\ x :> -\ df$
  ⟨*proof*⟩

lemma *GDERIV-diff*:
  $[\![GDERIV\ f\ x :> df;\ GDERIV\ g\ x :> dg]\!]$
  $\Longrightarrow GDERIV\ (\lambda x.\ f\ x - g\ x)\ x :> df - dg$
  ⟨*proof*⟩

lemma *GDERIV-scaleR*:
  $[\![DERIV\ f\ x :> df;\ GDERIV\ g\ x :> dg]\!]$
  $\Longrightarrow GDERIV\ (\lambda x.\ scaleR\ (f\ x)\ (g\ x))\ x$
  $:> (scaleR\ (f\ x)\ dg + scaleR\ df\ (g\ x))$
  ⟨*proof*⟩

lemma *GDERIV-mult*:
  $[\![GDERIV\ f\ x :> df;\ GDERIV\ g\ x :> dg]\!]$
  $\Longrightarrow GDERIV\ (\lambda x.\ f\ x * g\ x)\ x :> scaleR\ (f\ x)\ dg + scaleR\ (g\ x)\ df$

$\langle proof \rangle$

lemma *GDERIV-inverse*:
$[\![GDERIV\ f\ x :> df;\ f\ x \neq 0]\!]$
$\Longrightarrow GDERIV\ (\lambda x.\ inverse\ (f\ x))\ x :> -\ (inverse\ (f\ x))^2 *_R df$
$\langle proof \rangle$

lemma *GDERIV-norm*:
assumes $x \neq 0$ **shows** $GDERIV\ (\lambda x.\ norm\ x)\ x :> sgn\ x$
$\langle proof \rangle$

lemmas *FDERIV-norm* = *GDERIV-norm* [*unfolded gderiv-def*]

end

6 L2-Norm: Square root of sum of squares

theory *L2-Norm*
imports *NthRoot*
begin

definition
$setL2\ f\ A = sqrt\ (\sum i{\in}A.\ (f\ i)^2)$

lemma *setL2-cong*:
$[\![A = B;\ \bigwedge x.\ x \in B \Longrightarrow f\ x = g\ x]\!] \Longrightarrow setL2\ f\ A = setL2\ g\ B$
$\langle proof \rangle$

lemma *strong-setL2-cong*:
$[\![A = B;\ \bigwedge x.\ x \in B =simp=> f\ x = g\ x]\!] \Longrightarrow setL2\ f\ A = setL2\ g\ B$
$\langle proof \rangle$

lemma *setL2-infinite* [*simp*]: $\neg\ finite\ A \Longrightarrow setL2\ f\ A = 0$
$\langle proof \rangle$

lemma *setL2-empty* [*simp*]: $setL2\ f\ \{\} = 0$
$\langle proof \rangle$

lemma *setL2-insert* [*simp*]:
$[\![finite\ F;\ a \notin F]\!] \Longrightarrow$
$setL2\ f\ (insert\ a\ F) = sqrt\ ((f\ a)^2 + (setL2\ f\ F)^2)$
$\langle proof \rangle$

lemma *setL2-nonneg* [*simp*]: $0 \leq setL2\ f\ A$
$\langle proof \rangle$

lemma *setL2-0'*: $\forall a{\in}A.\ f\ a = 0 \Longrightarrow setL2\ f\ A = 0$
$\langle proof \rangle$

lemma *setL2-constant*: $setL2\ (\lambda x.\ y)\ A = sqrt\ (of\text{-}nat\ (card\ A)) * |y|$
⟨*proof*⟩

lemma *setL2-mono*:
assumes $\bigwedge i.\ i \in K \Longrightarrow f\ i \leq g\ i$
assumes $\bigwedge i.\ i \in K \Longrightarrow 0 \leq f\ i$
shows $setL2\ f\ K \leq setL2\ g\ K$
⟨*proof*⟩

lemma *setL2-strict-mono*:
assumes *finite* K **and** $K \neq \{\}$
assumes $\bigwedge i.\ i \in K \Longrightarrow f\ i < g\ i$
assumes $\bigwedge i.\ i \in K \Longrightarrow 0 \leq f\ i$
shows $setL2\ f\ K < setL2\ g\ K$
⟨*proof*⟩

lemma *setL2-right-distrib*:
$0 \leq r \Longrightarrow r * setL2\ f\ A = setL2\ (\lambda x.\ r * f\ x)\ A$
⟨*proof*⟩

lemma *setL2-left-distrib*:
$0 \leq r \Longrightarrow setL2\ f\ A * r = setL2\ (\lambda x.\ f\ x * r)\ A$
⟨*proof*⟩

lemma *setsum-nonneg-eq-0-iff*:
fixes $f :: 'a \Rightarrow 'b::ordered\text{-}ab\text{-}group\text{-}add$
shows $[\![finite\ A;\ \forall x{\in}A.\ 0 \leq f\ x]\!] \Longrightarrow setsum\ f\ A = 0 \longleftrightarrow (\forall x{\in}A.\ f\ x = 0)$
⟨*proof*⟩

lemma *setL2-eq-0-iff*: $finite\ A \Longrightarrow setL2\ f\ A = 0 \longleftrightarrow (\forall x{\in}A.\ f\ x = 0)$
⟨*proof*⟩

lemma *setL2-triangle-ineq*:
shows $setL2\ (\lambda i.\ f\ i + g\ i)\ A \leq setL2\ f\ A + setL2\ g\ A$
⟨*proof*⟩

lemma *sqrt-sum-squares-le-sum*:
$[\![0 \leq x;\ 0 \leq y]\!] \Longrightarrow sqrt\ (x^2 + y^2) \leq x + y$
⟨*proof*⟩

lemma *setL2-le-setsum* [*rule-format*]:
$(\forall i{\in}A.\ 0 \leq f\ i) \longrightarrow setL2\ f\ A \leq setsum\ f\ A$
⟨*proof*⟩

lemma *sqrt-sum-squares-le-sum-abs*: $sqrt\ (x^2 + y^2) \leq |x| + |y|$
⟨*proof*⟩

lemma *setL2-le-setsum-abs*: $setL2\ f\ A \leq (\sum i{\in}A.\ |f\ i|)$

⟨*proof*⟩

lemma *setL2-mult-ineq-lemma*:
fixes *a b c d* :: *real*
shows $2 * (a * c) * (b * d) \leq a^2 * d^2 + b^2 * c^2$
⟨*proof*⟩

lemma *setL2-mult-ineq*: $(\sum i{\in}A.\ |f\ i| * |g\ i|) \leq setL2\ f\ A * setL2\ g\ A$
⟨*proof*⟩

lemma *member-le-setL2*: $[\![finite\ A;\ i \in A]\!] \Longrightarrow f\ i \leq setL2\ f\ A$
⟨*proof*⟩

end

7 Numeral-Type: Numeral Syntax for Types

theory *Numeral-Type*
imports *Main*
begin

7.1 Preliminary lemmas

lemma (**in** *type-definition*) *univ*:
UNIV = Abs ‘ A
⟨*proof*⟩

lemma (**in** *type-definition*) *card*: *card (UNIV :: ′b set) = card A*
⟨*proof*⟩

7.2 Cardinalities of types

syntax *-type-card* :: *type => nat ((1CARD/(1′(-′))))*

translations *CARD(′t) => CONST card (CONST UNIV :: ′t set)*

⟨*ML*⟩

lemma *card-unit* [*simp*]: *CARD(unit) = 1*
⟨*proof*⟩

lemma *card-bool* [*simp*]: *CARD(bool) = 2*
⟨*proof*⟩

lemma *card-prod* [*simp*]: *CARD(′a × ′b) = CARD(′a::finite) * CARD(′b::finite)*
⟨*proof*⟩

lemma *card-sum* [*simp*]: *CARD(′a + ′b) = CARD(′a::finite) + CARD(′b::finite)*

⟨*proof*⟩

lemma *card-option* [*simp*]: $CARD('a\ option) = Suc\ CARD('a::finite)$
⟨*proof*⟩

lemma *card-set* [*simp*]: $CARD('a\ set) = 2\ \hat{}\ CARD('a::finite)$
⟨*proof*⟩

lemma *card-nat* [*simp*]: $CARD(nat) = 0$
⟨*proof*⟩

7.3 Classes with at least 1 and 2

Class finite already captures "at least 1"

lemma *zero-less-card-finite* [*simp*]: $0 < CARD('a::finite)$
⟨*proof*⟩

lemma *one-le-card-finite* [*simp*]: $Suc\ 0 \leq CARD('a::finite)$
⟨*proof*⟩

Class for cardinality "at least 2"

class *card2* = *finite* +
assumes *two-le-card*: $2 \leq CARD('a)$

lemma *one-less-card*: $Suc\ 0 < CARD('a::card2)$
⟨*proof*⟩

lemma *one-less-int-card*: $1 < int\ CARD('a::card2)$
⟨*proof*⟩

7.4 Numeral Types

typedef (**open**) *num0* = *UNIV* :: *nat set* ⟨*proof*⟩
typedef (**open**) *num1* = *UNIV* :: *unit set* ⟨*proof*⟩

typedef (**open**) $'a\ bit0 = \{0\ ..< 2 * int\ CARD('a::finite)\}$
⟨*proof*⟩

typedef (**open**) $'a\ bit1 = \{0\ ..< 1 + 2 * int\ CARD('a::finite)\}$
⟨*proof*⟩

lemma *card-num0* [*simp*]: $CARD\ (num0) = 0$
⟨*proof*⟩

lemma *card-num1* [*simp*]: $CARD(num1) = 1$
⟨*proof*⟩

lemma *card-bit0* [*simp*]: $CARD('a\ bit0) = 2 * CARD('a::finite)$
⟨*proof*⟩

lemma *card-bit1* [*simp*]: $CARD('a\ bit1) = Suc\ (2 * CARD('a::finite))$
⟨*proof*⟩

instance *num1* :: *finite*
⟨*proof*⟩

instance *bit0* :: (*finite*) *card2*
⟨*proof*⟩

instance *bit1* :: (*finite*) *card2*
⟨*proof*⟩

7.5 Locale for modular arithmetic subtypes

locale *mod-type* =
fixes $n :: int$
and $Rep :: 'a::\{zero,one,plus,times,uminus,minus\} \Rightarrow int$
and $Abs :: int \Rightarrow 'a::\{zero,one,plus,times,uminus,minus\}$
assumes *type*: *type-definition Rep Abs* $\{0..<n\}$
and *size1*: $1 < n$
and *zero-def*: $0 = Abs\ 0$
and *one-def*: $1 = Abs\ 1$
and *add-def*: $x + y = Abs\ ((Rep\ x + Rep\ y)\ mod\ n)$
and *mult-def*: $x * y = Abs\ ((Rep\ x * Rep\ y)\ mod\ n)$
and *diff-def*: $x - y = Abs\ ((Rep\ x - Rep\ y)\ mod\ n)$
and *minus-def*: $-\ x = Abs\ ((-\ Rep\ x)\ mod\ n)$
begin

lemma *size0*: $0 < n$
⟨*proof*⟩

lemmas *definitions* =
zero-def one-def add-def mult-def minus-def diff-def

lemma *Rep-less-n*: $Rep\ x < n$
⟨*proof*⟩

lemma *Rep-le-n*: $Rep\ x \leq n$
⟨*proof*⟩

lemma *Rep-inject-sym*: $x = y \longleftrightarrow Rep\ x = Rep\ y$
⟨*proof*⟩

lemma *Rep-inverse*: $Abs\ (Rep\ x) = x$
⟨*proof*⟩

lemma *Abs-inverse*: $m \in \{0..<n\} \Longrightarrow Rep\ (Abs\ m) = m$
⟨*proof*⟩

lemma *Rep-Abs-mod*: $Rep\ (Abs\ (m\ mod\ n)) = m\ mod\ n$
⟨*proof*⟩

lemma *Rep-Abs-0*: $Rep\ (Abs\ 0) = 0$
⟨*proof*⟩

lemma *Rep-0*: $Rep\ 0 = 0$
⟨*proof*⟩

lemma *Rep-Abs-1*: $Rep\ (Abs\ 1) = 1$
⟨*proof*⟩

lemma *Rep-1*: $Rep\ 1 = 1$
⟨*proof*⟩

lemma *Rep-mod*: $Rep\ x\ mod\ n = Rep\ x$
⟨*proof*⟩

lemmas *Rep-simps* =
  Rep-inject-sym Rep-inverse Rep-Abs-mod Rep-mod Rep-Abs-0 Rep-Abs-1

lemma *comm-ring-1*: $OFCLASS('a,\ comm\text{-}ring\text{-}1\text{-}class)$
⟨*proof*⟩

end

locale *mod-ring* = *mod-type* +
  constrains $n :: int$
  and $Rep :: 'a::\{number\text{-}ring\} \Rightarrow int$
  and $Abs :: int \Rightarrow 'a::\{number\text{-}ring\}$
begin

lemma *of-nat-eq*: $of\text{-}nat\ k = Abs\ (int\ k\ mod\ n)$
⟨*proof*⟩

lemma *of-int-eq*: $of\text{-}int\ z = Abs\ (z\ mod\ n)$
⟨*proof*⟩

lemma *Rep-number-of*:
  $Rep\ (number\text{-}of\ w) = number\text{-}of\ w\ mod\ n$
⟨*proof*⟩

lemma *iszero-number-of*:
  $iszero\ (number\text{-}of\ w::'a) \longleftrightarrow number\text{-}of\ w\ mod\ n = 0$
⟨*proof*⟩

lemma *cases*:
  assumes *1*: $\bigwedge z.\ [\![(x::'a) = of\text{-}int\ z;\ 0 \le z;\ z < n]\!] \Longrightarrow P$

shows P
$\langle proof \rangle$

lemma *induct*:
$(\bigwedge z.\ [\![0 \leq z;\ z < n]\!] \Longrightarrow P\ (\textit{of-int}\ z)) \Longrightarrow P\ (x::'a)$
$\langle proof \rangle$

end

7.6 Number ring instances

Unfortunately a number ring instance is not possible for *num1*, since 0 and 1 are not distinct.

instantiation *num1* :: {*comm-ring*,*comm-monoid-mult*,*number*}
begin

lemma *num1-eq-iff*: $(x::num1) = (y::num1) \longleftrightarrow \textit{True}$
$\langle proof \rangle$

instance $\langle proof \rangle$

end

instantiation
bit0 **and** *bit1* :: (*finite*) {*zero*,*one*,*plus*,*times*,*uminus*,*minus*}
begin

definition $\textit{Abs-bit0}' :: int \Rightarrow 'a\ bit0$ **where**
$\textit{Abs-bit0}'\ x = \textit{Abs-bit0}\ (x\ mod\ int\ CARD('a\ bit0))$

definition $\textit{Abs-bit1}' :: int \Rightarrow 'a\ bit1$ **where**
$\textit{Abs-bit1}'\ x = \textit{Abs-bit1}\ (x\ mod\ int\ CARD('a\ bit1))$

definition $0 = \textit{Abs-bit0}\ 0$
definition $1 = \textit{Abs-bit0}\ 1$
definition $x + y = \textit{Abs-bit0}'\ (\textit{Rep-bit0}\ x + \textit{Rep-bit0}\ y)$
definition $x * y = \textit{Abs-bit0}'\ (\textit{Rep-bit0}\ x * \textit{Rep-bit0}\ y)$
definition $x - y = \textit{Abs-bit0}'\ (\textit{Rep-bit0}\ x - \textit{Rep-bit0}\ y)$
definition $-\ x = \textit{Abs-bit0}'\ (-\ \textit{Rep-bit0}\ x)$

definition $0 = \textit{Abs-bit1}\ 0$
definition $1 = \textit{Abs-bit1}\ 1$
definition $x + y = \textit{Abs-bit1}'\ (\textit{Rep-bit1}\ x + \textit{Rep-bit1}\ y)$
definition $x * y = \textit{Abs-bit1}'\ (\textit{Rep-bit1}\ x * \textit{Rep-bit1}\ y)$
definition $x - y = \textit{Abs-bit1}'\ (\textit{Rep-bit1}\ x - \textit{Rep-bit1}\ y)$
definition $-\ x = \textit{Abs-bit1}'\ (-\ \textit{Rep-bit1}\ x)$

instance $\langle proof \rangle$

end

interpretation *bit0*:
  mod-type int CARD('a::finite bit0)
    Rep-bit0 :: 'a::finite bit0 ⇒ int
    Abs-bit0 :: int ⇒ 'a::finite bit0
⟨*proof*⟩

interpretation *bit1*:
  mod-type int CARD('a::finite bit1)
    Rep-bit1 :: 'a::finite bit1 ⇒ int
    Abs-bit1 :: int ⇒ 'a::finite bit1
⟨*proof*⟩

instance *bit0* :: (*finite*) *comm-ring-1*
  ⟨*proof*⟩

instance *bit1* :: (*finite*) *comm-ring-1*
  ⟨*proof*⟩

instantiation *bit0* **and** *bit1* :: (*finite*) *number-ring*
begin

definition (*number-of w* :: - *bit0*) = *of-int w*

definition (*number-of w* :: - *bit1*) = *of-int w*

instance ⟨*proof*⟩

end

interpretation *bit0*:
  mod-ring int CARD('a::finite bit0)
    Rep-bit0 :: 'a::finite bit0 ⇒ int
    Abs-bit0 :: int ⇒ 'a::finite bit0
  ⟨*proof*⟩

interpretation *bit1*:
  mod-ring int CARD('a::finite bit1)
    Rep-bit1 :: 'a::finite bit1 ⇒ int
    Abs-bit1 :: int ⇒ 'a::finite bit1
  ⟨*proof*⟩

Set up cases, induction, and arithmetic

lemmas *bit0-cases* [*case-names of-int*, *cases type*: *bit0*] = *bit0.cases*
lemmas *bit1-cases* [*case-names of-int*, *cases type*: *bit1*] = *bit1.cases*

lemmas *bit0-induct* [*case-names of-int*, *induct type*: *bit0*] = *bit0.induct*
lemmas *bit1-induct* [*case-names of-int*, *induct type*: *bit1*] = *bit1.induct*

lemmas *bit0-iszero-number-of* [*simp*] = *bit0.iszero-number-of*
lemmas *bit1-iszero-number-of* [*simp*] = *bit1.iszero-number-of*

7.7 Syntax

syntax
  -NumeralType :: *num-const* => *type* (-)
  -NumeralType0 :: *type* (*0*)
  -NumeralType1 :: *type* (*1*)

translations
  (*type*) *1* == (*type*) *num1*
  (*type*) *0* == (*type*) *num0*

⟨*ML*⟩

7.8 Examples

lemma *CARD*(*0*) = *0* ⟨*proof*⟩
lemma *CARD*(*17*) = *17* ⟨*proof*⟩
lemma *8* ∗ *11* ^ *3* − *6* = (*2*::*5*) ⟨*proof*⟩

end

8 Finite-Cartesian-Product: Definition of finite Cartesian product types.

theory *Finite-Cartesian-Product*
imports *Inner-Product L2-Norm Numeral-Type*
begin

8.1 Finite Cartesian products, with indexing and lambdas.

typedef (**open** *Cart*)
  (*'a*, *'b*) *cart* = *UNIV* :: ((*'b*::*finite*) ⇒ *'a*) *set*
  morphisms *Cart-nth Cart-lambda* ⟨*proof*⟩

notation
  Cart-nth (**infixl** $ *90*) **and**
  Cart-lambda (**binder** χ *10*)

syntax *-finite-cart* :: *type* ⇒ *type* ⇒ *type* ((- ^/ -) [*15*, *16*] *15*)

⟨*ML*⟩

lemma *stupid-ext*: $(\forall x.\ f\ x = g\ x) \longleftrightarrow (f = g)$
⟨*proof*⟩

lemma *Cart-eq*: $(x = y) \longleftrightarrow (\forall i.\ x\$i = y\$i)$
⟨*proof*⟩

lemma *Cart-lambda-beta* [*simp*]: $Cart\text{-}lambda\ g\ \$\ i = g\ i$
⟨*proof*⟩

lemma *Cart-lambda-unique*: $(\forall i.\ f\$i = g\ i) \longleftrightarrow Cart\text{-}lambda\ g = f$
⟨*proof*⟩

lemma *Cart-lambda-eta*: $(\chi\ i.\ (g\$i)) = g$
⟨*proof*⟩

8.2 Group operations and class instances

instantiation *cart* :: (*zero*,*finite*) *zero*
begin
definition *vector-zero-def* : $0 \equiv (\chi\ i.\ 0)$
instance ⟨*proof*⟩
end

instantiation *cart* :: (*plus*,*finite*) *plus*
begin
definition *vector-add-def* : $op\ + \equiv (\lambda\ x\ y.\ \ (\chi\ i.\ (x\$i) + (y\$i)))$
instance ⟨*proof*⟩
end

instantiation *cart* :: (*minus*,*finite*) *minus*
begin
definition *vector-minus-def* : $op\ - \equiv (\lambda\ x\ y.\ \ (\chi\ i.\ (x\$i) - (y\$i)))$
instance ⟨*proof*⟩
end

instantiation *cart* :: (*uminus*,*finite*) *uminus*
begin
definition *vector-uminus-def* : $uminus \equiv (\lambda\ x.\ \ (\chi\ i.\ -\ (x\$i)))$
instance ⟨*proof*⟩
end

lemma *zero-index* [*simp*]: $0\ \$\ i = 0$
⟨*proof*⟩

lemma *vector-add-component* [*simp*]: $(x + y)\$i = x\$i + y\$i$
⟨*proof*⟩

lemma *vector-minus-component* [*simp*]: $(x - y)\$i = x\$i - y\$i$
⟨*proof*⟩

lemma *vector-uminus-component* [*simp*]: $(-\ x)\$i = -\ (x\$i)$
⟨*proof*⟩

instance *cart* :: (*semigroup-add*, *finite*) *semigroup-add*
⟨*proof*⟩

instance *cart* :: (*ab-semigroup-add*, *finite*) *ab-semigroup-add*
⟨*proof*⟩

instance *cart* :: (*monoid-add*, *finite*) *monoid-add*
⟨*proof*⟩

instance *cart* :: (*comm-monoid-add*, *finite*) *comm-monoid-add*
⟨*proof*⟩

instance *cart* :: (*cancel-semigroup-add*, *finite*) *cancel-semigroup-add*
⟨*proof*⟩

instance *cart* :: (*cancel-ab-semigroup-add*, *finite*) *cancel-ab-semigroup-add*
⟨*proof*⟩

instance *cart* :: (*cancel-comm-monoid-add*, *finite*) *cancel-comm-monoid-add* ⟨*proof*⟩

instance *cart* :: (*group-add*, *finite*) *group-add*
⟨*proof*⟩

instance *cart* :: (*ab-group-add*, *finite*) *ab-group-add*
⟨*proof*⟩

8.3 Real vector space

instantiation *cart* :: (*real-vector*, *finite*) *real-vector*
begin

definition *vector-scaleR-def*: $scaleR = (\lambda\ r\ x.\ (\chi\ i.\ scaleR\ r\ (x\$i)))$

lemma *vector-scaleR-component* [*simp*]: $(scaleR\ r\ x)\$i = scaleR\ r\ (x\$i)$
⟨*proof*⟩

instance
⟨*proof*⟩

end

8.4 Topological space

instantiation *cart* :: (*topological-space*, *finite*) *topological-space*
begin

definition *open-vector-def*:
$open\ (S :: ('a\ \hat{}\ 'b)\ set) \longleftrightarrow$
$(\forall x{\in}S.\ \exists A.\ (\forall i.\ open\ (A\ i) \wedge x\$i \in A\ i) \wedge$
$(\forall y.\ (\forall i.\ y\$i \in A\ i) \longrightarrow y \in S))$

instance ⟨*proof*⟩

end

lemma *open-vector-box*: $\forall i.\ open\ (S\ i) \Longrightarrow open\ \{x.\ \forall i.\ x\ \$\ i \in S\ i\}$
⟨*proof*⟩

lemma *open-vimage-Cart-nth*: $open\ S \Longrightarrow open\ ((\lambda x.\ x\ \$\ i)\ -`\ S)$
⟨*proof*⟩

lemma *closed-vimage-Cart-nth*: $closed\ S \Longrightarrow closed\ ((\lambda x.\ x\ \$\ i)\ -`\ S)$
⟨*proof*⟩

lemma *closed-vector-box*: $\forall i.\ closed\ (S\ i) \Longrightarrow closed\ \{x.\ \forall i.\ x\ \$\ i \in S\ i\}$
⟨*proof*⟩

lemma *tendsto-Cart-nth* [*tendsto-intros*]:
assumes $((\lambda x.\ f\ x) \ {-}{-}{-}{>}\ a)\ net$
shows $((\lambda x.\ f\ x\ \$\ i) \ {-}{-}{-}{>}\ a\ \$\ i)\ net$
⟨*proof*⟩

lemma *eventually-Ball-finite*:
assumes *finite A* **and** $\forall y{\in}A.\ eventually\ (\lambda x.\ P\ x\ y)\ net$
shows $eventually\ (\lambda x.\ \forall y{\in}A.\ P\ x\ y)\ net$
⟨*proof*⟩

lemma *eventually-all-finite*:
fixes $P :: 'a \Rightarrow 'b{::}finite \Rightarrow bool$
assumes $\bigwedge y.\ eventually\ (\lambda x.\ P\ x\ y)\ net$
shows $eventually\ (\lambda x.\ \forall y.\ P\ x\ y)\ net$
⟨*proof*⟩

lemma *tendsto-vector*:
assumes $\bigwedge i.\ ((\lambda x.\ f\ x\ \$\ i) \ {-}{-}{-}{>}\ a\ \$\ i)\ net$
shows $((\lambda x.\ f\ x) \ {-}{-}{-}{>}\ a)\ net$
⟨*proof*⟩

lemma *tendsto-Cart-lambda* [*tendsto-intros*]:
assumes $\bigwedge i.\ ((\lambda x.\ f\ x\ i) \ {-}{-}{-}{>}\ a\ i)\ net$
shows $((\lambda x.\ \chi\ i.\ f\ x\ i) \ {-}{-}{-}{>}\ (\chi\ i.\ a\ i))\ net$
⟨*proof*⟩

8.5 Metric

lemma *finite-choice*: $finite\ A \Longrightarrow \forall x{\in}A.\ \exists y.\ P\ x\ y \Longrightarrow \exists f.\ \forall x{\in}A.\ P\ x\ (f\ x)$
⟨*proof*⟩

instantiation *cart* :: (*metric-space*, *finite*) *metric-space*
begin

definition *dist-vector-def*:
$dist\ x\ y = setL2\ (\lambda i.\ dist\ (x\$i)\ (y\$i))\ UNIV$

lemma *dist-nth-le*: $dist\ (x\ \$\ i)\ (y\ \$\ i) \le dist\ x\ y$
⟨*proof*⟩

instance ⟨*proof*⟩

end

lemma *Cauchy-Cart-nth*:
$Cauchy\ (\lambda n.\ X\ n) \Longrightarrow Cauchy\ (\lambda n.\ X\ n\ \$\ i)$
⟨*proof*⟩

lemma *Cauchy-vector*:
fixes $X :: nat \Rightarrow 'a{::}metric\text{-}space\ \hat{}\ 'n$
assumes X: $\bigwedge i.\ Cauchy\ (\lambda n.\ X\ n\ \$\ i)$
shows $Cauchy\ (\lambda n.\ X\ n)$
⟨*proof*⟩

instance *cart* :: (*complete-space*, *finite*) *complete-space*
⟨*proof*⟩

8.6 Normed vector space

instantiation *cart* :: (*real-normed-vector*, *finite*) *real-normed-vector*
begin

definition *norm-vector-def*:
$norm\ x = setL2\ (\lambda i.\ norm\ (x\$i))\ UNIV$

definition *vector-sgn-def*:
$sgn\ (x{::}'a\hat{}'b) = scaleR\ (inverse\ (norm\ x))\ x$

instance ⟨*proof*⟩

end

lemma *norm-nth-le*: $norm\ (x\ \$\ i) \le norm\ x$
⟨*proof*⟩

interpretation *Cart-nth*: $bounded\text{-}linear\ \lambda x.\ x\ \$\ i$

⟨*proof*⟩

instance *cart* :: (*banach*, *finite*) *banach* ⟨*proof*⟩

8.7 Inner product space

instantiation *cart* :: (*real-inner*, *finite*) *real-inner*
begin

definition *inner-vector-def*:
  inner x y = *setsum* (λi. *inner* ($x\$i$) ($y\$i$)) *UNIV*

instance ⟨*proof*⟩

end

end

9 Infinite-Set: Infinite Sets and Related Concepts

theory *Infinite-Set*
imports *Main*
begin

9.1 Infinite Sets

Some elementary facts about infinite sets, mostly by Stefan Merz. Beware! Because "infinite" merely abbreviates a negation, these lemmas may not work well with *blast*.

abbreviation
  infinite :: *'a set* ⇒ *bool* **where**
  infinite S == ¬ *finite S*

Infinite sets are non-empty, and if we remove some elements from an infinite set, the result is still infinite.

lemma *infinite-imp-nonempty*: *infinite S* ==> $S \neq \{\}$
  ⟨*proof*⟩

lemma *infinite-remove*:
  infinite S $\Longrightarrow$ *infinite* ($S - \{a\}$)
  ⟨*proof*⟩

lemma *Diff-infinite-finite*:
  assumes *T*: *finite T* **and** *S*: *infinite S*
  shows *infinite* ($S - T$)
  ⟨*proof*⟩

lemma *Un-infinite*: *infinite* $S \Longrightarrow$ *infinite* $(S \cup T)$
⟨*proof*⟩

lemma *infinite-Un*: *infinite* $(S \cup T) \longleftrightarrow$ *infinite* $S \vee$ *infinite* T
⟨*proof*⟩

lemma *infinite-super*:
assumes T: $S \subseteq T$ **and** S: *infinite* S
shows *infinite* T
⟨*proof*⟩

As a concrete example, we prove that the set of natural numbers is infinite.

lemma *finite-nat-bounded*:
assumes S: *finite* $(S::nat\ set)$
shows $\exists k.\ S \subseteq \{..<k\}$ (**is** $\exists k.\ ?bounded\ S\ k$)
⟨*proof*⟩

lemma *finite-nat-iff-bounded*:
finite $(S::nat\ set) = (\exists k.\ S \subseteq \{..<k\})$ (**is** $?lhs = ?rhs$)
⟨*proof*⟩

lemma *finite-nat-iff-bounded-le*:
finite $(S::nat\ set) = (\exists k.\ S \subseteq \{..k\})$ (**is** $?lhs = ?rhs$)
⟨*proof*⟩

lemma *infinite-nat-iff-unbounded*:
infinite $(S::nat\ set) = (\forall m.\ \exists n.\ m<n \wedge n\in S)$
(**is** $?lhs = ?rhs$)
⟨*proof*⟩

lemma *infinite-nat-iff-unbounded-le*:
infinite $(S::nat\ set) = (\forall m.\ \exists n.\ m\le n \wedge n\in S)$
(**is** $?lhs = ?rhs$)
⟨*proof*⟩

For a set of natural numbers to be infinite, it is enough to know that for any number larger than some k, there is some larger number that is an element of the set.

lemma *unbounded-k-infinite*:
assumes k: $\forall m.\ k<m \longrightarrow (\exists n.\ m<n \wedge n\in S)$
shows *infinite* $(S::nat\ set)$
⟨*proof*⟩

lemma *nat-infinite*: *infinite* $(UNIV :: nat\ set)$
⟨*proof*⟩

lemma *nat-not-finite*: *finite* $(UNIV::nat\ set) \Longrightarrow R$
⟨*proof*⟩

Every infinite set contains a countable subset. More precisely we show that a set S is infinite if and only if there exists an injective function from the naturals into S.

lemma *range-inj-infinite*:
$inj\ (f::nat \Rightarrow 'a) \Longrightarrow infinite\ (range\ f)$
⟨*proof*⟩

lemma *int-infinite* [*simp*]:
shows $infinite\ (UNIV::int\ set)$
⟨*proof*⟩

The "only if" direction is harder because it requires the construction of a sequence of pairwise different elements of an infinite set S. The idea is to construct a sequence of non-empty and infinite subsets of S obtained by successively removing elements of S.

lemma *linorder-injI*:
assumes *hyp*: $!!x\ y.\ x < (y::'a::linorder) ==> f\ x \neq f\ y$
shows $inj\ f$
⟨*proof*⟩

lemma *infinite-countable-subset*:
assumes *inf*: $infinite\ (S::'a\ set)$
shows $\exists f.\ inj\ (f::nat \Rightarrow 'a) \wedge range\ f \subseteq S$
⟨*proof*⟩

lemma *infinite-iff-countable-subset*:
$infinite\ S = (\exists f.\ inj\ (f::nat \Rightarrow 'a) \wedge range\ f \subseteq S)$
⟨*proof*⟩

For any function with infinite domain and finite range there is some element that is the image of infinitely many domain elements. In particular, any infinite sequence of elements from a finite set contains some element that occurs infinitely often.

lemma *inf-img-fin-dom*:
assumes *img*: $finite\ (f`A)$ **and** *dom*: $infinite\ A$
shows $\exists y \in f`A.\ infinite\ (f\ -`\ \{y\})$
⟨*proof*⟩

lemma *inf-img-fin-domE*:
assumes $finite\ (f`A)$ **and** $infinite\ A$
obtains y **where** $y \in f`A$ **and** $infinite\ (f\ -`\ \{y\})$
⟨*proof*⟩

9.2 Infinitely Many and Almost All

We often need to reason about the existence of infinitely many (resp., all but finitely many) objects satisfying some predicate, so we introduce corresponding binders and their proof rules.

definition
Inf-many :: $('a \Rightarrow bool) \Rightarrow bool$ (**binder** *INFM 10*) **where**
Inf-many P = *infinite* $\{x.\ P\ x\}$

definition
Alm-all :: $('a \Rightarrow bool) \Rightarrow bool$ (**binder** *MOST 10*) **where**
Alm-all $P = (\neg\ (INFM\ x.\ \neg\ P\ x))$

notation (*xsymbols*)
Inf-many (**binder** $\exists_\infty$ *10*) **and**
Alm-all (**binder** $\forall_\infty$ *10*)

notation (*HTML* **output**)
Inf-many (**binder** $\exists_\infty$ *10*) **and**
Alm-all (**binder** $\forall_\infty$ *10*)

lemma *INFM-iff-infinite*: $(INFM\ x.\ P\ x) \longleftrightarrow$ *infinite* $\{x.\ P\ x\}$
⟨*proof*⟩

lemma *MOST-iff-cofinite*: $(MOST\ x.\ P\ x) \longleftrightarrow$ *finite* $\{x.\ \neg\ P\ x\}$
⟨*proof*⟩

lemmas *MOST-iff-finiteNeg* = *MOST-iff-cofinite*

lemma *not-INFM* [*simp*]: $\neg\ (INFM\ x.\ P\ x) \longleftrightarrow (MOST\ x.\ \neg\ P\ x)$
⟨*proof*⟩

lemma *not-MOST* [*simp*]: $\neg\ (MOST\ x.\ P\ x) \longleftrightarrow (INFM\ x.\ \neg\ P\ x)$
⟨*proof*⟩

lemma *INFM-const* [*simp*]: $(INFM\ x{::}'a.\ P) \longleftrightarrow P \wedge$ *infinite* $(UNIV{::}'a\ set)$
⟨*proof*⟩

lemma *MOST-const* [*simp*]: $(MOST\ x{::}'a.\ P) \longleftrightarrow P \vee$ *finite* $(UNIV{::}'a\ set)$
⟨*proof*⟩

lemma *INFM-EX*: $(\exists_\infty x.\ P\ x) \Longrightarrow (\exists\, x.\ P\ x)$
⟨*proof*⟩

lemma *ALL-MOST*: $\forall\, x.\ P\ x \Longrightarrow \forall_\infty x.\ P\ x$
⟨*proof*⟩

lemma *INFM-E*: **assumes** *INFM* $x.\ P\ x$ **obtains** x **where** $P\ x$
⟨*proof*⟩

lemma *MOST-I*: **assumes** $\bigwedge x.\ P\ x$ **shows** *MOST* $x.\ P\ x$
⟨*proof*⟩

lemma *INFM-mono*:
  assumes *inf*: $\exists_\infty x.\ P\ x$ **and** *q*: $\bigwedge x.\ P\ x \Longrightarrow Q\ x$
  shows $\exists_\infty x.\ Q\ x$
⟨*proof*⟩

lemma *MOST-mono*: $\forall_\infty x.\ P\ x \Longrightarrow (\bigwedge x.\ P\ x \Longrightarrow Q\ x) \Longrightarrow \forall_\infty x.\ Q\ x$
  ⟨*proof*⟩

lemma *INFM-disj-distrib*:
  $(\exists_\infty x.\ P\ x \lor Q\ x) \longleftrightarrow (\exists_\infty x.\ P\ x) \lor (\exists_\infty x.\ Q\ x)$
  ⟨*proof*⟩

lemma *INFM-imp-distrib*:
  $(\mathit{INFM}\ x.\ P\ x \longrightarrow Q\ x) \longleftrightarrow ((\mathit{MOST}\ x.\ P\ x) \longrightarrow (\mathit{INFM}\ x.\ Q\ x))$
  ⟨*proof*⟩

lemma *MOST-conj-distrib*:
  $(\forall_\infty x.\ P\ x \land Q\ x) \longleftrightarrow (\forall_\infty x.\ P\ x) \land (\forall_\infty x.\ Q\ x)$
  ⟨*proof*⟩

lemma *MOST-conjI*:
  $\mathit{MOST}\ x.\ P\ x \Longrightarrow \mathit{MOST}\ x.\ Q\ x \Longrightarrow \mathit{MOST}\ x.\ P\ x \land Q\ x$
  ⟨*proof*⟩

lemma *INFM-conjI*:
  $\mathit{INFM}\ x.\ P\ x \Longrightarrow \mathit{MOST}\ x.\ Q\ x \Longrightarrow \mathit{INFM}\ x.\ P\ x \land Q\ x$
  ⟨*proof*⟩

lemma *MOST-rev-mp*:
  assumes $\forall_\infty x.\ P\ x$ **and** $\forall_\infty x.\ P\ x \longrightarrow Q\ x$
  shows $\forall_\infty x.\ Q\ x$
⟨*proof*⟩

lemma *MOST-imp-iff*:
  assumes $\mathit{MOST}\ x.\ P\ x$
  shows $(\mathit{MOST}\ x.\ P\ x \longrightarrow Q\ x) \longleftrightarrow (\mathit{MOST}\ x.\ Q\ x)$
⟨*proof*⟩

lemma *INFM-MOST-simps* [*simp*]:
  $\bigwedge P\ Q.\ (\mathit{INFM}\ x.\ P\ x \land Q) \longleftrightarrow (\mathit{INFM}\ x.\ P\ x) \land Q$
  $\bigwedge P\ Q.\ (\mathit{INFM}\ x.\ P \land Q\ x) \longleftrightarrow P \land (\mathit{INFM}\ x.\ Q\ x)$
  $\bigwedge P\ Q.\ (\mathit{MOST}\ x.\ P\ x \lor Q) \longleftrightarrow (\mathit{MOST}\ x.\ P\ x) \lor Q$
  $\bigwedge P\ Q.\ (\mathit{MOST}\ x.\ P \lor Q\ x) \longleftrightarrow P \lor (\mathit{MOST}\ x.\ Q\ x)$
  $\bigwedge P\ Q.\ (\mathit{MOST}\ x.\ P\ x \longrightarrow Q) \longleftrightarrow ((\mathit{INFM}\ x.\ P\ x) \longrightarrow Q)$
  $\bigwedge P\ Q.\ (\mathit{MOST}\ x.\ P \longrightarrow Q\ x) \longleftrightarrow (P \longrightarrow (\mathit{MOST}\ x.\ Q\ x))$
  ⟨*proof*⟩

Properties of quantifiers with injective functions.

lemma *INFM-inj*:
$INFM\ x.\ P\ (f\ x) \Longrightarrow inj\ f \Longrightarrow INFM\ x.\ P\ x$
⟨*proof*⟩

lemma *MOST-inj*:
$MOST\ x.\ P\ x \Longrightarrow inj\ f \Longrightarrow MOST\ x.\ P\ (f\ x)$
⟨*proof*⟩

Properties of quantifiers with singletons.

lemma *not-INFM-eq* [*simp*]:
$\neg\ (INFM\ x.\ x = a)$
$\neg\ (INFM\ x.\ a = x)$
⟨*proof*⟩

lemma *MOST-neq* [*simp*]:
$MOST\ x.\ x \neq a$
$MOST\ x.\ a \neq x$
⟨*proof*⟩

lemma *INFM-neq* [*simp*]:
$(INFM\ x{::}'a.\ x \neq a) \longleftrightarrow infinite\ (UNIV{::}'a\ set)$
$(INFM\ x{::}'a.\ a \neq x) \longleftrightarrow infinite\ (UNIV{::}'a\ set)$
⟨*proof*⟩

lemma *MOST-eq* [*simp*]:
$(MOST\ x{::}'a.\ x = a) \longleftrightarrow finite\ (UNIV{::}'a\ set)$
$(MOST\ x{::}'a.\ a = x) \longleftrightarrow finite\ (UNIV{::}'a\ set)$
⟨*proof*⟩

lemma *MOST-eq-imp*:
$MOST\ x.\ x = a \longrightarrow P\ x$
$MOST\ x.\ a = x \longrightarrow P\ x$
⟨*proof*⟩

Properties of quantifiers over the naturals.

lemma *INFM-nat*: $(\exists_\infty n.\ P\ (n{::}nat)) = (\forall m.\ \exists n.\ m<n \wedge P\ n)$
⟨*proof*⟩

lemma *INFM-nat-le*: $(\exists_\infty n.\ P\ (n{::}nat)) = (\forall m.\ \exists n.\ m\leq n \wedge P\ n)$
⟨*proof*⟩

lemma *MOST-nat*: $(\forall_\infty n.\ P\ (n{::}nat)) = (\exists m.\ \forall n.\ m<n \longrightarrow P\ n)$
⟨*proof*⟩

lemma *MOST-nat-le*: $(\forall_\infty n.\ P\ (n{::}nat)) = (\exists m.\ \forall n.\ m\leq n \longrightarrow P\ n)$
⟨*proof*⟩

9.3 Enumeration of an Infinite Set

The set's element type must be wellordered (e.g. the natural numbers).

primrec (**in** *wellorder*) *enumerate* :: *'a set* ⇒ *nat* ⇒ *'a* **where**
  enumerate-0: *enumerate S 0* = (*LEAST n*. $n \in S$)
| *enumerate-Suc*: *enumerate S* (*Suc n*) = *enumerate* ($S - \{$*LEAST n*. $n \in S\}$) *n*

lemma *enumerate-Suc'*:
  enumerate S (*Suc n*) = *enumerate* ($S - \{$*enumerate S 0*$\}$) *n*
⟨*proof*⟩

lemma *enumerate-in-set*: *infinite S* ⟹ *enumerate S n* : *S*
⟨*proof*⟩

declare *enumerate-0* [*simp del*] *enumerate-Suc* [*simp del*]

lemma *enumerate-step*: *infinite S* ⟹ *enumerate S n* < *enumerate S* (*Suc n*)
⟨*proof*⟩

lemma *enumerate-mono*: $m<n$ ⟹ *infinite S* ⟹ *enumerate S m* < *enumerate S n*
⟨*proof*⟩

9.4 Miscellaneous

A few trivial lemmas about sets that contain at most one element. These simplify the reasoning about deterministic automata.

definition
  atmost-one :: *'a set* ⇒ *bool* **where**
  atmost-one S = ($\forall x\ y.\ x \in S \wedge y \in S \longrightarrow x = y$)

lemma *atmost-one-empty*: $S = \{\}$ ⟹ *atmost-one S*
⟨*proof*⟩

lemma *atmost-one-singleton*: $S = \{x\}$ ⟹ *atmost-one S*
⟨*proof*⟩

lemma *atmost-one-unique* [*elim*]: *atmost-one S* ⟹ $x \in S$ ⟹ $y \in S$ ⟹ $y = x$
⟨*proof*⟩

end

10 Product-plus: Additive group operations on product types

theory *Product-plus*
imports *Main*
begin

10.1 Operations

instantiation $* :: (zero, zero)\ zero$
begin

definition *zero-prod-def*: $0 = (0,\ 0)$

instance ⟨*proof*⟩
end

instantiation $* :: (plus, plus)\ plus$
begin

definition *plus-prod-def*:
$x + y = (fst\ x + fst\ y,\ snd\ x + snd\ y)$

instance ⟨*proof*⟩
end

instantiation $* :: (minus, minus)\ minus$
begin

definition *minus-prod-def*:
$x - y = (fst\ x - fst\ y,\ snd\ x - snd\ y)$

instance ⟨*proof*⟩
end

instantiation $* :: (uminus, uminus)\ uminus$
begin

definition *uminus-prod-def*:
$- x = (- fst\ x,\ - snd\ x)$

instance ⟨*proof*⟩
end

lemma *fst-zero* [*simp*]: $fst\ 0 = 0$
⟨*proof*⟩

lemma *snd-zero* [*simp*]: $snd\ 0 = 0$
⟨*proof*⟩

lemma *fst-add* [*simp*]: *fst* $(x + y) =$ *fst* $x +$ *fst* y
⟨*proof*⟩

lemma *snd-add* [*simp*]: *snd* $(x + y) =$ *snd* $x +$ *snd* y
⟨*proof*⟩

lemma *fst-diff* [*simp*]: *fst* $(x - y) =$ *fst* $x -$ *fst* y
⟨*proof*⟩

lemma *snd-diff* [*simp*]: *snd* $(x - y) =$ *snd* $x -$ *snd* y
⟨*proof*⟩

lemma *fst-uminus* [*simp*]: *fst* $(- x) = -$ *fst* x
⟨*proof*⟩

lemma *snd-uminus* [*simp*]: *snd* $(- x) = -$ *snd* x
⟨*proof*⟩

lemma *add-Pair* [*simp*]: $(a, b) + (c, d) = (a + c, b + d)$
⟨*proof*⟩

lemma *diff-Pair* [*simp*]: $(a, b) - (c, d) = (a - c, b - d)$
⟨*proof*⟩

lemma *uminus-Pair* [*simp*, *code*]: $- (a, b) = (- a, - b)$
⟨*proof*⟩

lemmas *expand-prod-eq* = *Pair-fst-snd-eq*

10.2 Class instances

instance $*$ :: (*semigroup-add*, *semigroup-add*) *semigroup-add*
⟨*proof*⟩

instance $*$ :: (*ab-semigroup-add*, *ab-semigroup-add*) *ab-semigroup-add*
⟨*proof*⟩

instance $*$ :: (*monoid-add*, *monoid-add*) *monoid-add*
⟨*proof*⟩

instance $*$ :: (*comm-monoid-add*, *comm-monoid-add*) *comm-monoid-add*
⟨*proof*⟩

instance $*$ ::
(*cancel-semigroup-add*, *cancel-semigroup-add*) *cancel-semigroup-add*
⟨*proof*⟩

instance $*$ ::

(*cancel-ab-semigroup-add*, *cancel-ab-semigroup-add*) *cancel-ab-semigroup-add*
⟨*proof*⟩

instance $*$ ::
(*cancel-comm-monoid-add*, *cancel-comm-monoid-add*) *cancel-comm-monoid-add*
⟨*proof*⟩

instance $*$ :: (*group-add*, *group-add*) *group-add*
⟨*proof*⟩

instance $*$ :: (*ab-group-add*, *ab-group-add*) *ab-group-add*
⟨*proof*⟩

lemma *fst-setsum*: *fst* ($\sum x{\in}A.\ f\ x$) = ($\sum x{\in}A.\ fst\ (f\ x)$)
⟨*proof*⟩

lemma *snd-setsum*: *snd* ($\sum x{\in}A.\ f\ x$) = ($\sum x{\in}A.\ snd\ (f\ x)$)
⟨*proof*⟩

end

11 Product-Vector: Cartesian Products as Vector Spaces

theory *Product-Vector*
imports *Inner-Product Product-plus*
begin

11.1 Product is a real vector space

instantiation $*$ :: (*real-vector*, *real-vector*) *real-vector*
begin

definition *scaleR-prod-def*:
scaleR r A = (*scaleR r* (*fst A*), *scaleR r* (*snd A*))

lemma *fst-scaleR* [*simp*]: *fst* (*scaleR r A*) = *scaleR r* (*fst A*)
⟨*proof*⟩

lemma *snd-scaleR* [*simp*]: *snd* (*scaleR r A*) = *scaleR r* (*snd A*)
⟨*proof*⟩

lemma *scaleR-Pair* [*simp*]: *scaleR r* (*a*, *b*) = (*scaleR r a*, *scaleR r b*)
⟨*proof*⟩

instance ⟨*proof*⟩

end

11.2 Product is a topological space

instantiation
  $* :: (\textit{topological-space}, \textit{topological-space})\ \textit{topological-space}$
begin

definition *open-prod-def*:
  $\textit{open}\ (S :: ('a \times 'b)\ \textit{set}) \longleftrightarrow$
    $(\forall x{\in}S.\ \exists A\ B.\ \textit{open}\ A \wedge \textit{open}\ B \wedge x \in A \times B \wedge A \times B \subseteq S)$

lemma *open-prod-elim*:
  assumes $\textit{open}\ S$ **and** $x \in S$
  obtains $A\ B$ **where** $\textit{open}\ A$ **and** $\textit{open}\ B$ **and** $x \in A \times B$ **and** $A \times B \subseteq S$
$\langle \textit{proof} \rangle$

lemma *open-prod-intro*:
  assumes $\bigwedge x.\ x \in S \Longrightarrow \exists A\ B.\ \textit{open}\ A \wedge \textit{open}\ B \wedge x \in A \times B \wedge A \times B \subseteq S$
  shows $\textit{open}\ S$
$\langle \textit{proof} \rangle$

instance $\langle \textit{proof} \rangle$

end

lemma *open-Times*: $\textit{open}\ S \Longrightarrow \textit{open}\ T \Longrightarrow \textit{open}\ (S \times T)$
$\langle \textit{proof} \rangle$

lemma *fst-vimage-eq-Times*: $\textit{fst}\ -`\ S = S \times \textit{UNIV}$
$\langle \textit{proof} \rangle$

lemma *snd-vimage-eq-Times*: $\textit{snd}\ -`\ S = \textit{UNIV} \times S$
$\langle \textit{proof} \rangle$

lemma *open-vimage-fst*: $\textit{open}\ S \Longrightarrow \textit{open}\ (\textit{fst}\ -`\ S)$
$\langle \textit{proof} \rangle$

lemma *open-vimage-snd*: $\textit{open}\ S \Longrightarrow \textit{open}\ (\textit{snd}\ -`\ S)$
$\langle \textit{proof} \rangle$

lemma *closed-vimage-fst*: $\textit{closed}\ S \Longrightarrow \textit{closed}\ (\textit{fst}\ -`\ S)$
$\langle \textit{proof} \rangle$

lemma *closed-vimage-snd*: $\textit{closed}\ S \Longrightarrow \textit{closed}\ (\textit{snd}\ -`\ S)$
$\langle \textit{proof} \rangle$

lemma *closed-Times*: $\textit{closed}\ S \Longrightarrow \textit{closed}\ T \Longrightarrow \textit{closed}\ (S \times T)$
$\langle \textit{proof} \rangle$

lemma *openI*:
assumes $\bigwedge x.\ x \in S \Longrightarrow \exists\, T.\ open\ T \wedge x \in T \wedge T \subseteq S$
shows *open S*
⟨*proof*⟩

lemma *subset-fst-imageI*: $A \times B \subseteq S \Longrightarrow y \in B \Longrightarrow A \subseteq fst\ `\ S$
⟨*proof*⟩

lemma *subset-snd-imageI*: $A \times B \subseteq S \Longrightarrow x \in A \Longrightarrow B \subseteq snd\ `\ S$
⟨*proof*⟩

lemma *open-image-fst*: **assumes** *open S* **shows** *open (fst ` S)*
⟨*proof*⟩

lemma *open-image-snd*: **assumes** *open S* **shows** *open (snd ` S)*
⟨*proof*⟩

11.3 Product is a metric space

instantiation
$*$ :: (*metric-space*, *metric-space*) *metric-space*
begin

definition *dist-prod-def*:
$dist\ (x{::}'a \times 'b)\ y = sqrt\ ((dist\ (fst\ x)\ (fst\ y))^2 + (dist\ (snd\ x)\ (snd\ y))^2)$

lemma *dist-Pair-Pair*: $dist\ (a,\ b)\ (c,\ d) = sqrt\ ((dist\ a\ c)^2 + (dist\ b\ d)^2)$
⟨*proof*⟩

lemma *dist-fst-le*: $dist\ (fst\ x)\ (fst\ y) \leq dist\ x\ y$
⟨*proof*⟩

lemma *dist-snd-le*: $dist\ (snd\ x)\ (snd\ y) \leq dist\ x\ y$
⟨*proof*⟩

instance ⟨*proof*⟩

end

11.4 Continuity of operations

lemma *tendsto-fst* [*tendsto-intros*]:
assumes $(f\ {-}{-}{-}{>}\ a)\ net$
shows $((\lambda x.\ fst\ (f\ x))\ {-}{-}{-}{>}\ fst\ a)\ net$
⟨*proof*⟩

lemma *tendsto-snd* [*tendsto-intros*]:
assumes $(f\ {-}{-}{-}{>}\ a)\ net$
shows $((\lambda x.\ snd\ (f\ x))\ {-}{-}{-}{>}\ snd\ a)\ net$

⟨*proof*⟩

lemma *tendsto-Pair* [*tendsto-intros*]:
  assumes $(f \dashrightarrow a)$ *net* **and** $(g \dashrightarrow b)$ *net*
  shows $((\lambda x.\ (f\ x,\ g\ x)) \dashrightarrow (a,\ b))$ *net*
⟨*proof*⟩

lemma *Cauchy-fst*: *Cauchy* $X \Longrightarrow$ *Cauchy* $(\lambda n.\ fst\ (X\ n))$
⟨*proof*⟩

lemma *Cauchy-snd*: *Cauchy* $X \Longrightarrow$ *Cauchy* $(\lambda n.\ snd\ (X\ n))$
⟨*proof*⟩

lemma *Cauchy-Pair*:
  assumes *Cauchy* X **and** *Cauchy* Y
  shows *Cauchy* $(\lambda n.\ (X\ n,\ Y\ n))$
⟨*proof*⟩

lemma *isCont-Pair* [*simp*]:
  $[\![isCont\ f\ x;\ isCont\ g\ x]\!] \Longrightarrow isCont\ (\lambda x.\ (f\ x,\ g\ x))\ x$
  ⟨*proof*⟩

11.5 Product is a complete metric space

instance $*$ :: (*complete-space*, *complete-space*) *complete-space*
⟨*proof*⟩

11.6 Product is a normed vector space

instantiation
  $*$ :: (*real-normed-vector*, *real-normed-vector*) *real-normed-vector*
begin

definition *norm-prod-def*:
  $norm\ x = sqrt\ ((norm\ (fst\ x))^2 + (norm\ (snd\ x))^2)$

definition *sgn-prod-def*:
  $sgn\ (x::'a \times 'b) = scaleR\ (inverse\ (norm\ x))\ x$

lemma *norm-Pair*: $norm\ (a,\ b) = sqrt\ ((norm\ a)^2 + (norm\ b)^2)$
  ⟨*proof*⟩

instance ⟨*proof*⟩

end

instance $*$ :: (*banach*, *banach*) *banach* ⟨*proof*⟩

11.7 Product is an inner product space

instantiation $*$ :: (*real-inner*, *real-inner*) *real-inner*
begin

definition *inner-prod-def*:
inner x y = *inner* (*fst x*) (*fst y*) + *inner* (*snd x*) (*snd y*)

lemma *inner-Pair* [*simp*]: *inner* (*a*, *b*) (*c*, *d*) = *inner a c* + *inner b d*
⟨*proof*⟩

instance ⟨*proof*⟩

end

11.8 Pair operations are linear

interpretation *fst*: *bounded-linear fst*
⟨*proof*⟩

interpretation *snd*: *bounded-linear snd*
⟨*proof*⟩

TODO: move to NthRoot

lemma *sqrt-add-le-add-sqrt*:
assumes *x*: $0 \leq x$ **and** *y*: $0 \leq y$
shows *sqrt* $(x + y) \leq$ *sqrt* x + *sqrt* y
⟨*proof*⟩

lemma *bounded-linear-Pair*:
assumes *f*: *bounded-linear f*
assumes *g*: *bounded-linear g*
shows *bounded-linear* (λx. (*f x*, *g x*))
⟨*proof*⟩

11.9 Frechet derivatives involving pairs

lemma *FDERIV-Pair*:
assumes *f*: *FDERIV f x* :> f' **and** *g*: *FDERIV g x* :> g'
shows *FDERIV* (λx. (*f x*, *g x*)) *x* :> (λh. (f' *h*, g' *h*))
⟨*proof*⟩

end

12 Convex: Convexity in real vector spaces

theory *Convex*
imports *Product-Vector*
begin

12.1 Convexity.

definition
convex :: $'a$::*real-vector set* $\Rightarrow$ *bool* **where**
convex s $\longleftrightarrow$ ($\forall x{\in}s.\ \forall y{\in}s.\ \forall u{\geq}0.\ \forall v{\geq}0.\ u + v = 1 \longrightarrow u *_R x + v *_R y \in s$)

lemma *convex-alt*:
convex s $\longleftrightarrow$ ($\forall x{\in}s.\ \forall y{\in}s.\ \forall u.\ 0 \leq u \wedge u \leq 1 \longrightarrow ((1 - u) *_R x + u *_R y) \in s$)
(**is** - $\longleftrightarrow$ *?alt*)
⟨*proof*⟩

lemma *mem-convex*:
assumes *convex s* $a \in s$ $b \in s$ $0 \leq u$ $u \leq 1$
shows $((1 - u) *_R a + u *_R b) \in s$
⟨*proof*⟩

lemma *convex-empty*[*intro*]: *convex* {}
⟨*proof*⟩

lemma *convex-singleton*[*intro*]: *convex* {*a*}
⟨*proof*⟩

lemma *convex-UNIV*[*intro*]: *convex UNIV*
⟨*proof*⟩

lemma *convex-Inter*: ($\forall s{\in}f.$ *convex s*) ==> *convex*($\bigcap$ *f*)
⟨*proof*⟩

lemma *convex-Int*: *convex s* $\Longrightarrow$ *convex t* $\Longrightarrow$ *convex* ($s \cap t$)
⟨*proof*⟩

lemma *convex-halfspace-le*: *convex* {*x*. *inner a x* $\leq$ *b*}
⟨*proof*⟩

lemma *convex-halfspace-ge*: *convex* {*x*. *inner a x* $\geq$ *b*}
⟨*proof*⟩

lemma *convex-hyperplane*: *convex* {*x*. *inner a x* = *b*}
⟨*proof*⟩

lemma *convex-halfspace-lt*: *convex* {*x*. *inner a x* < *b*}
⟨*proof*⟩

lemma *convex-halfspace-gt*: *convex* {*x*. *inner a x* > *b*}
⟨*proof*⟩

lemma *convex-real-interval*:
fixes *a b* :: *real*

shows *convex* $\{a..\}$ **and** *convex* $\{..b\}$
and *convex* $\{a<..\}$ **and** *convex* $\{..<b\}$
and *convex* $\{a..b\}$ **and** *convex* $\{a<..b\}$
and *convex* $\{a..<b\}$ **and** *convex* $\{a<..<b\}$
$\langle proof \rangle$

12.2 Explicit expressions for convexity in terms of arbitrary sums.

lemma *convex-setsum*:
fixes C :: *'a::real-vector set*
assumes *finite s* **and** *convex C* **and** $(\sum\ i \in s.\ a\ i) = 1$
assumes $\bigwedge i.\ i \in s \Longrightarrow a\ i \geq 0$ **and** $\bigwedge i.\ i \in s \Longrightarrow y\ i \in C$
shows $(\sum\ j \in s.\ a\ j *_R y\ j) \in C$
$\langle proof \rangle$

lemma *convex*:
shows *convex s* $\longleftrightarrow (\forall (k::nat)\ u\ x.\ (\forall i.\ 1 \leq i \wedge i \leq k \longrightarrow 0 \leq u\ i \wedge x\ i \in s) \wedge$ $(setsum\ u\ \{1..k\} = 1)$
$\longrightarrow setsum\ (\lambda i.\ u\ i *_R x\ i)\ \{1..k\} \in s)$
$\langle proof \rangle$

lemma *convex-explicit*:
fixes *s* :: *'a::real-vector set*
shows *convex s* $\longleftrightarrow$
$(\forall t\ u.\ finite\ t \wedge t \subseteq s \wedge (\forall x \in t.\ 0 \leq u\ x) \wedge setsum\ u\ t = 1 \longrightarrow setsum\ (\lambda x.\ u\ x *_R x)\ t \in s)$
$\langle proof \rangle$

lemma *convex-finite*: **assumes** *finite s*
shows *convex s* $\longleftrightarrow (\forall u.\ (\forall x \in s.\ 0 \leq u\ x) \wedge setsum\ u\ s = 1$
$\longrightarrow setsum\ (\lambda x.\ u\ x *_R x)\ s \in s)$
$\langle proof \rangle$

definition
convex-on :: *'a::real-vector set* $\Rightarrow$ *('a* $\Rightarrow$ *real)* $\Rightarrow$ *bool* **where**
convex-on s f $\longleftrightarrow$
$(\forall x \in s.\ \forall y \in s.\ \forall u \geq 0.\ \forall v \geq 0.\ u + v = 1 \longrightarrow f\ (u *_R x + v *_R y) \leq u * f\ x + v * f\ y)$

lemma *convex-on-subset*: *convex-on t f* $\Longrightarrow s \subseteq t \Longrightarrow$ *convex-on s f*
$\langle proof \rangle$

lemma *convex-add*[*intro*]:
assumes *convex-on s f convex-on s g*
shows *convex-on s* $(\lambda x.\ f\ x + g\ x)$
$\langle proof \rangle$

lemma *convex-cmul*[*intro*]:
  assumes $0 \leq (c{::}real)$ *convex-on s f*
  shows *convex-on s* $(\lambda x.\ c * f\ x)$
⟨*proof*⟩

lemma *convex-lower*:
  assumes *convex-on s f* $x{\in}s$ $y \in s$ $0 \leq u$ $0 \leq v$ $u + v = 1$
  shows $f\ (u *_R x + v *_R y) \leq max\ (f\ x)\ (f\ y)$
⟨*proof*⟩

lemma *convex-distance*[*intro*]:
  fixes $s :: 'a{::}real\text{-}normed\text{-}vector\ set$
  shows *convex-on s* $(\lambda x.\ dist\ a\ x)$
⟨*proof*⟩

12.3 Arithmetic operations on sets preserve convexity.

lemma *convex-scaling*:
  assumes *convex s*
  shows*convex* $((\lambda x.\ c *_R x)$ ‘ $s)$
⟨*proof*⟩

lemma *convex-negations*: *convex s* $\Longrightarrow$ *convex* $((\lambda x.\ -x)$‘ $s)$
⟨*proof*⟩

lemma *convex-sums*:
  assumes *convex s convex t*
  shows *convex* $\{x + y|\ x\ y.\ x \in s \wedge y \in t\}$
⟨*proof*⟩

lemma *convex-differences*:
  assumes *convex s convex t*
  shows *convex* $\{x - y|\ x\ y.\ x \in s \wedge y \in t\}$
⟨*proof*⟩

lemma *convex-translation*: **assumes** *convex s* **shows** *convex* $((\lambda x.\ a + x)$ ‘ $s)$
⟨*proof*⟩

lemma *convex-affinity*: **assumes** *convex s* **shows** *convex* $((\lambda x.\ a + c *_R x)$ ‘ $s)$
⟨*proof*⟩

lemma *convex-linear-image*:
  assumes *c*:*convex s* **and** *l*:*bounded-linear f*
  shows *convex*$(f$ ‘ $s)$
⟨*proof*⟩

lemma *pos-is-convex*:
  shows *convex* $\{0 :: real <..\}$

⟨*proof*⟩

lemma *convex-on-setsum*:
  fixes a :: $'a \Rightarrow real$
  fixes y :: $'a \Rightarrow 'b$::*real-vector*
  fixes f :: $'b \Rightarrow real$
  assumes *finite* s $s \neq \{\}$
  assumes *convex-on* C f
  assumes *convex* C
  assumes $(\sum\ i \in s.\ a\ i) = 1$
  assumes $\bigwedge i.\ i \in s \Longrightarrow a\ i \geq 0$
  assumes $\bigwedge i.\ i \in s \Longrightarrow y\ i \in C$
  shows $f\ (\sum\ i \in s.\ a\ i *_R\ y\ i) \leq (\sum\ i \in s.\ a\ i * f\ (y\ i))$
⟨*proof*⟩

lemma *convex-on-alt*:
  fixes C :: $'a$::*real-vector set*
  assumes *convex* C
  shows *convex-on* C f =
  $(\forall\ x \in C.\ \forall\ y \in C.\ \forall\ \mu :: real.\ \mu \geq 0 \wedge \mu \leq 1$
    $\longrightarrow f\ (\mu *_R\ x + (1 - \mu) *_R\ y) \leq \mu * f\, x + (1 - \mu) * f\, y)$
⟨*proof*⟩

lemma *pos-convex-function*:
  fixes f :: $real \Rightarrow real$
  assumes *convex* C
  assumes *leq*: $\bigwedge x\ y.\ [\![x \in C\ ;\ y \in C]\!] \Longrightarrow f'\ x * (y - x) \leq f\ y - f\ x$
  shows *convex-on* C f
⟨*proof*⟩

lemma *atMostAtLeast-subset-convex*:
  fixes C :: *real set*
  assumes *convex* C
  assumes $x \in C$ $y \in C$ $x < y$
  shows $\{x\ ..\ y\} \subseteq C$
⟨*proof*⟩

lemma *f''-imp-f'*:
  fixes f :: $real \Rightarrow real$
  assumes *convex* C
  assumes f': $\bigwedge x.\ x \in C \Longrightarrow DERIV\ f\ x :> (f'\ x)$
  assumes f'': $\bigwedge x.\ x \in C \Longrightarrow DERIV\ f'\ x :> (f''\ x)$
  assumes *pos*: $\bigwedge x.\ x \in C \Longrightarrow f''\ x \geq 0$
  assumes $x \in C$ $y \in C$
  shows $f'\ x * (y - x) \leq f\ y - f\ x$
⟨*proof*⟩

lemma *f''-ge0-imp-convex*:

fixes f :: *real* $\Rightarrow$ *real*
assumes *conv*: *convex C*
assumes f': $\bigwedge x.\ x \in C \Longrightarrow DERIV\ f\ x :> (f'\ x)$
assumes f'': $\bigwedge x.\ x \in C \Longrightarrow DERIV\ f'\ x :> (f''\ x)$
assumes *pos*: $\bigwedge x.\ x \in C \Longrightarrow f''\ x \geq 0$
shows *convex-on C f*
⟨*proof*⟩

lemma *minus-log-convex*:
fixes b :: *real*
assumes $b > 1$
shows *convex-on* $\{0 <..\}$ $(\lambda\ x.\ -\ log\ b\ x)$
⟨*proof*⟩

end

13 Euclidean-Space: (Real) Vectors in Euclidean space, and elementary linear algebra.

theory *Euclidean-Space*
imports
Complex-Main ~~/src/HOL/Decision-Procs/Dense-Linear-Order
Finite-Cartesian-Product Infinite-Set Numeral-Type
Inner-Product L2-Norm Convex
uses *positivstellensatz.ML* (*normarith.ML*)
begin

13.1 Basic componentwise operations on vectors.

instantiation *cart* :: (*times*,*finite*) *times*
begin
definition *vector-mult-def* : $op * \equiv (\lambda\ x\ y.\ \ (\chi\ i.\ (x\$i) * (y\$i)))$
instance ⟨*proof*⟩
end

instantiation *cart* :: (*one*,*finite*) *one*
begin
definition *vector-one-def* : $1 \equiv (\chi\ i.\ 1)$
instance ⟨*proof*⟩
end

instantiation *cart* :: (*ord*,*finite*) *ord*
begin
definition *vector-le-def*:
less-eq $(x :: 'a\ \hat{}\ 'b)\ y = (ALL\ i.\ x\$i <= y\$i)$
definition *vector-less-def*: *less* $(x :: 'a\ \hat{}\ 'b)\ y = (ALL\ i.\ x\$i < y\$i)$
instance ⟨*proof*⟩

end

The ordering on one-dimensional vectors is linear.

class *cart-one* = **assumes** *UNIV-one*: *card* (*UNIV* :: *'a set*) = *Suc 0*
begin
  subclass *finite*
  ⟨*proof*⟩
end

instantiation *cart* :: (*linorder*,*cart-one*) *linorder* **begin**
instance ⟨*proof*⟩ **end**

Also the scalar-vector multiplication.

definition *vector-scalar-mult*:: *'a*::*times* ⇒ *'a* ^ *'n* ⇒ *'a* ^ *'n* (**infixl** **s 70*)
  where $c *s\ x = (\chi\ i.\ c * (x\$i))$

Constant Vectors

definition *vec* $x = (\chi\ i.\ x)$

13.2 A naive proof procedure to lift really trivial arithmetic stuff from the basis of the vector space.

⟨*ML*⟩

lemma *vec-0*[*simp*]: *vec* $0 = 0$ ⟨*proof*⟩
lemma *vec-1*[*simp*]: *vec* $1 = 1$ ⟨*proof*⟩

Obvious "component-pushing".

lemma *vec-component* [*simp*]: *vec* $x\ \$\ i = x$
  ⟨*proof*⟩

lemma *vector-mult-component* [*simp*]: $(x * y)\$i = x\$i * y\$i$
  ⟨*proof*⟩

lemma *vector-smult-component* [*simp*]: $(c *s\ y)\$i = c * (y\$i)$
  ⟨*proof*⟩

lemma *cond-component*: (*if b then x else y*)$i = (*if b then x*$i *else y*$i) ⟨*proof*⟩

lemmas *vector-component* =
  vec-component vector-add-component vector-mult-component
  vector-smult-component vector-minus-component vector-uminus-component
  vector-scaleR-component cond-component

13.3 Some frequently useful arithmetic lemmas over vectors.

instance *cart* :: (*semigroup-mult*,*finite*) *semigroup-mult*
  ⟨*proof*⟩

instance *cart* :: (*monoid-mult*,*finite*) *monoid-mult*
⟨*proof*⟩

instance *cart* :: (*ab-semigroup-mult*,*finite*) *ab-semigroup-mult*
⟨*proof*⟩

instance *cart* :: (*ab-semigroup-idem-mult*,*finite*) *ab-semigroup-idem-mult*
⟨*proof*⟩

instance *cart* :: (*comm-monoid-mult*,*finite*) *comm-monoid-mult*
⟨*proof*⟩

instance *cart* :: (*semiring*,*finite*) *semiring*
⟨*proof*⟩

instance *cart* :: (*semiring-0*,*finite*) *semiring-0*
⟨*proof*⟩
instance *cart* :: (*semiring-1*,*finite*) *semiring-1*
⟨*proof*⟩
instance *cart* :: (*comm-semiring*,*finite*) *comm-semiring*
⟨*proof*⟩

instance *cart* :: (*comm-semiring-0*,*finite*) *comm-semiring-0* ⟨*proof*⟩
instance *cart* :: (*cancel-comm-monoid-add*, *finite*) *cancel-comm-monoid-add* ⟨*proof*⟩
instance *cart* :: (*semiring-0-cancel*,*finite*) *semiring-0-cancel* ⟨*proof*⟩
instance *cart* :: (*comm-semiring-0-cancel*,*finite*) *comm-semiring-0-cancel* ⟨*proof*⟩
instance *cart* :: (*ring*,*finite*) *ring* ⟨*proof*⟩
instance *cart* :: (*semiring-1-cancel*,*finite*) *semiring-1-cancel* ⟨*proof*⟩
instance *cart* :: (*comm-semiring-1*,*finite*) *comm-semiring-1* ⟨*proof*⟩

instance *cart* :: (*ring-1*,*finite*) *ring-1* ⟨*proof*⟩

instance *cart* :: (*real-algebra*,*finite*) *real-algebra*
⟨*proof*⟩

instance *cart* :: (*real-algebra-1*,*finite*) *real-algebra-1* ⟨*proof*⟩

lemma *of-nat-index*:
(*of-nat n* :: *'a*::*semiring-1* ^*'n*)$*i* = *of-nat n*
⟨*proof*⟩

lemma *one-index*[*simp*]:
(*1* :: *'a*::*one* ^*'n*)$*i* = *1* ⟨*proof*⟩

instance *cart* :: (*semiring-char-0*,*finite*) *semiring-char-0*
⟨*proof*⟩

instance *cart* :: (*comm-ring-1*,*finite*) *comm-ring-1* ⟨*proof*⟩

instance *cart* :: (*ring-char-0*,*finite*) *ring-char-0* ⟨*proof*⟩

lemma *vector-smult-assoc*: $a *s (b *s x) = ((a::'a::semigroup\text{-}mult) * b) *s x$
⟨*proof*⟩
lemma *vector-sadd-rdistrib*: $((a::'a::semiring) + b) *s x = a *s x + b *s x$
⟨*proof*⟩
lemma *vector-add-ldistrib*: $(c::'a::semiring) *s (x + y) = c *s x + c *s y$
⟨*proof*⟩
lemma *vector-smult-lzero*[*simp*]: $(0::'a::mult\text{-}zero) *s x = 0$ ⟨*proof*⟩
lemma *vector-smult-lid*[*simp*]: $(1::'a::monoid\text{-}mult) *s x = x$ ⟨*proof*⟩
lemma *vector-ssub-ldistrib*: $(c::'a::ring) *s (x - y) = c *s x - c *s y$
⟨*proof*⟩
lemma *vector-smult-rneg*: $(c::'a::ring) *s -x = -(c *s x)$ ⟨*proof*⟩
lemma *vector-smult-lneg*: $- (c::'a::ring) *s x = -(c *s x)$ ⟨*proof*⟩
lemma *vector-sneg-minus1*: $-x = (- (1::'a::ring\text{-}1)) *s x$ ⟨*proof*⟩
lemma *vector-smult-rzero*[*simp*]: $c *s 0 = (0::'a::mult\text{-}zero \mathbin{\hat{}} 'n)$ ⟨*proof*⟩
lemma *vector-sub-rdistrib*: $((a::'a::ring) - b) *s x = a *s x - b *s x$
⟨*proof*⟩

lemma *vec-eq*[*simp*]: $(vec\ m = vec\ n) \longleftrightarrow (m = n)$
⟨*proof*⟩

abbreviation *inner-bullet* (**infix** · *70*) **where** $x \cdot y \equiv inner\ x\ y$

13.4 A connectedness or intermediate value lemma with several applications.

lemma *connected-real-lemma*:
fixes $f :: real \Rightarrow 'a::metric\text{-}space$
assumes *ab*: $a \le b$ **and** *fa*: $f\ a \in e1$ **and** *fb*: $f\ b \in e2$
and *dst*: $\bigwedge e\ x.\ a <= x \Longrightarrow x <= b \Longrightarrow 0 < e ==> \exists d > 0.\ \forall y.\ abs(y - x) < d \longrightarrow dist(f\ y)\ (f\ x) < e$
and *e1*: $\forall y \in e1.\ \exists e > 0.\ \forall y'.\ dist\ y'\ y < e \longrightarrow y' \in e1$
and *e2*: $\forall y \in e2.\ \exists e > 0.\ \forall y'.\ dist\ y'\ y < e \longrightarrow y' \in e2$
and *e12*: $\sim(\exists x \ge a.\ x <= b \wedge f\ x \in e1 \wedge f\ x \in e2)$
shows $\exists x \ge a.\ x <= b \wedge f\ x \notin e1 \wedge f\ x \notin e2$ (**is** $\exists\ x.\ ?P\ x$)
⟨*proof*⟩

One immediately useful corollary is the existence of square roots! — Should help to get rid of all the development of square-root for reals as a special case

lemma *square-bound-lemma*: $(x::real) < (1 + x) * (1 + x)$
⟨*proof*⟩

lemma *square-continuous*: $0 < (e::real) ==> \exists d.\ 0 < d \wedge (\forall y.\ abs(y - x) < d \longrightarrow abs(y * y - x * x) < e)$
⟨*proof*⟩

lemma *real-le-lsqrt*: $0 <= x \Longrightarrow 0 <= y \Longrightarrow x <= y\hat{}2 ==> sqrt\ x <= y$

⟨*proof*⟩

lemma *real-le-rsqrt*: $x\hat{}2 \leq y \Longrightarrow x \leq sqrt\ y$
⟨*proof*⟩

lemma *real-less-rsqrt*: $x\hat{}2 < y \Longrightarrow x < sqrt\ y$
⟨*proof*⟩

lemma *sqrt-even-pow2*: **assumes** *n*: *even n*
shows $sqrt(2\ \hat{}\ n) = 2\ \hat{}\ (n\ div\ 2)$
⟨*proof*⟩

lemma *real-div-sqrt*: $0 <= x ==> x\ /\ sqrt(x) = sqrt(x)$
⟨*proof*⟩

Hence derive more interesting properties of the norm.

lemma *norm-mul*[*simp*]: $norm(a *s\ x) = abs(a) * norm\ x$
⟨*proof*⟩

lemma *norm-eq-0-dot*: $(norm\ x = 0) \longleftrightarrow (inner\ x\ x = (0::real))$
⟨*proof*⟩
lemma *norm-eq-0-imp*: $norm\ x = 0 ==> x = (0::real\ \hat{}'n)$ ⟨*proof*⟩
lemma *vector-mul-eq-0*[*simp*]: $(a *s\ x = 0) \longleftrightarrow a = (0::'a::idom) \lor x = 0$
⟨*proof*⟩
lemma *vector-mul-lcancel*[*simp*]: $a *s\ x = a *s\ y \longleftrightarrow a = (0::real) \lor x = y$
⟨*proof*⟩
lemma *vector-mul-rcancel*[*simp*]: $a *s\ x = b *s\ x \longleftrightarrow (a::real) = b \lor x = 0$
⟨*proof*⟩
lemma *vector-mul-lcancel-imp*: $a \neq (0::real) ==>\ \ a *s\ x = a *s\ y ==> (x = y)$
⟨*proof*⟩
lemma *vector-mul-rcancel-imp*: $x \neq 0 \Longrightarrow (a::real) *s\ x = b *s\ x ==> a = b$
⟨*proof*⟩

lemma *norm-cauchy-schwarz*:
shows $inner\ x\ y <= norm\ x * norm\ y$
⟨*proof*⟩

lemma *norm-cauchy-schwarz-abs*:
shows $|inner\ x\ y| \leq norm\ x * norm\ y$
⟨*proof*⟩

lemma *norm-triangle-sub*:
fixes $x\ y :: 'a::real\text{-}normed\text{-}vector$
shows $norm\ x \leq norm\ y\ + norm\ (x - y)$
⟨*proof*⟩

lemma *component-le-norm*: $|x\$i| <= norm\ x$
⟨*proof*⟩

lemma *norm-bound-component-le*: $norm\ x <= e ==> |x\$i| <= e$
⟨*proof*⟩

lemma *norm-bound-component-lt*: $norm\ x < e ==> |x\$i| < e$
⟨*proof*⟩

lemma *norm-le-l1*: $norm\ x <= setsum(\lambda i.\ |x\$i|)\ UNIV$
⟨*proof*⟩

lemma *real-abs-norm*: $|norm\ x| = norm\ x$
⟨*proof*⟩
lemma *real-abs-sub-norm*: $|norm\ x - norm\ y| <= norm(x - y)$
⟨*proof*⟩
lemma *norm-le*: $norm(x) <= norm(y) \longleftrightarrow x \cdot x <= y \cdot y$
⟨*proof*⟩
lemma *norm-lt*: $norm(x) < norm(y) \longleftrightarrow x \cdot x < y \cdot y$
⟨*proof*⟩
lemma *norm-eq*: $norm(x) = norm\ (y) \longleftrightarrow x \cdot x = y \cdot y$
⟨*proof*⟩
lemma *norm-eq-1*: $norm(x) = 1 \longleftrightarrow x \cdot x = 1$
⟨*proof*⟩

Squaring equations and inequalities involving norms.

lemma *dot-square-norm*: $x \cdot x = norm(x)\hat{}2$
⟨*proof*⟩

lemma *norm-eq-square*: $norm(x) = a \longleftrightarrow 0 <= a \wedge x \cdot x = a\hat{}2$
⟨*proof*⟩

lemma *real-abs-le-square-iff*: $|x| \leq |y| \longleftrightarrow (x::real)\hat{}2 \leq y\hat{}2$
⟨*proof*⟩

lemma *norm-le-square*: $norm(x) <= a \longleftrightarrow 0 <= a \wedge x \cdot x <= a\hat{}2$
⟨*proof*⟩

lemma *norm-ge-square*: $norm(x) >= a \longleftrightarrow a <= 0 \vee x \cdot x >= a\ \hat{}\ 2$
⟨*proof*⟩

lemma *norm-lt-square*: $norm(x) < a \longleftrightarrow 0 < a \wedge x \cdot x < a\hat{}2$
⟨*proof*⟩
lemma *norm-gt-square*: $norm(x) > a \longleftrightarrow a < 0 \vee x \cdot x > a\hat{}2$
⟨*proof*⟩

Dot product in terms of the norm rather than conversely.

lemmas *inner-simps = inner.add-left inner.add-right inner.diff-right inner.diff-left*

inner.scaleR-left inner.scaleR-right

lemma *dot-norm*: $x \cdot y = (norm(x + y)\ \hat{}2 - norm\ x\ \hat{}\ 2 - norm\ y\ \hat{}\ 2)\ /\ 2$
⟨*proof*⟩

lemma *dot-norm-neg*: $x \cdot y = ((norm\ x\ \hat{}\ 2 + norm\ y\ \hat{}\ 2) - norm(x - y)\ \hat{}\ 2)$
$/\ 2$
⟨*proof*⟩

Equality of vectors in terms of *op* · products.

lemma *vector-eq*: $x = y \longleftrightarrow x \cdot x = x \cdot y \wedge y \cdot y = x \cdot x$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

13.5 General linear decision procedure for normed spaces.

lemma *norm-cmul-rule-thm*:
fixes x :: *'a::real-normed-vector*
shows $b >= norm(x) ==> |c| * b >= norm(scaleR\ c\ x)$
⟨*proof*⟩

lemma *norm-add-rule-thm*:
fixes *x1 x2* :: *'a::real-normed-vector*
shows $norm\ x1 \le b1 \Longrightarrow norm\ x2 \le b2 \Longrightarrow norm\ (x1 + x2) \le b1 + b2$
⟨*proof*⟩

lemma *ge-iff-diff-ge-0*: $(a::'a::linordered\text{-}ring) \ge b == a - b \ge 0$
⟨*proof*⟩

lemma *pth-1*:
fixes x :: *'a::real-normed-vector*
shows $x == scaleR\ 1\ x$ ⟨*proof*⟩

lemma *pth-2*:
fixes x :: *'a::real-normed-vector*
shows $x - y == x + -y$ ⟨*proof*⟩

lemma *pth-3*:
fixes x :: *'a::real-normed-vector*
shows $- x == scaleR\ (-1)\ x$ ⟨*proof*⟩

lemma *pth-4*:
fixes x :: *'a::real-normed-vector*
shows $scaleR\ 0\ x == 0$ **and** $scaleR\ c\ 0 = (0::'a)$ ⟨*proof*⟩

lemma *pth-5*:
fixes x :: *'a::real-normed-vector*
shows $scaleR\ c\ (scaleR\ d\ x) == scaleR\ (c * d)\ x$ ⟨*proof*⟩

lemma *pth-6*:
fixes x :: *'a::real-normed-vector*

shows *scaleR c (x + y) == scaleR c x + scaleR c y*
⟨*proof*⟩

lemma *pth-7*:
fixes *x* :: *'a::real-normed-vector*
shows *0 + x == x* **and** *x + 0 == x* ⟨*proof*⟩

lemma *pth-8*:
fixes *x* :: *'a::real-normed-vector*
shows *scaleR c x + scaleR d x == scaleR (c + d) x*
⟨*proof*⟩

lemma *pth-9*:
fixes *x* :: *'a::real-normed-vector* **shows**
(scaleR c x + z) + scaleR d x == scaleR (c + d) x + z
scaleR c x + (scaleR d x + z) == scaleR (c + d) x + z
(scaleR c x + w) + (scaleR d x + z) == scaleR (c + d) x + (w + z)
⟨*proof*⟩

lemma *pth-a*:
fixes *x* :: *'a::real-normed-vector*
shows *scaleR 0 x + y == y* ⟨*proof*⟩

lemma *pth-b*:
fixes *x* :: *'a::real-normed-vector* **shows**
scaleR c x + scaleR d y == scaleR c x + scaleR d y
(scaleR c x + z) + scaleR d y == scaleR c x + (z + scaleR d y)
scaleR c x + (scaleR d y + z) == scaleR c x + (scaleR d y + z)
(scaleR c x + w) + (scaleR d y + z) == scaleR c x + (w + (scaleR d y + z))
⟨*proof*⟩

lemma *pth-c*:
fixes *x* :: *'a::real-normed-vector* **shows**
scaleR c x + scaleR d y == scaleR d y + scaleR c x
(scaleR c x + z) + scaleR d y == scaleR d y + (scaleR c x + z)
scaleR c x + (scaleR d y + z) == scaleR d y + (scaleR c x + z)
(scaleR c x + w) + (scaleR d y + z) == scaleR d y + ((scaleR c x + w) + z)
⟨*proof*⟩

lemma *pth-d*:
fixes *x* :: *'a::real-normed-vector*
shows *x + 0 == x* ⟨*proof*⟩

lemma *norm-imp-pos-and-ge*:
fixes *x* :: *'a::real-normed-vector*
shows *norm x == n* $\Longrightarrow$ *norm x* $\geq$ *0* $\wedge$ *n* $\geq$ *norm x*
⟨*proof*⟩

lemma *real-eq-0-iff-le-ge-0*: *(x::real) = 0 == x* $\geq$ *0* $\wedge$ $-x \geq$ *0* ⟨*proof*⟩

lemma *norm-pths*:
fixes x :: $'a$::*real-normed-vector* **shows**
$x = y \longleftrightarrow norm\ (x - y) \leq 0$
$x \neq y \longleftrightarrow \neg\ (norm\ (x - y) \leq 0)$
⟨*proof*⟩

⟨*ML*⟩

Hence more metric properties.

lemma *dist-triangle-alt*:
fixes $x\ y\ z$ :: $'a$::*metric-space*
shows $dist\ y\ z <= dist\ x\ y + dist\ x\ z$
⟨*proof*⟩

lemma *dist-pos-lt*:
fixes $x\ y$ :: $'a$::*metric-space*
shows $x \neq y ==> 0 < dist\ x\ y$
⟨*proof*⟩

lemma *dist-nz*:
fixes $x\ y$ :: $'a$::*metric-space*
shows $x \neq y \longleftrightarrow 0 < dist\ x\ y$
⟨*proof*⟩

lemma *dist-triangle-le*:
fixes $x\ y\ z$ :: $'a$::*metric-space*
shows $dist\ x\ z + dist\ y\ z <= e \Longrightarrow dist\ x\ y <= e$
⟨*proof*⟩

lemma *dist-triangle-lt*:
fixes $x\ y\ z$ :: $'a$::*metric-space*
shows $dist\ x\ z + dist\ y\ z < e ==> dist\ x\ y < e$
⟨*proof*⟩

lemma *dist-triangle-half-l*:
fixes $x1\ x2\ y$ :: $'a$::*metric-space*
shows $dist\ x1\ y < e\ /\ 2 \Longrightarrow dist\ x2\ y < e\ /\ 2 \Longrightarrow dist\ x1\ x2 < e$
⟨*proof*⟩

lemma *dist-triangle-half-r*:
fixes $x1\ x2\ y$ :: $'a$::*metric-space*
shows $dist\ y\ x1 < e\ /\ 2 \Longrightarrow dist\ y\ x2 < e\ /\ 2 \Longrightarrow dist\ x1\ x2 < e$
⟨*proof*⟩

lemma *norm-triangle-half-r*:
shows $norm\ (y - x1) < e\ /\ 2 \Longrightarrow norm\ (y - x2) < e\ /\ 2 \Longrightarrow norm\ (x1 - x2) < e$

⟨*proof*⟩

lemma *norm-triangle-half-l*: **assumes** $norm\ (x - y) < e\ /\ 2\ norm\ (x' - (y)) < e\ /\ 2$
shows $norm\ (x - x') < e$
⟨*proof*⟩

lemma *norm-triangle-le*: $norm(x) + norm\ y <= e ==> norm(x + y) <= e$
⟨*proof*⟩

lemma *norm-triangle-lt*: $norm(x) + norm(y) < e ==> norm(x + y) < e$
⟨*proof*⟩

lemma *dist-triangle-add*:
fixes $x\ y\ x'\ y' :: 'a::real\text{-}normed\text{-}vector$
shows $dist\ (x + y)\ (x' + y') <= dist\ x\ x' + dist\ y\ y'$
⟨*proof*⟩

lemma *dist-mul*[*simp*]: $dist\ (c *s\ x)\ (c *s\ y) = |c| * dist\ x\ y$
⟨*proof*⟩

lemma *dist-triangle-add-half*:
fixes $x\ x'\ y\ y' :: 'a::real\text{-}normed\text{-}vector$
shows $dist\ x\ x' < e\ /\ 2 \Longrightarrow dist\ y\ y' < e\ /\ 2 \Longrightarrow dist(x + y)\ (x' + y') < e$
⟨*proof*⟩

lemma *setsum-component* [*simp*]:
fixes $f :: 'a \Rightarrow ('b::comm\text{-}monoid\text{-}add)\ \hat{}\ 'n$
shows $(setsum\ f\ S)\$i = setsum\ (\lambda x.\ (f\ x)\$i)\ S$
⟨*proof*⟩

lemma *setsum-eq*: $setsum\ f\ S = (\chi\ i.\ setsum\ (\lambda x.\ (f\ x)\$i\ )\ S)$
⟨*proof*⟩

lemma *setsum-clauses*:
shows $setsum\ f\ \{\} = 0$
and $finite\ S \Longrightarrow setsum\ f\ (insert\ x\ S) =$
$(if\ x \in S\ then\ setsum\ f\ S\ else\ f\ x + setsum\ f\ S)$
⟨*proof*⟩

lemma *setsum-cmul*:
fixes $f :: 'c \Rightarrow ('a::semiring\text{-}1)\ \hat{}\ 'n$
shows $setsum\ (\lambda x.\ c *s\ f\ x)\ S = c *s\ setsum\ f\ S$
⟨*proof*⟩

lemma *setsum-norm*:
fixes $f :: 'a \Rightarrow 'b::real\text{-}normed\text{-}vector$
assumes *fS*: *finite S*
shows $norm\ (setsum\ f\ S) <= setsum\ (\lambda x.\ norm(f\ x))\ S$

⟨*proof*⟩

lemma *setsum-norm-le*:
  fixes f :: $'a \Rightarrow 'b$::*real-normed-vector*
  assumes *fS*: *finite S*
  and *fg*: $\forall x \in S.$ *norm* $(f\ x) \leq g\ x$
  shows *norm* (*setsum f S*) $\leq$ *setsum g S*
⟨*proof*⟩

lemma *setsum-norm-bound*:
  fixes f :: $'a \Rightarrow 'b$::*real-normed-vector*
  assumes *fS*: *finite S*
  and K: $\forall x \in S.$ *norm* $(f\ x) \leq K$
  shows *norm* (*setsum f S*) $\leq$ *of-nat* (*card S*) $*$ K
  ⟨*proof*⟩

lemma *setsum-vmul*:
  fixes f :: $'a \Rightarrow 'b$::*semiring-0*
  assumes *fS*: *finite S*
  shows *setsum f S* $*s$ v = *setsum* $(\lambda x.\ f\ x *s\ v)$ S
⟨*proof*⟩

lemma *setsum-group*:
  assumes *fS*: *finite S* **and** *fT*: *finite T* **and** *fST*: f ‘ $S \subseteq T$
  shows *setsum* $(\lambda y.$ *setsum* g $\{x.\ x \in S \wedge f\ x = y\})$ T = *setsum g S*

⟨*proof*⟩

lemma *vsum-norm-allsubsets-bound*:
  fixes f:: $'a \Rightarrow$ *real* $\hat{}\,'n$
  assumes *fP*: *finite P* **and** *fPs*: $\bigwedge Q.\ Q \subseteq P \Longrightarrow$ *norm* (*setsum f Q*) $\leq e$
  shows *setsum* $(\lambda x.$ *norm* $(f\ x))$ $P \leq 2 *$ *real CARD*$('n) *\ e$
⟨*proof*⟩

lemma *dot-lsum*: *finite S* $\Longrightarrow$ *setsum f S* $\cdot\ y$ = *setsum* $(\lambda x.\ f\ x \cdot y)$ S
  ⟨*proof*⟩

lemma *dot-rsum*: *finite S* $\Longrightarrow$ $y\ \cdot$ *setsum f S* = *setsum* $(\lambda x.\ y \cdot f\ x)$ S
  ⟨*proof*⟩

13.6 Basis vectors in coordinate directions.

definition *basis* $k = (\chi\ i.$ *if* $i = k$ *then 1 else 0*$)$

lemma *basis-component* [*simp*]: *basis* k \$ i = (*if* k=i *then 1 else 0*)
  ⟨*proof*⟩

lemma *delta-mult-idempotent*:
$(if\ k{=}a\ then\ 1\ else\ (0{::}'a{::}semiring\text{-}1)) * (if\ k{=}a\ then\ 1\ else\ 0) = (if\ k{=}a\ then\ 1\ else\ 0)$ ⟨*proof*⟩

lemma *norm-basis*:
shows $norm\ (basis\ k :: real\ \hat{}\,'n) = 1$
⟨*proof*⟩

lemma *norm-basis-1*: $norm(basis\ 1 :: real\ \hat{}\,'n{::}\{finite,one\}) = 1$
⟨*proof*⟩

lemma *vector-choose-size*: $0 <= c ==> \exists (x{::}real\hat{}\,'n).\ norm\ x = c$
⟨*proof*⟩

lemma *vector-choose-dist*: **assumes** e: $0 <= e$
shows $\exists (y{::}real\hat{}\,'n).\ dist\ x\ y = e$
⟨*proof*⟩

lemma *basis-inj*: $inj\ (basis :: 'n \Rightarrow real\ \hat{}\,'n)$
⟨*proof*⟩

lemma *cond-value-iff*: $f\ (if\ b\ then\ x\ else\ y) = (if\ b\ then\ f\ x\ else\ f\ y)$
⟨*proof*⟩

lemma *basis-expansion*:
$setsum\ (\lambda i.\ (x\$i) *s\ basis\ i)\ UNIV = (x{::}('a{::}ring\text{-}1)\ \hat{}\,'n)$ (**is** $?lhs = ?rhs$ **is** $setsum\ ?f\ ?S = \text{-}$)
⟨*proof*⟩

lemma *smult-conv-scaleR*: $c *s\ x = scaleR\ c\ x$
⟨*proof*⟩

lemma *basis-expansion'*:
$setsum\ (\lambda i.\ (x\$i) *_R\ basis\ i)\ UNIV = x$
⟨*proof*⟩

lemma *basis-expansion-unique*:
$setsum\ (\lambda i.\ f\ i *s\ basis\ i)\ UNIV = (x{::}('a{::}comm\text{-}ring\text{-}1)\ \hat{}\,'n) \longleftrightarrow (\forall i.\ f\ i = x\$i)$
⟨*proof*⟩

lemma *cond-application-beta*: $(if\ b\ then\ f\ else\ g)\ x = (if\ b\ then\ f\ x\ else\ g\ x)$
⟨*proof*⟩

lemma *dot-basis*:
shows $basis\ i \cdot x = x\$i\ \ x \cdot (basis\ i) = (x\$i)$
⟨*proof*⟩

lemma *inner-basis*:
fixes x :: $'a$::{*real-inner*, *real-algebra-1*} ^ $'n$
shows *inner* (*basis* i) x = *inner* *1* (x \$ i)
and *inner* x (*basis* i) = *inner* (x \$ i) *1*
⟨*proof*⟩

lemma *basis-eq-0*: *basis* i = (*0*::$'a$::*semiring-1*^$'n$) ⟷ *False*
⟨*proof*⟩

lemma *basis-nonzero*:
shows *basis* $k \neq$ (*0*:: $'a$::*semiring-1* ^$'n$)
⟨*proof*⟩

lemma *vector-eq-ldot*: ($\forall x.\ x \cdot y = x \cdot z$) ⟷ $y = z$
⟨*proof*⟩

lemma *vector-eq-rdot*: ($\forall z.\ x \cdot z = y \cdot z$) ⟷ $x = y$
⟨*proof*⟩

13.7 Orthogonality.

definition *orthogonal* x y ⟷ ($x \cdot y$ = *0*)

lemma *orthogonal-basis*:
shows *orthogonal* (*basis* i) x ⟷ $x\$i$ = (*0*::*real*)
⟨*proof*⟩

lemma *orthogonal-basis-basis*:
shows *orthogonal* (*basis* i :: *real*^$'n$) (*basis* j) ⟷ $i \neq j$
⟨*proof*⟩

lemma *orthogonal-clauses*:
orthogonal a *0*
orthogonal a x ==> *orthogonal* a ($c *_R x$)
orthogonal a x ==> *orthogonal* a ($-x$)
orthogonal a x ⟹ *orthogonal* a y ==> *orthogonal* a ($x + y$)
orthogonal a x ⟹ *orthogonal* a y ==> *orthogonal* a ($x - y$)
orthogonal *0* a
orthogonal x a ==> *orthogonal* ($c *_R x$) a
orthogonal x a ==> *orthogonal* ($-x$) a
orthogonal x a ⟹ *orthogonal* y a ==> *orthogonal* ($x + y$) a
orthogonal x a ⟹ *orthogonal* y a ==> *orthogonal* ($x - y$) a
⟨*proof*⟩

lemma *orthogonal-commute*: *orthogonal* x y ⟷ *orthogonal* y x
⟨*proof*⟩

13.8 Linear functions.

definition

linear :: $('a::real\text{-}vector \Rightarrow 'b::real\text{-}vector) \Rightarrow bool$ **where**
linear f $\longleftrightarrow (\forall x\ y.\ f(x + y) = f\,x + f\,y) \wedge (\forall c\ x.\ f(c *_R x) = c *_R f\,x)$

lemma *linearI*: **assumes** $\bigwedge x\ y.\ f\ (x + y) = f\,x + f\,y$ $\bigwedge c\ x.\ f\ (c *_R x) = c *_R f$ x
shows *linear f* ⟨*proof*⟩

lemma *linear-compose-cmul*: *linear f* $==>$ *linear* $(\lambda x.\ c *_R f\,x)$
⟨*proof*⟩

lemma *linear-compose-neg*: *linear f* $==>$ *linear* $(\lambda x.\ -(f(x)))$
⟨*proof*⟩

lemma *linear-compose-add*: *linear f* $\Longrightarrow$ *linear g* $==>$ *linear* $(\lambda x.\ f(x) + g(x))$
⟨*proof*⟩

lemma *linear-compose-sub*: *linear f* $\Longrightarrow$ *linear g* $==>$ *linear* $(\lambda x.\ f\,x - g\,x)$
⟨*proof*⟩

lemma *linear-compose*: *linear f* $\Longrightarrow$ *linear g* $==>$ *linear* $(g\ o\ f)$
⟨*proof*⟩

lemma *linear-id*: *linear id* ⟨*proof*⟩

lemma *linear-zero*: *linear* $(\lambda x.\ 0)$ ⟨*proof*⟩

lemma *linear-compose-setsum*:
assumes *fS*: *finite S* **and** *lS*: $\forall a \in S.$ *linear* $(f\ a)$
shows *linear*$(\lambda x.\ setsum\ (\lambda a.\ f\ a\ x)\ S)$
⟨*proof*⟩

lemma *linear-vmul-component*:
assumes *lf*: *linear f*
shows *linear* $(\lambda x.\ f\,x\ \$\ k *_R v)$
⟨*proof*⟩

lemma *linear-0*: *linear f* $==> f\ 0 = 0$
⟨*proof*⟩

lemma *linear-cmul*: *linear f* $==> f(c *_R x) = c *_R f\,x$ ⟨*proof*⟩

lemma *linear-neg*: *linear f* $==> f\ (-x) = -\ f\,x$
⟨*proof*⟩

lemma *linear-add*: *linear f* $==> f(x + y) = f\,x + f\,y$ ⟨*proof*⟩

lemma *linear-sub*: *linear f* $==> f(x - y) = f\,x - f\,y$
⟨*proof*⟩

lemma *linear-setsum*:
assumes *lf*: *linear f* **and** *fS*: *finite S*
shows $f\ (setsum\ g\ S) = setsum\ (f\ o\ g)\ S$
⟨*proof*⟩

lemma *linear-setsum-mul*:
assumes *lf*: *linear f* **and** *fS*: *finite S*
shows $f\ (setsum\ (\lambda i.\ c\ i *_R v\ i)\ S) = setsum\ (\lambda i.\ c\ i *_R f\ (v\ i))\ S$
⟨*proof*⟩

lemma *linear-injective-0*:
assumes *lf*: *linear f*
shows $inj\ f \longleftrightarrow (\forall x.\ f\ x = 0 \longrightarrow x = 0)$
⟨*proof*⟩

lemma *linear-bounded*:
fixes f:: $real\ \hat{}'m \Rightarrow real\ \hat{}'n$
assumes *lf*: *linear f*
shows $\exists B.\ \forall x.\ norm\ (f\ x) \le B * norm\ x$
⟨*proof*⟩

lemma *linear-bounded-pos*:
fixes f:: $real\ \hat{}'n \Rightarrow real\ \hat{}'m$
assumes *lf*: *linear f*
shows $\exists B > 0.\ \forall x.\ norm\ (f\ x) \le B * norm\ x$
⟨*proof*⟩

lemma *linear-conv-bounded-linear*:
fixes f :: $real\ \hat{}\ \text{-} \Rightarrow real\ \hat{}\ \text{-}$
shows $linear\ f \longleftrightarrow bounded\text{-}linear\ f$
⟨*proof*⟩

lemma *bounded-linearI'*: **fixes** f::$real\hat{}'n \Rightarrow real\hat{}'m$
assumes $\bigwedge x\ y.\ f\ (x + y) = f\ x + f\ y$ $\bigwedge c\ x.\ f\ (c *_R x) = c *_R f\ x$
shows *bounded-linear f* ⟨*proof*⟩

13.9 Bilinear functions.

definition $bilinear\ f \longleftrightarrow (\forall x.\ linear(\lambda y.\ f\ x\ y)) \wedge (\forall y.\ linear(\lambda x.\ f\ x\ y))$

lemma *bilinear-ladd*: $bilinear\ h \Longrightarrow h\ (x + y)\ z = (h\ x\ z) + (h\ y\ z)$
⟨*proof*⟩
lemma *bilinear-radd*: $bilinear\ h \Longrightarrow h\ x\ (y + z) = (h\ x\ y) + (h\ x\ z)$
⟨*proof*⟩

lemma *bilinear-lmul*: $bilinear\ h \Longrightarrow h\ (c *_R x)\ y = c *_R (h\ x\ y)$
⟨*proof*⟩

lemma *bilinear-rmul*: $bilinear\ h \Longrightarrow h\ x\ (c *_R y) = c *_R (h\ x\ y)$

⟨*proof*⟩

lemma *bilinear-lneg*: *bilinear* h ==> $h\ (-\ x)\ y = -(h\ x\ y)$
⟨*proof*⟩

lemma *bilinear-rneg*: *bilinear* h ==> $h\ x\ (-\ y) = -\ h\ x\ y$
⟨*proof*⟩

lemma (**in** *ab-group-add*) *eq-add-iff*: $x = x + y \longleftrightarrow y = 0$
⟨*proof*⟩

lemma *bilinear-lzero*:
assumes *bh*: *bilinear* h **shows** $h\ 0\ x = 0$
⟨*proof*⟩

lemma *bilinear-rzero*:
assumes *bh*: *bilinear* h **shows** $h\ x\ 0 = 0$
⟨*proof*⟩

lemma *bilinear-lsub*: *bilinear* h ==> $h\ (x - y)\ z = h\ x\ z - h\ y\ z$
⟨*proof*⟩

lemma *bilinear-rsub*: *bilinear* h ==> $h\ z\ (x - y) = h\ z\ x - h\ z\ y$
⟨*proof*⟩

lemma *bilinear-setsum*:
assumes *bh*: *bilinear* h **and** *fS*: *finite* S **and** *fT*: *finite* T
shows h (*setsum* f S) (*setsum* g T) = *setsum* ($\lambda(i,j)$. $h\ (f\ i)\ (g\ j)$) ($S \times T$)
⟨*proof*⟩

lemma *bilinear-bounded*:
fixes h:: *real* ^'*m* ⇒ *real*^'*n* ⇒ *real* ^'*k*
assumes *bh*: *bilinear* h
shows $\exists B.\ \forall x\ y.\ norm\ (h\ x\ y) \leq B * norm\ x * norm\ y$
⟨*proof*⟩

lemma *bilinear-bounded-pos*:
fixes h:: *real* ^'*m* ⇒ *real*^'*n* ⇒ *real* ^'*k*
assumes *bh*: *bilinear* h
shows $\exists B > 0.\ \forall x\ y.\ norm\ (h\ x\ y) \leq B * norm\ x * norm\ y$
⟨*proof*⟩

lemma *bilinear-conv-bounded-bilinear*:
fixes h :: *real* ^ - ⇒ *real* ^ - ⇒ *real* ^ -
shows *bilinear* h ⟷ *bounded-bilinear* h
⟨*proof*⟩

13.10 Adjoints.

definition $adjoint\ f = (SOME\ f'.\ \forall x\ y.\ f\ x \cdot y = x \cdot f'\ y)$

lemma *adjoint-unique*:
assumes $\forall x\ y.\ inner\ (f\ x)\ y = inner\ x\ (g\ y)$
shows $adjoint\ f = g$
⟨*proof*⟩

lemma *choice-iff*: $(\forall x.\ \exists y.\ P\ x\ y) \longleftrightarrow (\exists f.\ \forall x.\ P\ x\ (f\ x))$ ⟨*proof*⟩

TODO: The following lemmas about adjoints should hold for any Hilbert space (i.e. complete inner product space). (see http://en.wikipedia.org/wiki/Hermitian_adjoint)

lemma *adjoint-works-lemma*:
fixes $f{::}\ real\ \hat{}'n \Rightarrow real\ \hat{}'m$
assumes *lf*: *linear f*
shows $\forall x\ y.\ f\ x \cdot y = x \cdot adjoint\ f\ y$
⟨*proof*⟩

lemma *adjoint-works*:
fixes $f{::}\ real\ \hat{}'n \Rightarrow real\ \hat{}'m$
assumes *lf*: *linear f*
shows $x \cdot adjoint\ f\ y = f\ x \cdot y$
⟨*proof*⟩

lemma *adjoint-linear*:
fixes $f{::}\ real\ \hat{}'n \Rightarrow real\ \hat{}'m$
assumes *lf*: *linear f*
shows *linear* (*adjoint f*)
⟨*proof*⟩

lemma *adjoint-clauses*:
fixes $f{::}\ real\ \hat{}'n \Rightarrow real\ \hat{}'m$
assumes *lf*: *linear f*
shows $x \cdot adjoint\ f\ y = f\ x \cdot y$
and $adjoint\ f\ y \cdot x = y \cdot f\ x$
⟨*proof*⟩

lemma *adjoint-adjoint*:
fixes $f{::}\ real\ \hat{}'n \Rightarrow real\ \hat{}'m$
assumes *lf*: *linear f*
shows $adjoint\ (adjoint\ f) = f$
⟨*proof*⟩

13.11 Matrix operations

Matrix notation. NB: an MxN matrix is of type $(('a,\ 'n)\ cart,\ 'm)\ cart$, not $(('a,\ 'm)\ cart,\ 'n)\ cart$

definition *matrix-matrix-mult* :: $('a::semiring\text{-}1)\ \hat{}\ 'n\hat{}'m \Rightarrow 'a\ \hat{}'p\hat{}'n \Rightarrow 'a\ \hat{}\ 'p\ \hat{}'m$ (**infixl** $**$ *70*)
where $m ** m' == (\chi\ i\ j.\ setsum\ (\lambda k.\ ((m\$i)\$k) * ((m'\$k)\$j))\ (UNIV :: 'n\ set)) :: 'a\ \hat{}\ 'p\ \hat{}'m$

definition *matrix-vector-mult* :: $('a::semiring\text{-}1)\ \hat{}'n\hat{}'m \Rightarrow 'a\ \hat{}'n \Rightarrow 'a\ \hat{}\ 'm$ (**infixl** $*v$ *70*)
where $m *v\ x \equiv (\chi\ i.\ setsum\ (\lambda j.\ ((m\$i)\$j) * (x\$j))\ (UNIV :: 'n\ set)) :: 'a\hat{}'m$

definition *vector-matrix-mult* :: $'a\ \hat{}\ 'm \Rightarrow ('a::semiring\text{-}1)\ \hat{}'n\hat{}'m \Rightarrow 'a\ \hat{}'n$ (**infixl** $v*$ *70*)
where $v\ v*\ m == (\chi\ j.\ setsum\ (\lambda i.\ ((m\$i)\$j) * (v\$i))\ (UNIV :: 'm\ set)) :: 'a\hat{}'n$

definition $(mat::'a::zero => 'a\ \hat{}'n\hat{}'n)\ k = (\chi\ i\ j.\ if\ i = j\ then\ k\ else\ 0)$
definition *transpose* **where**
$(transpose::'a\hat{}'n\hat{}'m \Rightarrow 'a\hat{}'m\hat{}'n)\ A = (\chi\ i\ j.\ ((A\$j)\$i))$
definition $(row::'m => 'a\ \hat{}'n\hat{}'m \Rightarrow 'a\ \hat{}'n)\ i\ A = (\chi\ j.\ ((A\$i)\$j))$
definition $(column::'n => 'a\hat{}'n\hat{}'m => 'a\hat{}'m)\ j\ A = (\chi\ i.\ ((A\$i)\$j))$
definition $rows(A::'a\hat{}'n\hat{}'m) = \{\ row\ i\ A \mid i.\ i \in (UNIV :: 'm\ set)\}$
definition $columns(A::'a\hat{}'n\hat{}'m) = \{\ column\ i\ A \mid i.\ i \in (UNIV :: 'n\ set)\}$

lemma *mat-0*[*simp*]: *mat 0 = 0* ⟨*proof*⟩
lemma *matrix-add-ldistrib*: $(A ** (B + C)) = (A ** B) + (A ** C)$
⟨*proof*⟩

lemma *matrix-mul-lid*:
fixes $A :: 'a::semiring\text{-}1\ \hat{}\ 'm\ \hat{}\ 'n$
shows $mat\ 1 ** A = A$
⟨*proof*⟩

lemma *matrix-mul-rid*:
fixes $A :: 'a::semiring\text{-}1\ \hat{}\ 'm\ \hat{}\ 'n$
shows $A ** mat\ 1 = A$
⟨*proof*⟩

lemma *matrix-mul-assoc*: $A ** (B ** C) = (A ** B) ** C$
⟨*proof*⟩

lemma *matrix-vector-mul-assoc*: $A *v (B *v x) = (A ** B) *v x$
⟨*proof*⟩

lemma *matrix-vector-mul-lid*: $mat\ 1 *v\ x = (x::'a::semiring\text{-}1\ \hat{}\ 'n)$
⟨*proof*⟩

lemma *matrix-transpose-mul*: $transpose(A ** B) = transpose\ B ** transpose\ (A::'a::comm\text{-}semiring\text{-}1\hat{}\text{-}\hat{}\text{-})$
⟨*proof*⟩

lemma *matrix-eq*:
  fixes $A\ B :: {'a}{::}semiring\text{-}1\ \hat{}\ {'n}\ \hat{}\ {'m}$
  shows $A = B \longleftrightarrow (\forall x.\ A *v\ x = B *v\ x)$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
  ⟨*proof*⟩

lemma *matrix-vector-mul-component*:
  shows $((A{::}real\hat{}\text{-}\hat{}\text{-}) *v\ x)\$k = (A\$k) \cdot x$
  ⟨*proof*⟩

lemma *dot-lmul-matrix*: $((x{::}real\ \hat{}\text{-})\ v{*}\ A) \cdot y = x \cdot (A *v\ y)$
  ⟨*proof*⟩

lemma *transpose-mat*: *transpose* (*mat n*) = *mat n*
  ⟨*proof*⟩

lemma *transpose-transpose*: *transpose*(*transpose A*) = *A*
  ⟨*proof*⟩

lemma *row-transpose*:
  fixes $A{::}\ {'a}{::}semiring\text{-}1\hat{}\text{-}\hat{}\text{-}$
  shows *row i* (*transpose A*) = *column i A*
  ⟨*proof*⟩

lemma *column-transpose*:
  fixes $A{::}\ {'a}{::}semiring\text{-}1\hat{}\text{-}\hat{}\text{-}$
  shows *column i* (*transpose A*) = *row i A*
  ⟨*proof*⟩

lemma *rows-transpose*: $rows(transpose\ (A{::}{'a}{::}semiring\text{-}1\hat{}\text{-}\hat{}\text{-})) = columns\ A$
⟨*proof*⟩

lemma *columns-transpose*: $columns(transpose\ (A{::}{'a}{::}semiring\text{-}1\hat{}\text{-}\hat{}\text{-})) = rows\ A$
⟨*proof*⟩

Two sometimes fruitful ways of looking at matrix-vector multiplication.

lemma *matrix-mult-dot*: $A *v\ x = (\chi\ i.\ A\$i \cdot x)$
  ⟨*proof*⟩

lemma *matrix-mult-vsum*: $(A{::}{'a}{::}comm\text{-}semiring\text{-}1\hat{}{'n}\hat{}{'m}) *v\ x = setsum\ (\lambda i.\ (x\$i) *s\ column\ i\ A)\ (UNIV{::}\ {'n}\ set)$
  ⟨*proof*⟩

lemma *vector-componentwise*:
  $(x{::}{'a}{::}ring\text{-}1\hat{}{'n}) = (\chi\ j.\ setsum\ (\lambda i.\ (x\$i) * (basis\ i :: {'a}\hat{}{'n})\$j)\ (UNIV :: {'n}\ set))$
  ⟨*proof*⟩

lemma *linear-componentwise*:
  fixes $f{::}\ real\ \hat{}{'m} \Rightarrow real\ \hat{}\ \text{-}$

assumes *lf*: *linear f*
shows $(f\ x)\$j = setsum\ (\lambda i.\ (x\$i) * (f\ (basis\ i)\$j))\ (UNIV :: 'm\ set)$ (**is** *?lhs = ?rhs*)
⟨*proof*⟩

Inverse matrices (not necessarily square)

definition $invertible(A::'a::semiring\text{-}1\hat{}'n\hat{}'m) \longleftrightarrow (\exists A'::'a\hat{}'m\hat{}'n.\ A ** A' = mat\ 1 \wedge A' ** A = mat\ 1)$

definition $matrix\text{-}inv(A::\ 'a::semiring\text{-}1\hat{}'n\hat{}'m) =$
$(SOME\ A'::'a\hat{}'m\hat{}'n.\ A ** A' = mat\ 1 \wedge A' ** A = mat\ 1)$

Correspondence between matrices and linear operators.

definition $matrix:: ('a::\{plus,times,\ one,\ zero\}\hat{}'m \Rightarrow 'a\ \hat{}\ 'n) \Rightarrow 'a\hat{}'m\hat{}'n$
where $matrix\ f = (\chi\ i\ j.\ (f(basis\ j))\$i)$

lemma *matrix-vector-mul-linear*: $linear(\lambda x.\ A *v\ (x::real\ \hat{}\ \text{-}))$
⟨*proof*⟩

lemma *matrix-works*: **assumes** *lf*: *linear f* **shows** $matrix\ f *v\ x = f\ (x::real\ \hat{}\ 'n)$
⟨*proof*⟩

lemma *matrix-vector-mul*: $linear\ f ==> f = (\lambda x.\ matrix\ f *v\ (x::real\ \hat{}\ 'n))$
⟨*proof*⟩

lemma *matrix-of-matrix-vector-mul*: $matrix(\lambda x.\ A *v\ (x :: real\ \hat{}\ 'n)) = A$
⟨*proof*⟩

lemma *matrix-compose*:
assumes *lf*: $linear\ (f::real\hat{}'n \Rightarrow real\hat{}'m)$
and *lg*: $linear\ (g::real\hat{}'m \Rightarrow real\hat{}\text{-})$
shows $matrix\ (g\ o\ f) = matrix\ g ** matrix\ f$
⟨*proof*⟩

lemma *matrix-vector-column*:$(A::'a::comm\text{-}semiring\text{-}1\hat{}'n\hat{}\text{-}) *v\ x = setsum\ (\lambda i.\ (x\$i) *s\ ((transpose\ A)\$i))\ (UNIV:: 'n\ set)$
⟨*proof*⟩

lemma *adjoint-matrix*: $adjoint(\lambda x.\ (A::real\hat{}'n\hat{}'m) *v\ x) = (\lambda x.\ transpose\ A *v\ x)$
⟨*proof*⟩

lemma *matrix-adjoint*: **assumes** *lf*: $linear\ (f :: real\hat{}'n \Rightarrow real\ \hat{}'m)$
shows $matrix(adjoint\ f) = transpose(matrix\ f)$
⟨*proof*⟩

13.12 Interlude: Some properties of real sets

lemma *seq-mono-lemma*: **assumes** $\forall (n{::}nat) \geq m.\ (d\ n :: real) < e\ n$ **and** $\forall n \geq m.\ e\ n <= e\ m$
shows $\forall n \geq m.\ d\ n < e\ m$
⟨*proof*⟩

lemma *infinite-enumerate*: **assumes** *fS*: *infinite S*
shows $\exists r.\ subseq\ r \wedge (\forall n.\ r\ n \in S)$
⟨*proof*⟩

lemma *approachable-lt-le*: $(\exists (d{::}real)>0.\ \forall x.\ f\ x < d \longrightarrow P\ x) \longleftrightarrow (\exists d>0.\ \forall x.\ f\ x \leq d \longrightarrow P\ x)$
⟨*proof*⟩

lemma *triangle-lemma*:
assumes x: $0 <= (x{::}real)$ **and** y:$0 <= y$ **and** z: $0 <= z$ **and** xy: $x\hat{}2 <= y\hat{}2 + z\hat{}2$
shows $x <= y + z$
⟨*proof*⟩

lemma *lambda-skolem*: $(\forall i.\ \exists x.\ P\ i\ x) \longleftrightarrow$
$(\exists x{::}'a\ \hat{}\ 'n.\ \forall i.\ P\ i\ (x\$i))$ (**is** $?lhs \longleftrightarrow ?rhs$)
⟨*proof*⟩

lemma *vec-in-image-vec*: $vec\ x \in (vec\ `\ S) \longleftrightarrow x \in S$ ⟨*proof*⟩

lemma *vec-add*: $vec(x + y) = vec\ x + vec\ y$ ⟨*proof*⟩
lemma *vec-sub*: $vec(x - y) = vec\ x - vec\ y$ ⟨*proof*⟩
lemma *vec-cmul*: $vec(c*\ x) = c\ {*}s\ vec\ x$ ⟨*proof*⟩
lemma *vec-neg*: $vec(-\ x) = -\ vec\ x$ ⟨*proof*⟩

lemma *vec-setsum*: **assumes** *fS*: *finite S*
shows $vec(setsum\ f\ S) = setsum\ (vec\ o\ f)\ S$
⟨*proof*⟩

lemma *setsum-Plus*:
$[\![finite\ A;\ finite\ B]\!] \Longrightarrow$
$(\sum x{\in}A <+> B.\ g\ x) = (\sum x{\in}A.\ g\ (Inl\ x)) + (\sum x{\in}B.\ g\ (Inr\ x))$
⟨*proof*⟩

lemma *setsum-UNIV-sum*:
fixes $g :: 'a{::}finite + 'b{::}finite \Rightarrow$ -
shows $(\sum x{\in}UNIV.\ g\ x) = (\sum x{\in}UNIV.\ g\ (Inl\ x)) + (\sum x{\in}UNIV.\ g\ (Inr\ x))$
⟨*proof*⟩

TODO: move to NthRoot

lemma *sqrt-add-le-add-sqrt*:
  assumes x: $0 \le x$ **and** y: $0 \le y$
  shows $sqrt\ (x + y) \le sqrt\ x + sqrt\ y$
⟨*proof*⟩

13.13 A generic notion of "hull" (convex, affine, conic hull and closure).

definition $hull :: 'a\ set\ set \Rightarrow 'a\ set \Rightarrow 'a\ set$ (**infixl** *hull* 75) **where**
  $S\ hull\ s = Inter\ \{t.\ t \in S \wedge s \subseteq t\}$

lemma *hull-same*: $s \in S \Longrightarrow S\ hull\ s = s$
  ⟨*proof*⟩

lemma *hull-in*: $(\bigwedge T.\ T \subseteq S ==> Inter\ T \in S) ==> (S\ hull\ s) \in S$
⟨*proof*⟩

lemma *hull-eq*: $(\bigwedge T.\ T \subseteq S ==> Inter\ T \in S) ==> (S\ hull\ s) = s \longleftrightarrow s \in S$
⟨*proof*⟩

lemma *hull-hull*: $S\ hull\ (S\ hull\ s) = S\ hull\ s$
  ⟨*proof*⟩

lemma *hull-subset*[*intro*]: $s \subseteq (S\ hull\ s)$
  ⟨*proof*⟩

lemma *hull-mono*: $s \subseteq t ==> (S\ hull\ s) \subseteq (S\ hull\ t)$
  ⟨*proof*⟩

lemma *hull-antimono*: $S \subseteq T ==> (T\ hull\ s) \subseteq (S\ hull\ s)$
  ⟨*proof*⟩

lemma *hull-minimal*: $s \subseteq t \Longrightarrow t \in S ==> (S\ hull\ s) \subseteq t$
  ⟨*proof*⟩

lemma *subset-hull*: $t \in S ==> S\ hull\ s \subseteq t \longleftrightarrow s \subseteq t$
  ⟨*proof*⟩

lemma *hull-unique*: $s \subseteq t \Longrightarrow t \in S \Longrightarrow (\bigwedge t'.\ s \subseteq t' \Longrightarrow t' \in S ==> t \subseteq t')$
    $==> (S\ hull\ s = t)$
⟨*proof*⟩

lemma *hull-induct*: $(\bigwedge x.\ x\in S \Longrightarrow P\ x) \Longrightarrow Q\ \{x.\ P\ x\} \Longrightarrow \forall x\in Q\ hull\ S.\ P\ x$
  ⟨*proof*⟩

lemma *hull-inc*: $x \in S \Longrightarrow x \in P\ hull\ S$ ⟨*proof*⟩

lemma *hull-union-subset*: $(S\ hull\ s) \cup (S\ hull\ t) \subseteq (S\ hull\ (s \cup t))$

⟨*proof*⟩

lemma *hull-union*: **assumes** T: $\bigwedge T.\ T \subseteq S \text{ ==> } \mathit{Inter}\ T \in S$
shows $S\ \mathit{hull}\ (s \cup t) = S\ \mathit{hull}\ (S\ \mathit{hull}\ s \cup S\ \mathit{hull}\ t)$
⟨*proof*⟩

lemma *hull-redundant-eq*: $a \in (S\ \mathit{hull}\ s) \longleftrightarrow (S\ \mathit{hull}\ (\mathit{insert}\ a\ s) = S\ \mathit{hull}\ s)$
⟨*proof*⟩

lemma *hull-redundant*: $a \in (S\ \mathit{hull}\ s) \text{ ==> } (S\ \mathit{hull}\ (\mathit{insert}\ a\ s) = S\ \mathit{hull}\ s)$
⟨*proof*⟩

Archimedian properties and useful consequences.

lemma *real-arch-simple*: $\exists\, n.\ x \text{ <= } \mathit{real}\ (n{::}\mathit{nat})$
⟨*proof*⟩
lemmas *real-arch-lt* = *reals-Archimedean2*

lemmas *real-arch* = *reals-Archimedean3*

lemma *real-arch-inv*: $0 < e \longleftrightarrow (\exists\, n{::}\mathit{nat}.\ n \neq 0 \wedge 0 < \mathit{inverse}\ (\mathit{real}\ n) \wedge \mathit{inverse}\ (\mathit{real}\ n) < e)$
⟨*proof*⟩

lemma *real-pow-lbound*: $0 \text{ <= } x \text{ ==> } 1 + \mathit{real}\ n * x \text{ <= } (1 + x)\ \hat{}\ n$
⟨*proof*⟩

lemma *real-arch-pow*: **assumes** x: $1 < (x{::}\mathit{real})$ **shows** $\exists\, n.\ y < x\hat{}n$
⟨*proof*⟩

lemma *real-arch-pow2*: $\exists\, n.\ (x{::}\mathit{real}) < 2\hat{}\ n$
⟨*proof*⟩

lemma *real-arch-pow-inv*: **assumes** y: $(y{::}\mathit{real}) > 0$ **and** $x1$: $x < 1$
shows $\exists\, n.\ x\hat{}n < y$
⟨*proof*⟩

lemma *forall-pos-mono*: $(\bigwedge d\ e{::}\mathit{real}.\ d < e \Longrightarrow P\ d \text{ ==> } P\ e) \Longrightarrow (\bigwedge n{::}\mathit{nat}.\ n \neq 0 \text{ ==> } P(\mathit{inverse}(\mathit{real}\ n))) \Longrightarrow (\bigwedge e.\ 0 < e \text{ ==> } P\ e)$
⟨*proof*⟩

lemma *forall-pos-mono-1*: $(\bigwedge d\ e{::}\mathit{real}.\ d < e \Longrightarrow P\ d \text{ ==> } P\ e) \Longrightarrow (\bigwedge n.\ P(\mathit{inverse}(\mathit{real}\ (\mathit{Suc}\ n)))) \text{ ==> } 0 < e \text{ ==> } P\ e$
⟨*proof*⟩

lemma *real-archimedian-rdiv-eq-0*: **assumes** $x0$: $x \geq 0$ **and** c: $c \geq 0$ **and** xc: $\forall\, (m{::}\mathit{nat}){>}0.\ \mathit{real}\ m * x \leq c$
shows $x = 0$
⟨*proof*⟩

13.14 Geometric progression

lemma *sum-gp-basic*: $((1::'a::\{field\}) - x) * setsum\ (\lambda i.\ x\hat{}i)\ \{0\ ..\ n\} = (1 - x\hat{}(Suc\ n))$
(**is** $?lhs = ?rhs$)
⟨*proof*⟩

lemma *sum-gp-multiplied*: **assumes** *mn*: $m <= n$
shows $((1::'a::\{field\}) - x) * setsum\ (op\ \hat{}\ x)\ \{m..n\} = x\hat{}m - x\hat{}\ Suc\ n$
(**is** $?lhs = ?rhs$)
⟨*proof*⟩

lemma *sum-gp*: $setsum\ (op\ \hat{}\ (x::'a::\{field\}))\ \{m\ ..\ n\} =$
$(if\ n < m\ then\ 0\ else\ if\ x = 1\ then\ of\text{-}nat\ ((n + 1) - m)$
$else\ (x\hat{}\ m - x\hat{}\ (Suc\ n))\ /\ (1 - x))$
⟨*proof*⟩

lemma *sum-gp-offset*: $setsum\ (op\ \hat{}\ (x::'a::\{field\}))\ \{m\ ..\ m+n\} =$
$(if\ x = 1\ then\ of\text{-}nat\ n + 1\ else\ x\hat{}m * (1 - x\hat{}Suc\ n)\ /\ (1 - x))$
⟨*proof*⟩

13.15 A bit of linear algebra.

definition
subspace :: *'a::real-vector set* ⇒ *bool* **where**
$subspace\ S \longleftrightarrow 0 \in S \land (\forall x\in S.\ \forall y \in S.\ x + y \in S) \land (\forall c.\ \forall x \in S.\ c *_R x \in S)$

definition $span\ S = (subspace\ hull\ S)$
definition $dependent\ S \longleftrightarrow (\exists a \in S.\ a \in span(S - \{a\}))$
abbreviation $independent\ s == \sim(dependent\ s)$

Closure properties of subspaces.

lemma *subspace-UNIV*[*simp*]: *subspace*(*UNIV*) ⟨*proof*⟩

lemma *subspace-0*: $subspace\ S ==> 0 \in S$ ⟨*proof*⟩

lemma *subspace-add*: $subspace\ S \Longrightarrow x \in S \Longrightarrow y \in S ==> x + y \in S$
⟨*proof*⟩

lemma *subspace-mul*: $subspace\ S \Longrightarrow x \in S \Longrightarrow c *_R x \in S$
⟨*proof*⟩

lemma *subspace-neg*: $subspace\ S \Longrightarrow x \in S \Longrightarrow - x \in S$
⟨*proof*⟩

lemma *subspace-sub*: $subspace\ S \Longrightarrow x \in S \Longrightarrow y \in S \Longrightarrow x - y \in S$
⟨*proof*⟩

lemma *subspace-setsum*:

assumes *sA*: *subspace A* **and** *fB*: *finite B*
and *f*: $\forall x \in B.\ f\ x \in A$
shows *setsum f B* $\in A$
⟨*proof*⟩

lemma *subspace-linear-image*:
assumes *lf*: *linear f* **and** *sS*: *subspace S*
shows *subspace*(f ' S)
⟨*proof*⟩

lemma *subspace-linear-preimage*: *linear f* ==> *subspace S* ==> *subspace* $\{x.\ f\ x \in S\}$
⟨*proof*⟩

lemma *subspace-trivial*: *subspace* $\{0\}$
⟨*proof*⟩

lemma *subspace-inter*: *subspace A* $\Longrightarrow$ *subspace B* ==> *subspace* $(A \cap B)$
⟨*proof*⟩

lemma *span-mono*: $A \subseteq B$ ==> *span A* $\subseteq$ *span B*
⟨*proof*⟩

lemma *subspace-span*: *subspace*(*span S*)
⟨*proof*⟩

lemma *span-clauses*:
$a \in S$ ==> $a \in$ *span S*
$0 \in$ *span S*
$x \in$ *span S* $\Longrightarrow$ $y \in$ *span S* ==> $x + y \in$ *span S*
$x \in$ *span S* $\Longrightarrow$ $c *_R x \in$ *span S*
⟨*proof*⟩

lemma *span-induct*: **assumes** *SP*: $\bigwedge x.\ x \in S$ ==> $P\ x$
and *P*: *subspace P* **and** *x*: $x \in$ *span S* **shows** $P\ x$
⟨*proof*⟩

lemma *span-empty*: *span* $\{\} = \{0\}$
⟨*proof*⟩

lemma *independent-empty*: *independent* $\{\}$
⟨*proof*⟩

lemma *independent-mono*: *independent A* $\Longrightarrow$ $B \subseteq A$ ==> *independent B*
⟨*proof*⟩

lemma *span-subspace*: $A \subseteq B \Longrightarrow B \leq$ *span A* $\Longrightarrow$ *subspace B* $\Longrightarrow$ *span* $A = B$
⟨*proof*⟩

lemma *span-induct'*: **assumes** *SP*: $\forall x \in S.\ P\ x$
and *P*: *subspace P* **shows** $\forall x \in span\ S.\ P\ x$
⟨*proof*⟩

inductive *span-induct-alt-help* **for** S:: $'a$::*real-vector* $\Rightarrow$ *bool*
where
span-induct-alt-help-0: *span-induct-alt-help S 0*
| *span-induct-alt-help-S*: $x \in S \Longrightarrow$ *span-induct-alt-help S z* $\Longrightarrow$ *span-induct-alt-help S* $(c *_R x + z)$

lemma *span-induct-alt'*:
assumes *h0*: *h 0* **and** *hS*: $\bigwedge c\ x\ y.\ x \in S \Longrightarrow h\ y \Longrightarrow h\ (c *_R x + y)$ **shows** $\forall x \in span\ S.\ h\ x$
⟨*proof*⟩

lemma *span-induct-alt*:
assumes *h0*: *h 0* **and** *hS*: $\bigwedge c\ x\ y.\ x \in S \Longrightarrow h\ y \Longrightarrow h\ (c *_R x + y)$ **and** *x*: $x \in span\ S$
shows *h x*
⟨*proof*⟩

Individual closure properties.

lemma *span-superset*: $x \in S$ ==> $x \in span\ S$ ⟨*proof*⟩

lemma *span-0*: $0 \in span\ S$ ⟨*proof*⟩

lemma *span-add*: $x \in span\ S \Longrightarrow y \in span\ S$ ==> $x + y \in span\ S$
⟨*proof*⟩

lemma *span-mul*: $x \in span\ S$ ==> $(c *_R x) \in span\ S$
⟨*proof*⟩

lemma *span-neg*: $x \in span\ S$ ==> $-\ x \in span\ S$
⟨*proof*⟩

lemma *span-sub*: $x \in span\ S \Longrightarrow y \in span\ S$ ==> $x - y \in span\ S$
⟨*proof*⟩

lemma *span-setsum*: *finite A* $\Longrightarrow \forall x \in A.\ f\ x \in span\ S$ ==> *setsum f A* $\in span\ S$
⟨*proof*⟩

lemma *span-add-eq*: $x \in span\ S \Longrightarrow x + y \in span\ S \longleftrightarrow y \in span\ S$
⟨*proof*⟩

Mapping under linear image.

lemma *span-linear-image*: **assumes** *lf*: *linear f*
shows *span* $(f\ `\ S) = f\ `\ (span\ S)$

⟨*proof*⟩

The key breakdown property.

lemma *span-breakdown*:
assumes *bS*: $b \in S$ **and** *aS*: $a \in span\ S$
shows $\exists k.\ a - k *_R b \in span\ (S - \{b\})$ (**is** *?P a*)
⟨*proof*⟩

lemma *span-breakdown-eq*:
$x \in span\ (insert\ a\ S) \longleftrightarrow (\exists k.\ (x - k *_R a) \in span\ S)$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

Hence some "reversal" results.

lemma *in-span-insert*:
assumes *a*: $a \in span\ (insert\ b\ S)$ **and** *na*: $a \notin span\ S$
shows $b \in span\ (insert\ a\ S)$
⟨*proof*⟩

lemma *in-span-delete*:
assumes *a*: $a \in span\ S$
and *na*: $a \notin span\ (S - \{b\})$
shows $b \in span\ (insert\ a\ (S - \{b\}))$
⟨*proof*⟩

Transitivity property.

lemma *span-trans*:
assumes *x*: $x \in span\ S$ **and** *y*: $y \in span\ (insert\ x\ S)$
shows $y \in span\ S$
⟨*proof*⟩

An explicit expansion is sometimes needed.

lemma *span-explicit*:
$span\ P = \{y.\ \exists S\ u.\ finite\ S \land S \subseteq P \land setsum\ (\lambda v.\ u\ v *_R v)\ S = y\}$
(**is** - = *?E* **is** - = $\{y.\ ?h\ y\}$ **is** - = $\{y.\ \exists S\ u.\ ?Q\ S\ u\ y\}$)
⟨*proof*⟩

lemma *dependent-explicit*:
$dependent\ P \longleftrightarrow (\exists S\ u.\ finite\ S \land S \subseteq P \land (\exists v \in S.\ u\ v \neq 0 \land setsum\ (\lambda v.\ u\ v *_R v)\ S = 0))$ (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *span-finite*:
assumes *fS*: *finite S*
shows $span\ S = \{y.\ \exists u.\ setsum\ (\lambda v.\ u\ v *_R v)\ S = y\}$
(**is** - = *?rhs*)
⟨*proof*⟩

Standard bases are a spanning set, and obviously finite.

lemma *span-stdbasis*:*span* $\{basis\ i :: real\hat{}'n \mid i.\ i \in (UNIV :: 'n\ set)\} = UNIV$
⟨*proof*⟩

lemma *finite-stdbasis*: *finite* $\{basis\ i ::real\hat{}'n \mid i.\ i\in (UNIV:: 'n\ set)\}$ (**is** *finite* *?S*)
⟨*proof*⟩

lemma *card-stdbasis*: *card* $\{basis\ i ::real\hat{}'n \mid i.\ i\in (UNIV :: 'n\ set)\} = CARD('n)$ (**is** *card ?S* = -)
⟨*proof*⟩

lemma *independent-stdbasis-lemma*:
  assumes *x*: $(x::real\ \hat{}\ 'n) \in span\ (basis\ `\ S)$
  and *iS*: $i \notin S$
  shows $(x\$i) = 0$
⟨*proof*⟩

lemma *independent-stdbasis*: *independent* $\{basis\ i ::real\hat{}'n \mid i.\ i\in (UNIV :: 'n\ set)\}$
⟨*proof*⟩

This is useful for building a basis step-by-step.

lemma *independent-insert*:
  $independent(insert\ a\ S) \longleftrightarrow$
    $(if\ a \in S\ then\ independent\ S$
        $else\ independent\ S \wedge a \notin span\ S)$ (**is** $?lhs \longleftrightarrow ?rhs$)
⟨*proof*⟩

The degenerate case of the Exchange Lemma.

lemma *mem-delete*: $x \in (A - \{a\}) \longleftrightarrow x \neq a \wedge x \in A$
  ⟨*proof*⟩

lemma *span-span*: $span\ (span\ A) = span\ A$
  ⟨*proof*⟩

lemma *span-inc*: $S \subseteq span\ S$
  ⟨*proof*⟩

lemma *spanning-subset-independent*:
  assumes *BA*: $B \subseteq A$ **and** *iA*: *independent A*
  and *AsB*: $A \subseteq span\ B$
  shows $A = B$
⟨*proof*⟩

The general case of the Exchange Lemma, the key to what follows.

lemma *exchange-lemma*:
  assumes *f*:*finite t* **and** *i*: *independent s*
  and *sp*:$s \subseteq span\ t$

shows $\exists\, t'.\ (card\ t' = card\ t) \wedge finite\ t' \wedge s \subseteq t' \wedge t' \subseteq s \cup t \wedge s \subseteq span\ t'$
$\langle proof \rangle$

This implies corresponding size bounds.

lemma *independent-span-bound*:
assumes f: *finite t* **and** i: *independent s* **and** sp:$s \subseteq span\ t$
shows $finite\ s \wedge card\ s \le card\ t$
$\langle proof \rangle$

lemma *finite-Atleast-Atmost-nat*[*simp*]: $finite\ \{f\ x\ |x.\ x \in (UNIV::'a::finite\ set)\}$
$\langle proof \rangle$

lemma *independent-bound*:
fixes S:: $(real\hat{\ }'n)\ set$
shows $independent\ S \Longrightarrow finite\ S \wedge card\ S <= CARD('n)$
$\langle proof \rangle$

lemma *dependent-biggerset*: $(finite\ (S::(real\ \hat{\ }'n)\ set) ==> card\ S > CARD('n))$ $==> dependent\ S$
$\langle proof \rangle$

Hence we can create a maximal independent subset.

lemma *maximal-independent-subset-extend*:
assumes sv: $(S::(real\hat{\ }'n)\ set) \subseteq V$ **and** iS: *independent S*
shows $\exists\, B.\ S \subseteq B \wedge B \subseteq V \wedge independent\ B \wedge V \subseteq span\ B$
$\langle proof \rangle$

lemma *maximal-independent-subset*:
$\exists\,(B::\ (real\ \hat{\ }'n)\ set).\ B \subseteq V \wedge independent\ B \wedge V \subseteq span\ B$
$\langle proof \rangle$

Notion of dimension.

definition $dim\ V = (SOME\ n.\ \exists\, B.\ B \subseteq V \wedge independent\ B \wedge V \subseteq span\ B \wedge (card\ B = n))$

lemma *basis-exists*: $\exists\, B.\ (B :: (real\ \hat{\ }'n)\ set) \subseteq V \wedge independent\ B \wedge V \subseteq span\ B \wedge (card\ B = dim\ V)$
$\langle proof \rangle$

Consequences of independence or spanning for cardinality.

lemma *independent-card-le-dim*:
assumes $(B::(real\ \hat{\ }'n)\ set) \subseteq V$ **and** *independent B* **shows** $card\ B \le dim\ V$
$\langle proof \rangle$

lemma *span-card-ge-dim*: $(B::(real\ \hat{\ }'n)\ set) \subseteq V \Longrightarrow V \subseteq span\ B \Longrightarrow finite\ B \Longrightarrow dim\ V \le card\ B$

⟨*proof*⟩

lemma *basis-card-eq-dim*:
$B \subseteq (V:: (real\ \hat{}\ 'n)\ set) \Longrightarrow V \subseteq span\ B \Longrightarrow independent\ B \Longrightarrow finite\ B \wedge card\ B = dim\ V$
⟨*proof*⟩

lemma *dim-unique*: $(B::(real\ \hat{}\ 'n)\ set) \subseteq V \Longrightarrow V \subseteq span\ B \Longrightarrow independent\ B \Longrightarrow card\ B = n \Longrightarrow dim\ V = n$
⟨*proof*⟩

More lemmas about dimension.

lemma *dim-univ*: $dim\ (UNIV :: (real\hat{}'n)\ set) = CARD('n)$
⟨*proof*⟩

lemma *dim-subset*:
$(S:: (real\ \hat{}\ 'n)\ set) \subseteq T \Longrightarrow dim\ S \leq dim\ T$
⟨*proof*⟩

lemma *dim-subset-univ*: $dim\ (S:: (real\hat{}'n)\ set) \leq CARD('n)$
⟨*proof*⟩

Converses to those.

lemma *card-ge-dim-independent*:
assumes BV:$(B::(real\ \hat{}\ 'n)\ set) \subseteq V$ **and** iB:$independent\ B$ **and** dVB:$dim\ V \leq card\ B$
shows $V \subseteq span\ B$
⟨*proof*⟩

lemma *card-le-dim-spanning*:
assumes BV: $(B:: (real\ \hat{}\ 'n)\ set) \subseteq V$ **and** VB: $V \subseteq span\ B$
and fB: $finite\ B$ **and** dVB: $dim\ V \geq card\ B$
shows $independent\ B$
⟨*proof*⟩

lemma *card-eq-dim*: $(B:: (real\ \hat{}\ 'n)\ set) \subseteq V \Longrightarrow card\ B = dim\ V \Longrightarrow finite\ B \Longrightarrow independent\ B \longleftrightarrow V \subseteq span\ B$
⟨*proof*⟩

More general size bound lemmas.

lemma *independent-bound-general*:
$independent\ (S:: (real\hat{}'n)\ set) \Longrightarrow finite\ S \wedge card\ S \leq dim\ S$
⟨*proof*⟩

lemma *dependent-biggerset-general*: $(finite\ (S:: (real\hat{}'n)\ set) \Longrightarrow card\ S > dim\ S) \Longrightarrow dependent\ S$
⟨*proof*⟩

lemma *dim-span*: $dim\ (span\ (S:: (real\ \hat{}\ 'n)\ set)) = dim\ S$

⟨*proof*⟩

lemma *subset-le-dim*: $(S{::}\ (real\ \hat{}'n)\ set) \subseteq span\ T \Longrightarrow dim\ S \leq dim\ T$
⟨*proof*⟩

lemma *span-eq-dim*: $span\ (S{::}\ (real\ \hat{}'n)\ set) = span\ T ==> dim\ S = dim\ T$
⟨*proof*⟩

lemma *spans-image*:
assumes *lf*: *linear f* **and** *VB*: $V \subseteq span\ B$
shows $f\ `\ V \subseteq span\ (f\ `\ B)$
⟨*proof*⟩

lemma *dim-image-le*:
fixes $f :: real\hat{}'n \Rightarrow real\hat{}'m$
assumes *lf*: *linear f* **shows** $dim\ (f\ `\ S) \leq dim\ (S{::}\ (real\ \hat{}'n)\ set)$
⟨*proof*⟩

Relation between bases and injectivity/surjectivity of map.

lemma *spanning-surjective-image*:
assumes *us*: $UNIV \subseteq span\ S$
and *lf*: *linear f* **and** *sf*: *surj f*
shows $UNIV \subseteq span\ (f\ `\ S)$
⟨*proof*⟩

lemma *independent-injective-image*:
assumes *iS*: *independent S* **and** *lf*: *linear f* **and** *fi*: *inj f*
shows *independent* $(f\ `\ S)$
⟨*proof*⟩

Picking an orthogonal replacement for a spanning set.

definition $pairwise\ R\ S \longleftrightarrow (\forall x \in S.\ \forall y \in S.\ x \neq y \longrightarrow R\ x\ y)$

lemma *vector-sub-project-orthogonal*: $(b{::}real\hat{}'n) \cdot (x - ((b \cdot x) / (b \cdot b)) *s\ b) = 0$
⟨*proof*⟩

lemma *basis-orthogonal*:
fixes $B :: (real\ \hat{}'n)\ set$
assumes *fB*: *finite B*
shows $\exists C.\ finite\ C \wedge card\ C \leq card\ B \wedge span\ C = span\ B \wedge pairwise\ orthogonal\ C$
(**is** $\exists C.\ ?P\ B\ C$)
⟨*proof*⟩
thm *pairwise-def*
⟨*proof*⟩

lemma *orthogonal-basis-exists*:
fixes $V :: (real\ \hat{}'n)\ set$

shows $\exists B.$ *independent* $B \wedge B \subseteq$ *span* $V \wedge V \subseteq$ *span* $B \wedge$ (*card* B = *dim* V)
$\wedge$ *pairwise orthogonal* B
⟨*proof*⟩

lemma *span-eq*: *span* S = *span* $T \longleftrightarrow S \subseteq$ *span* $T \wedge T \subseteq$ *span* S
⟨*proof*⟩

Low-dimensional subset is in a hyperplane (weak orthogonal complement).

lemma *span-not-univ-orthogonal*:
assumes *sU*: *span* $S \neq$ *UNIV*
shows $\exists$(*a*:: *real* ^′*n*). $a \neq 0 \wedge (\forall x \in$ *span* $S.\ a \cdot x = 0)$
⟨*proof*⟩

lemma *span-not-univ-subset-hyperplane*:
assumes *SU*: *span* $S \neq$ (*UNIV* ::(*real*^′*n*) *set*)
shows $\exists\ a.\ a \neq 0 \wedge$ *span* $S \subseteq \{x.\ a \cdot x = 0\}$
⟨*proof*⟩

lemma *lowdim-subset-hyperplane*:
assumes *d*: *dim* $S <$ *CARD*(′*n*::*finite*)
shows $\exists$(*a*::*real* ^′*n*). $a \neq 0 \wedge$ *span* $S \subseteq \{x.\ a \cdot x = 0\}$
⟨*proof*⟩

We can extend a linear basis-basis injection to the whole set.

lemma *linear-indep-image-lemma*:
assumes *lf*: *linear f* **and** *fB*: *finite B*
and *ifB*: *independent* (*f* ‘ *B*)
and *fi*: *inj-on f B* **and** *xsB*: $x \in$ *span B*
and *fx*: $f\ x = 0$
shows $x = 0$
⟨*proof*⟩

We can extend a linear mapping from basis.

lemma *linear-independent-extend-lemma*:
fixes *f* :: ′*a*::*real-vector* ⇒ ′*b*::*real-vector*
assumes *fi*: *finite B* **and** *ib*: *independent B*
shows $\exists g.\ (\forall x \in$ *span* $B.\ \forall y \in$ *span* $B.\ g\ (x + y) = g\ x + g\ y)$
$\wedge\ (\forall x \in$ *span* $B.\ \forall c.\ g\ (c *_R x) = c *_R g\ x)$
$\wedge\ (\forall x \in B.\ g\ x = f\ x)$
⟨*proof*⟩

lemma *linear-independent-extend*:
assumes *iB*: *independent* (*B*:: (*real* ^′*n*) *set*)
shows $\exists g.$ *linear* $g \wedge (\forall x \in B.\ g\ x = f\ x)$
⟨*proof*⟩

Can construct an isomorphism between spaces of same dimension.

lemma *card-le-inj*: **assumes** *fA*: *finite A* **and** *fB*: *finite B*

and *c*: *card* $A \leq$ *card* B **shows** $(\exists f.\ f$ ‘ $A \subseteq B \wedge$ *inj-on* $f\ A)$
⟨*proof*⟩

lemma *card-subset-eq*: **assumes** *fB*: *finite* B **and** *AB*: $A \subseteq B$ **and**
c: *card* $A =$ *card* B
shows $A = B$
⟨*proof*⟩

lemma *subspace-isomorphism*:
assumes *s*: *subspace* $(S$:: $(real\ \hat{}'n)\ set)$
and *t*: *subspace* $(T$ :: $(real\ \hat{}'m)\ set)$
and *d*: *dim* $S =$ *dim* T
shows $\exists f.$ *linear* $f \wedge f$ ‘ $S = T \wedge$ *inj-on* $f\ S$
⟨*proof*⟩

Linear functions are equal on a subspace if they are on a spanning set.

lemma *subspace-kernel*:
assumes *lf*: *linear* f
shows *subspace* $\{x.\ f\ x = 0\}$
⟨*proof*⟩

lemma *linear-eq-0-span*:
assumes *lf*: *linear* f **and** *f0*: $\forall x{\in}B.\ f\ x = 0$
shows $\forall x \in$ *span* $B.\ f\ x = 0$
⟨*proof*⟩

lemma *linear-eq-0*:
assumes *lf*: *linear* f **and** *SB*: $S \subseteq$ *span* B **and** *f0*: $\forall x{\in}B.\ f\ x = 0$
shows $\forall x \in S.\ f\ x = 0$
⟨*proof*⟩

lemma *linear-eq*:
assumes *lf*: *linear* f **and** *lg*: *linear* g **and** S: $S \subseteq$ *span* B
and *fg*: $\forall\ x{\in}\ B.\ f\ x = g\ x$
shows $\forall x{\in}\ S.\ f\ x = g\ x$
⟨*proof*⟩

lemma *linear-eq-stdbasis*:
assumes *lf*: *linear* $(f$::$real\hat{}'m \Rightarrow$ -$)$ **and** *lg*: *linear* g
and *fg*: $\forall i.\ f\ (basis\ i) = g(basis\ i)$
shows $f = g$
⟨*proof*⟩

Similar results for bilinear functions.

lemma *bilinear-eq*:
assumes *bf*: *bilinear* f
and *bg*: *bilinear* g
and *SB*: $S \subseteq$ *span* B **and** *TC*: $T \subseteq$ *span* C
and *fg*: $\forall x{\in}\ B.\ \forall y{\in}\ C.\ f\ x\ y = g\ x\ y$

shows $\forall x{\in}S.\ \forall y{\in}T.\ f\ x\ y = g\ x\ y$
⟨*proof*⟩

lemma *bilinear-eq-stdbasis*:
assumes *bf*: *bilinear* (f:: *real*ˆ′*m* ⇒ *real*ˆ′*n* ⇒ -)
and *bg*: *bilinear g*
and *fg*: $\forall\ i\ j.\ f$ (*basis i*) (*basis j*) = g (*basis i*) (*basis j*)
shows $f = g$
⟨*proof*⟩

Detailed theorems about left and right invertibility in general case.

lemma *left-invertible-transpose*:
(∃(B). B ∗∗ *transpose* (A) = *mat* (*1*::′*a*::*comm-semiring-1*)) ⟷ (∃(B). A ∗∗ B = *mat 1*)
⟨*proof*⟩

lemma *right-invertible-transpose*:
(∃(B). *transpose* (A) ∗∗ B = *mat* (*1*::′*a*::*comm-semiring-1*)) ⟷ (∃(B). B ∗∗ A = *mat 1*)
⟨*proof*⟩

lemma *linear-injective-left-inverse*:
assumes *lf*: *linear* (*f*::*real* ˆ′*n* ⇒ *real* ˆ′*m*) **and** *fi*: *inj f*
shows ∃ *g*. *linear g* ∧ *g o f* = *id*
⟨*proof*⟩

lemma *linear-surjective-right-inverse*:
assumes *lf*: *linear* (*f*:: *real* ˆ′*m* ⇒ *real* ˆ′*n*) **and** *sf*: *surj f*
shows ∃ *g*. *linear g* ∧ *f o g* = *id*
⟨*proof*⟩

lemma *matrix-left-invertible-injective*:
(∃ B. (B::*real*ˆ′*m*ˆ′*n*) ∗∗ (A::*real*ˆ′*n*ˆ′*m*) = *mat 1*) ⟷ (∀ x y. A ∗*v* x = A ∗*v* y ⟶ x = y)
⟨*proof*⟩

lemma *matrix-left-invertible-ker*:
(∃ B. (B::*real* ˆ′*m*ˆ′*n*) ∗∗ (A::*real*ˆ′*n*ˆ′*m*) = *mat 1*) ⟷ (∀ x. A ∗*v* x = *0* ⟶ x = *0*)
⟨*proof*⟩

lemma *matrix-right-invertible-surjective*:
(∃ B. (A::*real*ˆ′*n*ˆ′*m*) ∗∗ (B::*real*ˆ′*m*ˆ′*n*) = *mat 1*) ⟷ *surj* (λx. A ∗*v* x)
⟨*proof*⟩

lemma *matrix-left-invertible-independent-columns*:
fixes A :: *real*ˆ′*n*ˆ′*m*
shows (∃(B::*real* ˆ′*m*ˆ′*n*). B ∗∗ A = *mat 1*) ⟷ (∀ c. *setsum* (λi. c i ∗*s column* i A) (*UNIV* :: ′*n set*) = *0* ⟶ (∀ i. c i = *0*))

(**is** *?lhs* $\longleftrightarrow$ *?rhs*)
$\langle proof \rangle$

lemma *matrix-right-invertible-independent-rows*:
fixes A :: *real*$\hat{}$*'n*$\hat{}$*'m*
shows $(\exists (B::real\hat{}'m\hat{}'n).\ A ** B = mat\ 1) \longleftrightarrow (\forall c.\ setsum\ (\lambda i.\ c\ i *s\ row\ i\ A)\ (UNIV :: 'm\ set) = 0 \longrightarrow (\forall i.\ c\ i = 0))$
$\langle proof \rangle$

lemma *matrix-right-invertible-span-columns*:
$(\exists (B::real\ \hat{}'n\hat{}'m).\ (A::real\ \hat{}'m\hat{}'n) ** B = mat\ 1) \longleftrightarrow span\ (columns\ A) = UNIV$ (**is** *?lhs* = *?rhs*)
$\langle proof \rangle$

lemma *matrix-left-invertible-span-rows*:
$(\exists (B::real\hat{}'m\hat{}'n).\ B ** (A::real\hat{}'n\hat{}'m) = mat\ 1) \longleftrightarrow span\ (rows\ A) = UNIV$
$\langle proof \rangle$

An injective map $(real, 'n)\ cart \Rightarrow (real, 'n)\ cart$ is also surjective.

lemma *linear-injective-imp-surjective*:
assumes *lf*: *linear* $(f::\ real\ \hat{}'n \Rightarrow real\ \hat{}'n)$ **and** *fi*: *inj f*
shows *surj f*
$\langle proof \rangle$

And vice versa.

lemma *surjective-iff-injective-gen*:
assumes *fS*: *finite S* **and** *fT*: *finite T* **and** *c*: *card S* = *card T*
and *ST*: $f\ `\ S \subseteq T$
shows $(\forall y \in T.\ \exists x \in S.\ f\ x = y) \longleftrightarrow inj\text{-}on\ f\ S$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
$\langle proof \rangle$

lemma *linear-surjective-imp-injective*:
assumes *lf*: *linear* $(f::real\ \hat{}'n => real\ \hat{}'n)$ **and** *sf*: *surj f*
shows *inj f*
$\langle proof \rangle$

Hence either is enough for isomorphism.

lemma *left-right-inverse-eq*:
assumes *fg*: $f\ o\ g = id$ **and** *gh*: $g\ o\ h = id$
shows $f = h$
$\langle proof \rangle$

lemma *isomorphism-expand*:
$f\ o\ g = id \land g\ o\ f = id \longleftrightarrow (\forall x.\ f(g\ x) = x) \land (\forall x.\ g(f\ x) = x)$
$\langle proof \rangle$

lemma *linear-injective-isomorphism*:
assumes *lf*: *linear* $(f :: real\hat{}'n \Rightarrow real\ \hat{}'n)$ **and** *fi*: *inj f*
shows $\exists f'.\ linear\ f' \land (\forall x.\ f'\ (f\ x) = x) \land (\forall x.\ f\ (f'\ x) = x)$

⟨*proof*⟩

lemma *linear-surjective-isomorphism*:
  assumes *lf*: *linear* (f::*real* ^'*n* ⇒ *real* ^'*n*) **and** *sf*: *surj f*
  shows $\exists f'.$ *linear* $f' \wedge (\forall x.\ f'\ (f\ x) = x) \wedge (\forall x.\ f\ (f'\ x) = x)$
⟨*proof*⟩

Left and right inverses are the same for (*real*, '*n*) *cart* ⇒ (*real*, '*n*) *cart*.

lemma *linear-inverse-left*:
  assumes *lf*: *linear* (f::*real* ^'*n* ⇒ *real* ^'*n*) **and** *lf'*: *linear* f'
  shows $f\ o\ f' = id \longleftrightarrow f'\ o\ f = id$
⟨*proof*⟩

Moreover, a one-sided inverse is automatically linear.

lemma *left-inverse-linear*:
  assumes *lf*: *linear* (f::*real* ^'*n* ⇒ *real* ^'*n*) **and** *gf*: $g\ o\ f = id$
  shows *linear g*
⟨*proof*⟩

lemma *right-inverse-linear*:
  assumes *lf*: *linear* (f:: *real* ^'*n* ⇒ *real* ^'*n*) **and** *gf*: $f\ o\ g = id$
  shows *linear g*
⟨*proof*⟩

The same result in terms of square matrices.

lemma *matrix-left-right-inverse*:
  fixes $A\ A'$ :: *real* ^'*n*^'*n*
  shows $A ** A' = mat\ 1 \longleftrightarrow A' ** A = mat\ 1$
⟨*proof*⟩

Considering an n-element vector as an n-by-1 or 1-by-n matrix.

definition *rowvector* $v = (\chi\ i\ j.\ (v\$j))$

definition *columnvector* $v = (\chi\ i\ j.\ (v\$i))$

lemma *transpose-columnvector*:
 transpose(*columnvector v*) = *rowvector v*
 ⟨*proof*⟩

lemma *transpose-rowvector*: *transpose*(*rowvector v*) = *columnvector v*
 ⟨*proof*⟩

lemma *dot-rowvector-columnvector*:
 columnvector (A *∗v* v) = A ∗∗ *columnvector v*
 ⟨*proof*⟩

lemma *dot-matrix-product*: (x::*real*^'*n*) · y = (((*rowvector x* ::*real*^'*n*^*1*) ∗∗ (*columnvector* y :: *real*^*1*^'*n*))\$*1*)\$*1*

⟨*proof*⟩

lemma *dot-matrix-vector-mul*:
fixes $A\ B :: real\ \hat{}'n\ \hat{}'n$ **and** $x\ y :: real\ \hat{}'n$
shows $(A *v\ x) \cdot (B *v\ y) =$
$(((rowvector\ x :: real\hat{}'n\hat{}1) ** ((transpose\ A ** B) ** (columnvector\ y :: real\ \hat{}1\hat{}'n)))\$1)\$1$
⟨*proof*⟩

13.16 Infinity norm

definition $infnorm\ (x::real\hat{}'n) = Sup\ \{abs(x\$i)\ |i.\ i\in (UNIV :: 'n\ set)\}$

lemma *numseg-dimindex-nonempty*: $\exists\, i.\ i \in (UNIV :: 'n\ set)$
⟨*proof*⟩

lemma *infnorm-set-image*:
$\{abs(x\$i)\ |i.\ i\in (UNIV :: \text{-}\ set)\} =$
$(\lambda i.\ abs(x\$i))$ ‘ $(UNIV)$ ⟨*proof*⟩

lemma *infnorm-set-lemma*:
shows $finite\ \{abs((x::'a::abs\ \hat{}'n)\$i)\ |i.\ i\in (UNIV :: 'n\ set)\}$
and $\{abs(x\$i)\ |i.\ i\in (UNIV :: 'n::finite\ set)\} \neq \{\}$
⟨*proof*⟩

lemma *infnorm-pos-le*: $0 \le infnorm\ (x::real\hat{}'n)$
⟨*proof*⟩

lemma *infnorm-triangle*: $infnorm\ ((x::real\hat{}'n) + y) \le infnorm\ x + infnorm\ y$
⟨*proof*⟩

lemma *infnorm-eq-0*: $infnorm\ x = 0 \longleftrightarrow (x::real\ \hat{}'n) = 0$
⟨*proof*⟩

lemma *infnorm-0*: $infnorm\ 0 = 0$
⟨*proof*⟩

lemma *infnorm-neg*: $infnorm\ (-\ x) = infnorm\ x$
⟨*proof*⟩

lemma *infnorm-sub*: $infnorm\ (x - y) = infnorm\ (y - x)$
⟨*proof*⟩

lemma *real-abs-sub-infnorm*: $|\ infnorm\ x - infnorm\ y| \le infnorm\ (x - y)$
⟨*proof*⟩

lemma *real-abs-infnorm*: $|infnorm\ x| = infnorm\ x$
⟨*proof*⟩

lemma *component-le-infnorm*:
shows $|x\$i| \le$ *infnorm* $(x::real\hat{\ }'n)$
⟨*proof*⟩

lemma *infnorm-mul-lemma*: *infnorm*$(a *s\ x) <= |a| *$ *infnorm* x
⟨*proof*⟩

lemma *infnorm-mul*: *infnorm*$(a *s\ x) = abs\ a *$ *infnorm* x
⟨*proof*⟩

lemma *infnorm-pos-lt*: *infnorm* $x > 0 \longleftrightarrow x \neq 0$
⟨*proof*⟩

Prove that it differs only up to a bound from Euclidean norm.

lemma *infnorm-le-norm*: *infnorm* $x \le$ *norm* x
⟨*proof*⟩
lemma *card-enum*: *card* $\{1\ ..\ n\} = n$ ⟨*proof*⟩
lemma *norm-le-infnorm*: *norm*$(x) <=$ *sqrt*(*real* *CARD*$('n)) *$ *infnorm*$(x::real\ \hat{\ }'n)$
⟨*proof*⟩

Equality in Cauchy-Schwarz and triangle inequalities.

lemma *norm-cauchy-schwarz-eq*: $x \cdot y =$ *norm* $x *$ *norm* $y \longleftrightarrow$ *norm* $x *_R y =$ *norm* $y *_R x$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

lemma *norm-cauchy-schwarz-abs-eq*:
shows *abs*$(x \cdot y) =$ *norm* $x *$ *norm* $y \longleftrightarrow$
norm $x *_R y =$ *norm* $y *_R x \vee$ *norm*$(x) *_R y = -$ *norm* $y *_R x$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

lemma *norm-triangle-eq*:
fixes $x\ y :: 'a::real\text{-}inner$
shows *norm*$(x + y) =$ *norm* $x +$ *norm* $y \longleftrightarrow$ *norm* $x *_R y =$ *norm* $y *_R x$
⟨*proof*⟩

13.17 Collinearity

definition
collinear :: $'a::real\text{-}vector\ set \Rightarrow bool$ **where**
collinear $S \longleftrightarrow (\exists u.\ \forall x \in S.\ \forall\ y \in S.\ \exists c.\ x - y = c *_R u)$

lemma *collinear-empty*: *collinear* {} ⟨*proof*⟩

lemma *collinear-sing*: *collinear* $\{x\}$
⟨*proof*⟩

lemma *collinear-2*: *collinear* $\{x, y\}$
⟨*proof*⟩

lemma *collinear-lemma*: *collinear* $\{0,x,y\} \longleftrightarrow x = 0 \vee y = 0 \vee (\exists c.\ y = c *_R x)$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

lemma *norm-cauchy-schwarz-equal*:
shows $abs(x \cdot y) = norm\ x * norm\ y \longleftrightarrow collinear\ \{0,x,y\}$
⟨*proof*⟩

end

14 Permutations: Permutations, both general and specifically on finite sets.

theory *Permutations*
imports *Parity Fact*
begin

definition *permutes* (**infixr** *permutes 41*) **where**
$(p\ permutes\ S) \longleftrightarrow (\forall x.\ x \notin S \longrightarrow p\ x = x) \wedge (\forall y.\ \exists!x.\ p\ x = y)$

lemma *swapid-sym*: *Fun.swap a b id* = *Fun.swap b a id*
⟨*proof*⟩
lemma *swap-id-refl*: *Fun.swap a a id* = *id* ⟨*proof*⟩
lemma *swap-id-sym*: *Fun.swap a b id* = *Fun.swap b a id*
⟨*proof*⟩
lemma *swap-id-idempotent*[*simp*]: *Fun.swap a b id o Fun.swap a b id* = *id*
⟨*proof*⟩

lemma *inv-unique-comp*: **assumes** *fg*: *f o g* = *id* **and** *gf*: *g o f* = *id*
shows *inv f* = *g*
⟨*proof*⟩

lemma *inverse-swap-id*: *inv* (*Fun.swap a b id*) = *Fun.swap a b id*
⟨*proof*⟩

lemma *swap-id-eq*: *Fun.swap a b id x* = (*if x* = *a then b else if x* = *b then a else x*)
⟨*proof*⟩

lemma *permutes-in-image*: p *permutes* $S \Longrightarrow p\ x \in S \longleftrightarrow x \in S$
⟨*proof*⟩

lemma *permutes-image*: **assumes** *pS*: p *permutes* S **shows** p ‘ $S = S$
⟨*proof*⟩

lemma *permutes-inj*: p *permutes* S ==> *inj* p
⟨*proof*⟩

lemma *permutes-surj*: p *permutes* s ==> *surj* p
⟨*proof*⟩

lemma *permutes-inv-o*: **assumes** *pS*: p *permutes* S
shows $p\ o\ inv\ p = id$
and $inv\ p\ o\ p = id$
⟨*proof*⟩

lemma *permutes-inverses*:
fixes $p :: 'a \Rightarrow 'a$
assumes *pS*: p *permutes* S
shows $p\ (inv\ p\ x) = x$
and $inv\ p\ (p\ x) = x$
⟨*proof*⟩

lemma *permutes-subset*: p *permutes* $S \Longrightarrow S \subseteq T$ ==> p *permutes* T
⟨*proof*⟩

lemma *permutes-empty*[*simp*]: p *permutes* $\{\} \longleftrightarrow p = id$
⟨*proof*⟩

lemma *permutes-sing*[*simp*]: p *permutes* $\{a\} \longleftrightarrow p = id$
⟨*proof*⟩

lemma *permutes-univ*: p *permutes* *UNIV* $\longleftrightarrow (\forall\, y.\ \exists !x.\ p\ x = y)$
⟨*proof*⟩

lemma *permutes-inv-eq*: p *permutes* S ==> $inv\ p\ y = x \longleftrightarrow p\ x = y$
⟨*proof*⟩

lemma *permutes-swap-id*: $a \in S \Longrightarrow b \in S$ ==> *Fun.swap* $a\ b$ *id permutes* S
⟨*proof*⟩

lemma *permutes-superset*:
p *permutes* $S \Longrightarrow (\forall\, x \in S - T.\ p\ x = x) \Longrightarrow p$ *permutes* T
⟨*proof*⟩

lemma *permutes-id*: *id permutes S* ⟨*proof*⟩

lemma *permutes-compose*: *p permutes S* $\Longrightarrow$ *q permutes S ==> q o p permutes S*
⟨*proof*⟩

lemma *permutes-inv*: **assumes** *pS*: *p permutes S* **shows** *inv p permutes S*
⟨*proof*⟩

lemma *permutes-inv-inv*: **assumes** *pS*: *p permutes S* **shows** *inv* (*inv p*) = *p*
⟨*proof*⟩

lemma *permutes-insert-lemma*:
assumes *pS*: *p permutes* (*insert a S*)
shows *Fun.swap a* (*p a*) *id o p permutes S*
⟨*proof*⟩

lemma *permutes-insert*: {*p*. *p permutes* (*insert a S*)} =
(λ(*b*,*p*). *Fun.swap a b id o p*) ' {(*b*,*p*). *b* ∈ *insert a S* ∧ *p* ∈ {*p*. *p permutes S*}}
⟨*proof*⟩

lemma *card-permutations*: **assumes** *Sn*: *card S* = *n* **and** *fS*: *finite S*
shows *card* {*p*. *p permutes S*} = *fact n*
⟨*proof*⟩

lemma *finite-permutations*: **assumes** *fS*: *finite S* **shows** *finite* {*p*. *p permutes S*}
⟨*proof*⟩

lemma (**in** *ab-semigroup-mult*) *fold-image-permute*: **assumes** *fS*: *finite S* **and** *pS*: *p permutes S*
shows *fold-image times f z S* = *fold-image times* (*f o p*) *z S*
⟨*proof*⟩

lemma (**in** *ab-semigroup-add*) *fold-image-permute*: **assumes** *fS*: *finite S* **and** *pS*: *p permutes S*
shows *fold-image plus f z S* = *fold-image plus* (*f o p*) *z S*
⟨*proof*⟩

lemma *setsum-permute*: **assumes** *pS*: *p permutes S*
shows *setsum f S = setsum (f o p) S*
⟨*proof*⟩

lemma *setsum-permute-natseg*:**assumes** *pS*: *p permutes* $\{m .. n\}$
shows *setsum f* $\{m .. n\}$ = *setsum (f o p)* $\{m .. n\}$
⟨*proof*⟩

lemma *setprod-permute*: **assumes** *pS*: *p permutes S*
shows *setprod f S = setprod (f o p) S*
⟨*proof*⟩

lemma *setprod-permute-natseg*:**assumes** *pS*: *p permutes* $\{m .. n\}$
shows *setprod f* $\{m .. n\}$ = *setprod (f o p)* $\{m .. n\}$
⟨*proof*⟩

lemma *swap-id-common*: $a \neq c \Longrightarrow b \neq c \Longrightarrow$ *Fun.swap a b id o Fun.swap a c id = Fun.swap b c id o Fun.swap a b id* ⟨*proof*⟩

lemma *swap-id-common'*: $\sim(a = b) \Longrightarrow \sim(a = c) \Longrightarrow$ *Fun.swap a c id o Fun.swap b c id = Fun.swap b c id o Fun.swap a b id* ⟨*proof*⟩

lemma *swap-id-independent*: $\sim(a = c) \Longrightarrow \sim(a = d) \Longrightarrow \sim(b = c) \Longrightarrow \sim(b = d)$ ==> *Fun.swap a b id o Fun.swap c d id = Fun.swap c d id o Fun.swap a b id*
⟨*proof*⟩

inductive *swapidseq* :: *nat* ⇒ (*'a* ⇒ *'a*) ⇒ *bool* **where**
id[*simp*]: *swapidseq 0 id*
| *comp-Suc*: *swapidseq n p* $\Longrightarrow a \neq b \Longrightarrow$ *swapidseq (Suc n) (Fun.swap a b id o p)*

declare *id*[*unfolded id-def*, *simp*]
definition *permutation p* ⟷ (∃ *n. swapidseq n p*)

lemma *permutation-id*[*simp*]: *permutation id*⟨*proof*⟩
declare *permutation-id*[*unfolded id-def*, *simp*]

lemma *swapidseq-swap*: *swapidseq* (*if* $a = b$ *then 0 else 1*) (*Fun.swap a b id*)
⟨*proof*⟩

lemma *permutation-swap-id*: *permutation* (*Fun.swap a b id*)
⟨*proof*⟩

lemma *swapidseq-comp-add*: *swapidseq n p* $\Longrightarrow$ *swapidseq m q* ==> *swapidseq* ($n + m$) (*p o q*)
⟨*proof*⟩

lemma *permutation-compose*: *permutation p* $\Longrightarrow$ *permutation q* ==> *permutation*(*p o q*)
⟨*proof*⟩

lemma *swapidseq-endswap*: *swapidseq n p* $\Longrightarrow$ $a \neq b$ ==> *swapidseq* (*Suc n*) (*p o Fun.swap a b id*)
⟨*proof*⟩

lemma *swapidseq-inverse-exists*: *swapidseq n p* ==> $\exists\, q$. *swapidseq n q* $\wedge$ *p o q* $=$ *id* $\wedge$ *q o p* $=$ *id*
⟨*proof*⟩

lemma *swapidseq-inverse*: **assumes** *H*: *swapidseq n p* **shows** *swapidseq n* (*inv p*)
⟨*proof*⟩

lemma *permutation-inverse*: *permutation p* ==> *permutation* (*inv p*)
⟨*proof*⟩

lemma *symmetry-lemma*:($\bigwedge$*a b c d*. *P a b c d* ==> *P a b d c*) $\Longrightarrow$
($\bigwedge$*a b c d*. $a \neq b \Longrightarrow c \neq d \Longrightarrow (a = c \wedge b = d \vee a = c \wedge b \neq d \vee a \neq c \wedge b = d \vee a \neq c \wedge a \neq d \wedge b \neq c \wedge b \neq d)$ ==> *P a b c d*)
==> ($\bigwedge$*a b c d*. $a \neq b$ --> $c \neq d \longrightarrow$ *P a b c d*) ⟨*proof*⟩

lemma *swap-general*: $a \neq b \Longrightarrow c \neq d \Longrightarrow$ *Fun.swap a b id o Fun.swap c d id* $=$ *id* $\vee$
($\exists\, x\, y\, z$. $x \neq a \wedge y \neq a \wedge z \neq a \wedge x \neq y \wedge$ *Fun.swap a b id o Fun.swap c d id* $=$ *Fun.swap x y id o Fun.swap a z id*)
⟨*proof*⟩

lemma *swapidseq-id-iff* [*simp*]: *swapidseq 0 p* $\longleftrightarrow$ $p = id$
⟨*proof*⟩

lemma *swapidseq-cases*: *swapidseq n p* $\longleftrightarrow$ (n=*0* $\wedge$ $p = id$ $\vee$ ($\exists\, a\, b\, q\, m$. $n =$ *Suc*

$m \wedge p = \mathit{Fun.swap}\ a\ b\ \mathit{id}\ o\ q \wedge \mathit{swapidseq}\ m\ q \wedge a \neq b))$
⟨*proof*⟩
lemma *fixing-swapidseq-decrease*:
assumes *spn*: $\mathit{swapidseq}\ n\ p$ **and** *ab*: $a \neq b$ **and** *pa*: $(\mathit{Fun.swap}\ a\ b\ \mathit{id}\ o\ p)\ a = a$
shows $n \neq 0 \wedge \mathit{swapidseq}\ (n - 1)\ (\mathit{Fun.swap}\ a\ b\ \mathit{id}\ o\ p)$
⟨*proof*⟩

lemma *swapidseq-identity-even*:
assumes $\mathit{swapidseq}\ n\ (\mathit{id} :: 'a \Rightarrow 'a)$ **shows** $\mathit{even}\ n$
⟨*proof*⟩

definition $\mathit{evenperm}\ p = \mathit{even}\ (\mathit{SOME}\ n.\ \mathit{swapidseq}\ n\ p)$

lemma *swapidseq-even-even*: **assumes**
m: $\mathit{swapidseq}\ m\ p$ **and** *n*: $\mathit{swapidseq}\ n\ p$
shows $\mathit{even}\ m \longleftrightarrow \mathit{even}\ n$
⟨*proof*⟩

lemma *evenperm-unique*: **assumes** *p*: $\mathit{swapidseq}\ n\ p$ **and** *n*:$\mathit{even}\ n = b$
shows $\mathit{evenperm}\ p = b$
⟨*proof*⟩

lemma *evenperm-id*[*simp*]: $\mathit{evenperm}\ \mathit{id} = \mathit{True}$
⟨*proof*⟩

lemma *evenperm-swap*: $\mathit{evenperm}\ (\mathit{Fun.swap}\ a\ b\ \mathit{id}) = (a = b)$
⟨*proof*⟩

lemma *evenperm-comp*:
assumes *p*: $\mathit{permutation}\ p$ **and** *q*:$\mathit{permutation}\ q$
shows $\mathit{evenperm}\ (p\ o\ q) = (\mathit{evenperm}\ p = \mathit{evenperm}\ q)$
⟨*proof*⟩

lemma *evenperm-inv*: **assumes** *p*: $\mathit{permutation}\ p$
shows $\mathit{evenperm}\ (\mathit{inv}\ p) = \mathit{evenperm}\ p$
⟨*proof*⟩

lemma *bij-iff*: $bij\ f \longleftrightarrow (\forall\, x.\ \exists!y.\ f\ y = x)$
⟨*proof*⟩

lemma *permutation-bijective*:
assumes *p*: *permutation p*
shows *bij p*
⟨*proof*⟩

lemma *permutation-finite-support*: **assumes** *p*: *permutation p*
shows $\mathit{finite}\ \{x.\ p\ x \neq x\}$
⟨*proof*⟩

lemma *bij-inv-eq-iff*: $bij\ p ==> x = inv\ p\ y \longleftrightarrow p\ x = y$
⟨*proof*⟩

lemma *bij-swap-comp*:
assumes *bp*: *bij p* **shows** $Fun.swap\ a\ b\ id\ o\ p = Fun.swap\ (inv\ p\ a)\ (inv\ p\ b)\ p$
⟨*proof*⟩

lemma *bij-swap-ompose-bij*: $bij\ p \Longrightarrow bij\ (Fun.swap\ a\ b\ id\ o\ p)$
⟨*proof*⟩

lemma *permutation-lemma*:
assumes *fS*: *finite S* **and** *p*: *bij p* **and** *pS*: $\forall\, x.\ x \notin S \longrightarrow p\ x = x$
shows *permutation p*
⟨*proof*⟩

lemma *permutation*: $permutation\ p \longleftrightarrow bij\ p \wedge \mathit{finite}\ \{x.\ p\ x \neq x\}$
(**is** $?lhs \longleftrightarrow ?b \wedge ?f$)
⟨*proof*⟩

lemma *permutation-inverse-works*: **assumes** *p*: *permutation p*
shows $inv\ p\ o\ p = id\ p\ o\ inv\ p = id$
⟨*proof*⟩

lemma *permutation-inverse-compose*:
assumes *p*: *permutation p* **and** *q*: *permutation q*
shows $inv\ (p\ o\ q) = inv\ q\ o\ inv\ p$
⟨*proof*⟩

lemma *permutation-permutes*: $permutation\ p \longleftrightarrow (\exists\, S.\ \mathit{finite}\ S \wedge p\ permutes\ S)$
⟨*proof*⟩

lemma *permutes-induct*: *finite* $S \Longrightarrow$ $P\ id$ $\Longrightarrow (\bigwedge a\ b\ p.\ a \in S \Longrightarrow b \in S \Longrightarrow P\ p \Longrightarrow P\ p \Longrightarrow permutation\ p ==> P\ (Fun.swap\ a\ b\ id\ o\ p))$
$==> (\bigwedge p.\ p\ permutes\ S ==> P\ p)$
$\langle proof \rangle$

definition $sign\ p = (if\ evenperm\ p\ then\ (1::int)\ else\ -1)$

lemma *sign-nz*: $sign\ p \neq 0$ $\langle proof \rangle$
lemma *sign-id*: $sign\ id = 1$ $\langle proof \rangle$
lemma *sign-inverse*: $permutation\ p ==> sign\ (inv\ p) = sign\ p$
$\langle proof \rangle$
lemma *sign-compose*: $permutation\ p \Longrightarrow permutation\ q ==> sign\ (p\ o\ q) = sign(p) * sign(q)$ $\langle proof \rangle$
lemma *sign-swap-id*: $sign\ (Fun.swap\ a\ b\ id) = (if\ a = b\ then\ 1\ else\ -1)$
$\langle proof \rangle$
lemma *sign-idempotent*: $sign\ p * sign\ p = 1$ $\langle proof \rangle$

lemma *permutes-natset-le*:
assumes p: $p\ permutes\ (S::'a::wellorder\ set)$ **and** le: $\forall\, i \in S.\ \ p\ i <= i$ **shows** $p = id$
$\langle proof \rangle$

lemma *permutes-natset-ge*:
assumes p: $p\ permutes\ (S::'a::wellorder\ set)$ **and** le: $\forall\, i \in S.\ \ p\ i \geq i$ **shows** $p = id$
$\langle proof \rangle$

lemma *image-inverse-permutations*: $\{inv\ p\ |p.\ p\ permutes\ S\} = \{p.\ p\ permutes\ S\}$
$\langle proof \rangle$

lemma *image-compose-permutations-left*:
assumes q: $q\ permutes\ S$ **shows** $\{q\ o\ p\ |\ p.\ p\ permutes\ S\} = \{p\ .\ p\ permutes\ S\}$
$\langle proof \rangle$

lemma *image-compose-permutations-right*:
assumes q: $q\ permutes\ S$

shows $\{p\ o\ q \mid p.\ p\ permutes\ S\} = \{p\ .\ p\ permutes\ S\}$
⟨*proof*⟩

lemma *permutes-in-seg*: $p\ permutes\ \{1\ ..n\} \Longrightarrow i \in \{1..n\} ==> 1 <= p\ i \wedge p\ i <= n$

⟨*proof*⟩

term *setsum*
lemma *setsum-permutations-inverse*: $setsum\ f\ \{p.\ p\ permutes\ S\} = setsum\ (\lambda p.\ f(inv\ p))\ \{p.\ p\ permutes\ S\}$ (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *setum-permutations-compose-left*:
assumes *q*: *q permutes S*
shows $setsum\ f\ \{p.\ p\ permutes\ S\} =$
$setsum\ (\lambda p.\ f(q\ o\ p))\ \{p.\ p\ permutes\ S\}$ (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *sum-permutations-compose-right*:
assumes *q*: *q permutes S*
shows $setsum\ f\ \{p.\ p\ permutes\ S\} =$
$setsum\ (\lambda p.\ f(p\ o\ q))\ \{p.\ p\ permutes\ S\}$ (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *setsum-over-permutations-insert*:
assumes *fS*: *finite S* **and** *aS*: $a \notin S$
shows $setsum\ f\ \{p.\ p\ permutes\ (insert\ a\ S)\} = setsum\ (\lambda b.\ setsum\ (\lambda q.\ f\ (Fun.swap\ a\ b\ id\ o\ q))\ \{p.\ p\ permutes\ S\})\ (insert\ a\ S)$
⟨*proof*⟩

end

15 Glbs: Definitions of Lower Bounds and Greatest Lower Bounds, analogous to Lubs

theory *Glbs*
imports *Lubs*
begin

definition
greatestP :: $['a =>bool,'a::ord] => bool$ **where**
greatestP P x = $(P\ x\ \&\ Collect\ P *<=\ x)$

definition
isLb :: ['a set, 'a set, 'a::ord] => *bool* **where**
isLb R S x = (*x* <=* *S* & *x*: *R*)

definition
isGlb :: ['a set, 'a set, 'a::ord] => *bool* **where**
isGlb R S x = *greatestP* (*isLb R S*) *x*

definition
lbs :: ['a set, 'a::ord set] => 'a set **where**
lbs R S = *Collect* (*isLb R S*)

15.1 Rules about the Operators *greatestP*, *isLb* and *isGlb*

lemma *leastPD1*: *greatestP P x* ==> *P x*
⟨*proof*⟩

lemma *greatestPD2*: *greatestP P x* ==> *Collect P* *<= *x*
⟨*proof*⟩

lemma *greatestPD3*: [| *greatestP P x*; *y*: *Collect P* |] ==> *x* >= *y*
⟨*proof*⟩

lemma *isGlbD1*: *isGlb R S x* ==> *x* <=* *S*
⟨*proof*⟩

lemma *isGlbD1a*: *isGlb R S x* ==> *x*: *R*
⟨*proof*⟩

lemma *isGlb-isLb*: *isGlb R S x* ==> *isLb R S x*
⟨*proof*⟩

lemma *isGlbD2*: [| *isGlb R S x*; *y* : *S* |] ==> *y* >= *x*
⟨*proof*⟩

lemma *isGlbD3*: *isGlb R S x* ==> *greatestP*(*isLb R S*) *x*
⟨*proof*⟩

lemma *isGlbI1*: *greatestP*(*isLb R S*) *x* ==> *isGlb R S x*
⟨*proof*⟩

lemma *isGlbI2*: [| *isLb R S x*; *Collect* (*isLb R S*) *<= *x* |] ==> *isGlb R S x*
⟨*proof*⟩

lemma *isLbD*: [| *isLb R S x*; *y* : *S* |] ==> *y* >= *x*
⟨*proof*⟩

lemma *isLbD2*: *isLb R S x* ==> *x* <=* *S*

⟨*proof*⟩

lemma *isLbD2a*: *isLb R S x ==> x*: *R*
⟨*proof*⟩

lemma *isLbI*: [| *x <=* S* ; *x*: *R* |] ==> *isLb R S x*
⟨*proof*⟩

lemma *isGlb-le-isLb*: [| *isGlb R S x*; *isLb R S y* |] ==> *x >= y*
⟨*proof*⟩

lemma *isGlb-ubs*: *isGlb R S x ==> lbs R S *<= x*
⟨*proof*⟩

end

16 Topology-Euclidean-Space: Elementary topology in Euclidean space.

theory *Topology-Euclidean-Space*
imports *SEQ Euclidean-Space Glbs*
begin

16.1 General notion of a topology

definition *istopology* $L \longleftrightarrow \{\} \in L \land (\forall S \in L.\ \forall T \in L.\ S \cap T \in L) \land (\forall K.\ K \subseteq L \longrightarrow \bigcup K \in L)$
typedef (**open**) *'a topology* = $\{L::('a\ set)\ set.\ istopology\ L\}$
  morphisms *openin topology*
  ⟨*proof*⟩

lemma *istopology-open-in*[*intro*]: *istopology*(*openin U*)
  ⟨*proof*⟩

lemma *topology-inverse'*: *istopology U* $\Longrightarrow$ *openin* (*topology U*) = *U*
  ⟨*proof*⟩

lemma *topology-inverse-iff*: *istopology U* $\longleftrightarrow$ *openin* (*topology U*) = *U*
  ⟨*proof*⟩

lemma *topology-eq*: $T1 = T2 \longleftrightarrow (\forall S.\ openin\ T1\ S \longleftrightarrow openin\ T2\ S)$
⟨*proof*⟩

Infer the "universe" from union of all sets in the topology.

definition *topspace* $T = \bigcup\{S.\ openin\ T\ S\}$

16.2 Main properties of open sets

lemma *openin-clauses*:
fixes $U :: 'a\ topology$
shows $openin\ U\ \{\}$
$\bigwedge S\ T.\ openin\ U\ S \Longrightarrow openin\ U\ T \Longrightarrow openin\ U\ (S \cap T)$
$\bigwedge K.\ (\forall S \in K.\ openin\ U\ S) \Longrightarrow openin\ U\ (\bigcup K)$
$\langle proof \rangle$

lemma *openin-subset*[*intro*]: $openin\ U\ S \Longrightarrow S \subseteq topspace\ U$
$\langle proof \rangle$
lemma *openin-empty*[*simp*]: $openin\ U\ \{\}$ $\langle proof \rangle$

lemma *openin-Int*[*intro*]: $openin\ U\ S \Longrightarrow openin\ U\ T \Longrightarrow openin\ U\ (S \cap T)$
$\langle proof \rangle$

lemma *openin-Union*[*intro*]: $(\forall S \in K.\ openin\ U\ S) \Longrightarrow openin\ U\ (\bigcup\ K)$
$\langle proof \rangle$

lemma *openin-Un*[*intro*]: $openin\ U\ S \Longrightarrow openin\ U\ T \Longrightarrow openin\ U\ (S \cup T)$
$\langle proof \rangle$

lemma *openin-topspace*[*intro*, *simp*]: $openin\ U\ (topspace\ U)$ $\langle proof \rangle$

lemma *openin-subopen*: $openin\ U\ S \longleftrightarrow (\forall x \in S.\ \exists T.\ openin\ U\ T \wedge x \in T \wedge T \subseteq S)$ (**is** $?lhs \longleftrightarrow ?rhs$)
$\langle proof \rangle$

16.3 Closed sets

definition $closedin\ U\ S \longleftrightarrow S \subseteq topspace\ U \wedge openin\ U\ (topspace\ U - S)$

lemma *closedin-subset*: $closedin\ U\ S \Longrightarrow S \subseteq topspace\ U$ $\langle proof \rangle$
lemma *closedin-empty*[*simp*]: $closedin\ U\ \{\}$ $\langle proof \rangle$
lemma *closedin-topspace*[*intro*,*simp*]:
$closedin\ U\ (topspace\ U)$ $\langle proof \rangle$
lemma *closedin-Un*[*intro*]: $closedin\ U\ S \Longrightarrow closedin\ U\ T \Longrightarrow closedin\ U\ (S \cup T)$
$\langle proof \rangle$

lemma *Diff-Inter*[*intro*]: $A - \bigcap S = \bigcup\ \{A - s | s.\ s \in S\}$ $\langle proof \rangle$
lemma *closedin-Inter*[*intro*]: **assumes** Ke: $K \neq \{\}$ **and** Kc: $\forall S \in K.\ closedin\ U\ S$
shows $closedin\ U\ (\bigcap\ K)$ $\langle proof \rangle$

lemma *closedin-Int*[*intro*]: $closedin\ U\ S \Longrightarrow closedin\ U\ T \Longrightarrow closedin\ U\ (S \cap T)$
$\langle proof \rangle$

lemma *Diff-Diff-Int*: $A - (A - B) = A \cap B$ $\langle proof \rangle$

lemma *openin-closedin-eq*: $openin\ U\ S \longleftrightarrow S \subseteq topspace\ U \wedge closedin\ U\ (topspace\ U - S)$
⟨*proof*⟩

lemma *openin-closedin*: $S \subseteq topspace\ U \Longrightarrow (openin\ U\ S \longleftrightarrow closedin\ U\ (topspace\ U - S))$
⟨*proof*⟩

lemma *openin-diff* [*intro*]: **assumes** oS: $openin\ U\ S$ **and** cT: $closedin\ U\ T$ **shows** $openin\ U\ (S - T)$
⟨*proof*⟩

lemma *closedin-diff* [*intro*]: **assumes** oS: $closedin\ U\ S$ **and** cT: $openin\ U\ T$ **shows** $closedin\ U\ (S - T)$
⟨*proof*⟩

16.4 Subspace topology.

definition $subtopology\ U\ V = topology\ \{S \cap V\ |S.\ openin\ U\ S\}$

lemma *istopology-subtopology*: $istopology\ \{S \cap V\ |S.\ openin\ U\ S\}$ (**is** *istopology ?L*)
⟨*proof*⟩

lemma *openin-subtopology*:
$openin\ (subtopology\ U\ V)\ S \longleftrightarrow (\exists\ T.\ (openin\ U\ T) \wedge (S = T \cap V))$
⟨*proof*⟩

lemma *topspace-subtopology*: $topspace(subtopology\ U\ V) = topspace\ U \cap V$
⟨*proof*⟩

lemma *closedin-subtopology*:
$closedin\ (subtopology\ U\ V)\ S \longleftrightarrow (\exists\ T.\ closedin\ U\ T \wedge S = T \cap V)$
⟨*proof*⟩

lemma *openin-subtopology-refl*: $openin\ (subtopology\ U\ V)\ V \longleftrightarrow V \subseteq topspace\ U$
⟨*proof*⟩

lemma *subtopology-superset*: **assumes** UV: $topspace\ U \subseteq V$
shows $subtopology\ U\ V = U$
⟨*proof*⟩

lemma *subtopology-topspace*[*simp*]: $subtopology\ U\ (topspace\ U) = U$
⟨*proof*⟩

lemma *subtopology-UNIV*[*simp*]: $subtopology\ U\ UNIV = U$
⟨*proof*⟩

16.5 The universal Euclidean versions are what we use most of the time

definition
euclidean :: *'a::topological-space topology* **where**
euclidean = *topology open*

lemma *open-openin*: *open S* $\longleftrightarrow$ *openin euclidean S*
⟨*proof*⟩

lemma *topspace-euclidean*: *topspace euclidean* = *UNIV*
⟨*proof*⟩

lemma *topspace-euclidean-subtopology*[*simp*]: *topspace* (*subtopology euclidean S*) = *S*
⟨*proof*⟩

lemma *closed-closedin*: *closed S* $\longleftrightarrow$ *closedin euclidean S*
⟨*proof*⟩

lemma *open-subopen*: *open S* $\longleftrightarrow$ ($\forall x \in S.\ \exists T.$ *open* $T \wedge x \in T \wedge T \subseteq S$)
⟨*proof*⟩

16.6 Open and closed balls.

definition
ball :: *'a::metric-space* $\Rightarrow$ *real* $\Rightarrow$ *'a set* **where**
ball x e = $\{y.$ *dist x y* $< e\}$

definition
cball :: *'a::metric-space* $\Rightarrow$ *real* $\Rightarrow$ *'a set* **where**
cball x e = $\{y.$ *dist x y* $\leq e\}$

lemma *mem-ball*[*simp*]: $y \in$ *ball x e* $\longleftrightarrow$ *dist x y* $< e$ ⟨*proof*⟩
lemma *mem-cball*[*simp*]: $y \in$ *cball x e* $\longleftrightarrow$ *dist x y* $\leq e$ ⟨*proof*⟩

lemma *mem-ball-0* [*simp*]:
fixes *x* :: *'a::real-normed-vector*
shows $x \in$ *ball 0 e* $\longleftrightarrow$ *norm x* $< e$
⟨*proof*⟩

lemma *mem-cball-0* [*simp*]:
fixes *x* :: *'a::real-normed-vector*
shows $x \in$ *cball 0 e* $\longleftrightarrow$ *norm x* $\leq e$
⟨*proof*⟩

lemma *centre-in-cball*[*simp*]: $x \in$ *cball x e* $\longleftrightarrow$ $0 \leq e$ ⟨*proof*⟩
lemma *ball-subset-cball*[*simp,intro*]: *ball x e* $\subseteq$ *cball x e* ⟨*proof*⟩
lemma *subset-ball*[*intro*]: *d* <= *e* ==> *ball x d* $\subseteq$ *ball x e* ⟨*proof*⟩
lemma *subset-cball*[*intro*]: *d* <= *e* ==> *cball x d* $\subseteq$ *cball x e* ⟨*proof*⟩

lemma *ball-max-Un*: $ball\ a\ (max\ r\ s) = ball\ a\ r \cup ball\ a\ s$
⟨*proof*⟩

lemma *ball-min-Int*: $ball\ a\ (min\ r\ s) = ball\ a\ r \cap ball\ a\ s$
⟨*proof*⟩

lemma *diff-less-iff*: $(a::real) - b > 0 \longleftrightarrow a > b$
$(a::real) - b < 0 \longleftrightarrow a < b$
$a - b < c \longleftrightarrow a < c + b$ $a - b > c \longleftrightarrow a > c + b$ ⟨*proof*⟩
lemma *diff-le-iff*: $(a::real) - b \geq 0 \longleftrightarrow a \geq b$ $(a::real) - b \leq 0 \longleftrightarrow a \leq b$
$a - b \leq c \longleftrightarrow a \leq c + b$ $a - b \geq c \longleftrightarrow a \geq c + b$ ⟨*proof*⟩

lemma *open-ball*[*intro*, *simp*]: $open\ (ball\ x\ e)$
⟨*proof*⟩

lemma *centre-in-ball*[*simp*]: $x \in ball\ x\ e \longleftrightarrow e > 0$ ⟨*proof*⟩
lemma *open-contains-ball*: $open\ S \longleftrightarrow (\forall x{\in}S.\ \exists e{>}0.\ ball\ x\ e \subseteq S)$
⟨*proof*⟩

lemma *openE*[*elim?*]:
assumes $open\ S$ $x{\in}S$
obtains e **where** $e{>}0$ $ball\ x\ e \subseteq S$
⟨*proof*⟩

lemma *open-contains-ball-eq*: $open\ S \Longrightarrow \forall x.\ x{\in}S \longleftrightarrow (\exists e{>}0.\ ball\ x\ e \subseteq S)$
⟨*proof*⟩

lemma *ball-eq-empty*[*simp*]: $ball\ x\ e = \{\} \longleftrightarrow e \leq 0$
⟨*proof*⟩

lemma *ball-empty*[*intro*]: $e \leq 0 ==> ball\ x\ e = \{\}$ ⟨*proof*⟩

16.7 Basic "localization" results are handy for connectedness.

lemma *openin-open*: $openin\ (subtopology\ euclidean\ U)\ S \longleftrightarrow (\exists T.\ open\ T \wedge (S = U \cap T))$
⟨*proof*⟩

lemma *openin-open-Int*[*intro*]: $open\ S \Longrightarrow openin\ (subtopology\ euclidean\ U)\ (U \cap S)$
⟨*proof*⟩

lemma *open-openin-trans*[*trans*]:
$open\ S \Longrightarrow open\ T \Longrightarrow T \subseteq S \Longrightarrow openin\ (subtopology\ euclidean\ S)\ T$
⟨*proof*⟩

lemma *open-subset*: $S \subseteq T \Longrightarrow open\ S \Longrightarrow openin\ (subtopology\ euclidean\ T)\ S$
⟨*proof*⟩

lemma *closedin-closed*: *closedin* (*subtopology euclidean* U) $S \longleftrightarrow (\exists\, T.\ closed\ T \wedge S = U \cap T)$
⟨*proof*⟩

lemma *closedin-closed-Int*: *closed* S ==> *closedin* (*subtopology euclidean* U) ($U \cap S$)
⟨*proof*⟩

lemma *closed-closedin-trans*: *closed* $S \Longrightarrow$ *closed* $T \Longrightarrow T \subseteq S \Longrightarrow$ *closedin* (*subtopology euclidean* S) T
⟨*proof*⟩

lemma *closed-subset*: $S \subseteq T \Longrightarrow$ *closed* $S \Longrightarrow$ *closedin* (*subtopology euclidean* T) S
⟨*proof*⟩

lemma *openin-euclidean-subtopology-iff*:
fixes S U :: *'a::metric-space set*
shows *openin* (*subtopology euclidean* U) S
$\longleftrightarrow S \subseteq U \wedge (\forall x{\in}S.\ \exists e{>}0.\ \forall x'{\in}U.\ dist\ x'\ x < e \longrightarrow x'{\in}\ S)$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

These "transitivity" results are handy too.

lemma *openin-trans*[*trans*]: *openin* (*subtopology euclidean* T) $S \Longrightarrow$ *openin* (*subtopology euclidean* U) T
$\Longrightarrow$ *openin* (*subtopology euclidean* U) S
⟨*proof*⟩

lemma *openin-open-trans*: *openin* (*subtopology euclidean* T) $S \Longrightarrow$ *open* $T \Longrightarrow$ *open* S
⟨*proof*⟩

lemma *closedin-trans*[*trans*]:
closedin (*subtopology euclidean* T) $S \Longrightarrow$
closedin (*subtopology euclidean* U) T
==> *closedin* (*subtopology euclidean* U) S
⟨*proof*⟩

lemma *closedin-closed-trans*: *closedin* (*subtopology euclidean* T) $S \Longrightarrow$ *closed* $T \Longrightarrow$ *closed* S
⟨*proof*⟩

16.8 Connectedness

definition *connected* $S \longleftrightarrow$
$\sim(\exists\, e1\ e2.\ open\ e1 \wedge open\ e2 \wedge S \subseteq (e1 \cup e2) \wedge (e1 \cap e2 \cap S = \{\})$
$\wedge \sim(e1 \cap S = \{\}) \wedge \sim(e2 \cap S = \{\}))$

lemma *connected-local*:
connected $S \longleftrightarrow {}^{\sim}(\exists\, e1\ e2.$
$\quad openin\ (subtopology\ euclidean\ S)\ e1\ \wedge$
$\quad openin\ (subtopology\ euclidean\ S)\ e2\ \wedge$
$\quad S \subseteq e1 \cup e2\ \wedge$
$\quad e1 \cap e2 = \{\}\ \wedge$
$\quad {}^{\sim}(e1 = \{\})\ \wedge$
$\quad {}^{\sim}(e2 = \{\}))$
⟨*proof*⟩

lemma *exists-diff*:
fixes $P :: {}'a\ set \Rightarrow bool$
shows $(\exists\, S.\ P(-\ S)) \longleftrightarrow (\exists\, S.\ P\ S)$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

lemma *connected-clopen*: *connected* $S \longleftrightarrow$
$(\forall\, T.\ openin\ (subtopology\ euclidean\ S)\ T\ \wedge$
$closedin\ (subtopology\ euclidean\ S)\ T \longrightarrow T = \{\} \vee T = S)$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

lemma *connected-empty*[*simp*, *intro*]: *connected* {}
⟨*proof*⟩

16.9 Hausdorff and other separation properties

class *t0-space* = *topological-space* +
assumes *t0-space*: $x \neq y \Longrightarrow \exists\, U.\ open\ U \wedge \neg\ (x \in U \longleftrightarrow y \in U)$

class *t1-space* = *topological-space* +
assumes *t1-space*: $x \neq y \Longrightarrow \exists\, U.\ open\ U \wedge x \in U \wedge y \notin U$

instance *t1-space* $\subseteq$ *t0-space*
⟨*proof*⟩

lemma *separation-t1*:
fixes $x\ y :: {}'a{::}t1\text{-}space$
shows $x \neq y \longleftrightarrow (\exists\, U.\ open\ U \wedge x \in U \wedge y \notin U)$
⟨*proof*⟩

lemma *closed-sing*:
fixes $a :: {}'a{::}t1\text{-}space$
shows *closed* $\{a\}$
⟨*proof*⟩

lemma *closed-insert* [*simp*]:
fixes $a :: {}'a{::}t1\text{-}space$
assumes *closed* S **shows** *closed* (*insert* a S)

⟨*proof*⟩

lemma *finite-imp-closed*:
fixes S :: *'a::t1-space set*
shows *finite* $S \Longrightarrow$ *closed* S
⟨*proof*⟩

T2 spaces are also known as Hausdorff spaces.

class *t2-space* = *topological-space* +
assumes *hausdorff*: $x \neq y \Longrightarrow \exists\, U\ V.\ open\ U \wedge open\ V \wedge x \in U \wedge y \in V \wedge U \cap V = \{\}$

instance *t2-space* ⊆ *t1-space*
⟨*proof*⟩

instance *metric-space* ⊆ *t2-space*
⟨*proof*⟩

lemma *separation-t2*:
fixes $x\ y$ :: *'a::t2-space*
shows $x \neq y \longleftrightarrow (\exists\, U\ V.\ open\ U \wedge open\ V \wedge x \in U \wedge y \in V \wedge U \cap V = \{\})$
⟨*proof*⟩

lemma *separation-t0*:
fixes $x\ y$ :: *'a::t0-space*
shows $x \neq y \longleftrightarrow (\exists\, U.\ open\ U \wedge \sim(x{\in}U \longleftrightarrow y{\in}U))$
⟨*proof*⟩

16.10 Limit points

definition
islimpt:: *'a::topological-space* ⇒ *'a set* ⇒ *bool*
(**infixr** *islimpt 60*) **where**
x *islimpt* $S \longleftrightarrow (\forall\, T.\ x{\in}T \longrightarrow open\ T \longrightarrow (\exists\, y{\in}S.\ y{\in}T \wedge y{\neq}x))$

lemma *islimptI*:
assumes $\bigwedge T.\ x \in T \Longrightarrow open\ T \Longrightarrow \exists\, y{\in}S.\ y \in T \wedge y \neq x$
shows x *islimpt* S
⟨*proof*⟩

lemma *islimptE*:
assumes x *islimpt* S **and** $x \in T$ **and** *open* T
obtains y **where** $y \in S$ **and** $y \in T$ **and** $y \neq x$
⟨*proof*⟩

lemma *islimpt-subset*: x *islimpt* $S \Longrightarrow S \subseteq T$ ==> x *islimpt* T ⟨*proof*⟩

lemma *islimpt-approachable*:
fixes x :: *'a::metric-space*

shows x *islimpt* $S \longleftrightarrow (\forall\, e{>}0.\ \exists\, x'{\in}S.\ x' \neq x \wedge dist\ x'\ x < e)$
⟨*proof*⟩

lemma *islimpt-approachable-le*:
fixes x :: *'a::metric-space*
shows x *islimpt* $S \longleftrightarrow (\forall\, e{>}0.\ \exists\, x'{\in}\ S.\ x' \neq x \wedge dist\ x'\ x <= e)$
⟨*proof*⟩

class *perfect-space* =

assumes *islimpt-UNIV* [*simp*, *intro*]: (*x::'a::metric-space*) *islimpt UNIV*

lemma *perfect-choose-dist*:
fixes x :: *'a::perfect-space*
shows $0 < r \Longrightarrow \exists\, a.\ a \neq x \wedge dist\ a\ x < r$
⟨*proof*⟩

instance *real* :: *perfect-space*
⟨*proof*⟩

instance *cart* :: (*perfect-space*, *finite*) *perfect-space*
⟨*proof*⟩

lemma *closed-limpt*: *closed* $S \longleftrightarrow (\forall\, x.\ x$ *islimpt* $S \longrightarrow x \in S)$
⟨*proof*⟩

lemma *islimpt-EMPTY* [*simp*]: $\neg\ x$ *islimpt* $\{\}$
⟨*proof*⟩

lemma *closed-positive-orthant*: *closed* $\{x{::}real\hat{\ }'n.\ \forall\, i.\ 0 \le x\$i\}$
⟨*proof*⟩

lemma *finite-set-avoid*:
fixes a :: *'a::metric-space*
assumes *fS*: *finite* S **shows** $\exists\, d{>}0.\ \forall\, x{\in}S.\ x \neq a \longrightarrow d <= dist\ a\ x$
⟨*proof*⟩

lemma *islimpt-finite*:
fixes S :: *'a::metric-space set*
assumes *fS*: *finite* S **shows** $\neg\ a$ *islimpt* S
⟨*proof*⟩

lemma *islimpt-Un*: x *islimpt* $(S \cup T) \longleftrightarrow x$ *islimpt* $S \vee x$ *islimpt* T
⟨*proof*⟩

lemma *discrete-imp-closed*:
fixes S :: *'a::metric-space set*
assumes e: $0 < e$ **and** d: $\forall\, x \in S.\ \forall\, y \in S.\ dist\ y\ x < e \longrightarrow y = x$
shows *closed* S

⟨*proof*⟩

16.11 Interior of a Set

definition *interior* $S = \{x.\ \exists\, T.\ open\ T \wedge x \in T \wedge T \subseteq S\}$

lemma *interior-eq*: *interior* $S = S \longleftrightarrow open\ S$
⟨*proof*⟩

lemma *interior-open*: *open* S ==> (*interior* $S = S$) ⟨*proof*⟩

lemma *interior-empty*[*simp*]: *interior* $\{\} = \{\}$ ⟨*proof*⟩

lemma *open-interior*[*simp*, *intro*]: *open*(*interior* S)
⟨*proof*⟩

lemma *interior-interior*[*simp*]: *interior*(*interior* S) = *interior* S ⟨*proof*⟩
lemma *interior-subset*: *interior* $S \subseteq S$ ⟨*proof*⟩
lemma *subset-interior*: $S \subseteq T$ ==> (*interior* S) $\subseteq$ (*interior* T) ⟨*proof*⟩
lemma *interior-maximal*: $T \subseteq S \Longrightarrow open\ T$ ==> $T \subseteq$ (*interior* S) ⟨*proof*⟩
lemma *interior-unique*: $T \subseteq S \Longrightarrow open\ T \Longrightarrow (\forall\, T'.\ T' \subseteq S \wedge open\ T' \longrightarrow T' \subseteq T) \Longrightarrow interior\ S = T$
⟨*proof*⟩
lemma *mem-interior*: $x \in interior\ S \longleftrightarrow (\exists\, e.\ 0 < e \wedge ball\ x\ e \subseteq S)$
⟨*proof*⟩

lemma *open-subset-interior*: *open* S ==> $S \subseteq interior\ T \longleftrightarrow S \subseteq T$
⟨*proof*⟩

lemma *interior-inter*[*simp*]: *interior*$(S \cap T)$ = *interior* $S \cap$ *interior* T
⟨*proof*⟩

lemma *interior-limit-point* [*intro*]:
fixes x :: $'a$::*perfect-space*
assumes x: $x \in interior\ S$ **shows** x *islimpt* S
⟨*proof*⟩

lemma *interior-closed-Un-empty-interior*:
assumes *cS*: *closed* S **and** *iT*: *interior* $T = \{\}$
shows *interior*$(S \cup T)$ = *interior* S
⟨*proof*⟩

16.12 Closure of a Set

definition *closure* $S = S \cup \{x \mid x.\ x\ islimpt\ S\}$

lemma *closure-interior*: *closure* $S = -$ *interior* $(- S)$
⟨*proof*⟩

lemma *interior-closure*: *interior* $S = -$ (*closure* $(- S)$)

⟨proof⟩

lemma *closed-closure*[*simp*, *intro*]: *closed* (*closure* S)
⟨proof⟩

lemma *closure-hull*: *closure* S = *closed hull* S
⟨proof⟩

lemma *closure-eq*: *closure* $S = S \longleftrightarrow$ *closed* S
⟨proof⟩

lemma *closure-closed*[*simp*]: *closed* $S \Longrightarrow$ *closure* $S = S$
⟨proof⟩

lemma *closure-closure*[*simp*]: *closure* (*closure* S) = *closure* S
⟨proof⟩

lemma *closure-subset*: $S \subseteq$ *closure* S
⟨proof⟩

lemma *subset-closure*: $S \subseteq T \Longrightarrow$ *closure* $S \subseteq$ *closure* T
⟨proof⟩

lemma *closure-minimal*: $S \subseteq T \Longrightarrow$ *closed* $T \Longrightarrow$ *closure* $S \subseteq T$
⟨proof⟩

lemma *closure-unique*: $S \subseteq T \wedge$ *closed* $T \wedge (\forall\ T'.\ S \subseteq T' \wedge$ *closed* $T' \longrightarrow T \subseteq T') \Longrightarrow$ *closure* $S = T$
⟨proof⟩

lemma *closure-empty*[*simp*]: *closure* $\{\} = \{\}$
⟨proof⟩

lemma *closure-univ*[*simp*]: *closure* *UNIV* = *UNIV*
⟨proof⟩

lemma *closure-eq-empty*: *closure* $S = \{\} \longleftrightarrow S = \{\}$
⟨proof⟩

lemma *closure-subset-eq*: *closure* $S \subseteq S \longleftrightarrow$ *closed* S
⟨proof⟩

lemma *open-inter-closure-eq-empty*:
open $S \Longrightarrow (S \cap$ *closure* $T) = \{\} \longleftrightarrow S \cap T = \{\}$
⟨proof⟩

lemma *open-inter-closure-subset*:
open $S \Longrightarrow (S \cap$ (*closure* $T)) \subseteq$ *closure*$(S \cap T)$
⟨proof⟩

lemma *closure-complement*: $closure(- S) = - interior(S)$
⟨*proof*⟩

lemma *interior-complement*: $interior(- S) = - closure(S)$
⟨*proof*⟩

16.13 Frontier (aka boundary)

definition $frontier\ S = closure\ S - interior\ S$

lemma *frontier-closed*: $closed(frontier\ S)$
⟨*proof*⟩

lemma *frontier-closures*: $frontier\ S = (closure\ S) \cap (closure(- S))$
⟨*proof*⟩

lemma *frontier-straddle*:
fixes $a :: 'a::metric\text{-}space$
shows $a \in frontier\ S \longleftrightarrow (\forall e>0.\ (\exists x\in S.\ dist\ a\ x < e) \wedge (\exists x.\ x \notin S \wedge dist\ a\ x < e))$ (**is** *?lhs* ⟷ *?rhs*)
⟨*proof*⟩

lemma *frontier-subset-closed*: $closed\ S \Longrightarrow frontier\ S \subseteq S$
⟨*proof*⟩

lemma *frontier-empty*[*simp*]: $frontier\ \{\} = \{\}$
⟨*proof*⟩

lemma *frontier-subset-eq*: $frontier\ S \subseteq S \longleftrightarrow closed\ S$
⟨*proof*⟩

lemma *frontier-complement*: $frontier(- S) = frontier\ S$
⟨*proof*⟩

lemma *frontier-disjoint-eq*: $frontier\ S \cap S = \{\} \longleftrightarrow open\ S$
⟨*proof*⟩

16.14 Nets and the "eventually true" quantifier

Common nets and The "within" modifier for nets.

definition
$at\text{-}infinity :: 'a::real\text{-}normed\text{-}vector\ net$ **where**
$at\text{-}infinity = Abs\text{-}net\ (\lambda P.\ \exists r.\ \forall x.\ r \le norm\ x \longrightarrow P\ x)$

definition
$indirection :: 'a::real\text{-}normed\text{-}vector \Rightarrow 'a \Rightarrow 'a\ net$ (**infixr** *indirection 70*) **where**
$a\ indirection\ v = (at\ a)\ within\ \{b.\ \exists c\ge 0.\ b - a = scaleR\ c\ v\}$

Prove That They are all nets.

lemma *eventually-at-infinity*:
  $eventually\ P\ at\text{-}infinity \longleftrightarrow (\exists\, b.\ \forall\, x.\ norm\ x >= b \longrightarrow P\ x)$
⟨*proof*⟩

Identify Trivial limits, where we can't approach arbitrarily closely.

definition
  $trivial\text{-}limit :: {'a}\ net \Rightarrow bool$ **where**
  $trivial\text{-}limit\ net \longleftrightarrow eventually\ (\lambda x.\ False)\ net$

lemma *trivial-limit-within*:
  shows $trivial\text{-}limit\ (at\ a\ within\ S) \longleftrightarrow \neg\ a\ islimpt\ S$
⟨*proof*⟩

lemma *trivial-limit-at-iff*: $trivial\text{-}limit\ (at\ a) \longleftrightarrow \neg\ a\ islimpt\ UNIV$
  ⟨*proof*⟩

lemma *trivial-limit-at*:
  fixes $a :: {'a}{::}perfect\text{-}space$
  shows $\neg\ trivial\text{-}limit\ (at\ a)$
  ⟨*proof*⟩

lemma *trivial-limit-at-infinity*:
  $\neg\ trivial\text{-}limit\ (at\text{-}infinity :: ({'a}{::}\{real\text{-}normed\text{-}vector,zero\text{-}neq\text{-}one\})\ net)$

  ⟨*proof*⟩

lemma *trivial-limit-sequentially*[*intro*]: $\neg\ trivial\text{-}limit\ sequentially$
  ⟨*proof*⟩

Some property holds "sufficiently close" to the limit point.

lemma *eventually-at*:
  $eventually\ P\ (at\ a) \longleftrightarrow (\exists\, d{>}0.\ \forall\, x.\ 0 < dist\ x\ a \wedge dist\ x\ a < d \longrightarrow P\ x)$
⟨*proof*⟩

lemma *eventually-within*: $eventually\ P\ (at\ a\ within\ S) \longleftrightarrow$
    $(\exists\, d{>}0.\ \forall\, x{\in}S.\ 0 < dist\ x\ a \wedge dist\ x\ a < d \longrightarrow P\ x)$
⟨*proof*⟩

lemma *eventually-within-le*: $eventually\ P\ (at\ a\ within\ S) \longleftrightarrow$
    $(\exists\, d{>}0.\ \forall\, x{\in}S.\ 0 < dist\ x\ a \wedge dist\ x\ a <= d \longrightarrow P\ x)$ (**is** $?lhs = ?rhs$)
⟨*proof*⟩

lemma *eventually-happens*: $eventually\ P\ net ==> trivial\text{-}limit\ net \vee (\exists\, x.\ P\ x)$
  ⟨*proof*⟩

lemma *always-eventually*: $(\forall\, x.\ P\ x) ==> eventually\ P\ net$
⟨*proof*⟩

lemma *trivial-limit-eventually*: $\textit{trivial-limit net} \Longrightarrow \textit{eventually } P \textit{ net}$
⟨*proof*⟩

lemma *eventually-False*: $\textit{eventually } (\lambda x.\ \textit{False}) \textit{ net} \longleftrightarrow \textit{trivial-limit net}$
⟨*proof*⟩

lemma *trivial-limit-eq*: $\textit{trivial-limit net} \longleftrightarrow (\forall P.\ \textit{eventually } P \textit{ net})$
⟨*proof*⟩

Combining theorems for "eventually"

lemma *eventually-conjI*:
$[\![\textit{eventually } (\lambda x.\ P\ x) \textit{ net};\ \textit{eventually } (\lambda x.\ Q\ x) \textit{ net}]\!]$
$\Longrightarrow \textit{eventually } (\lambda x.\ P\ x \wedge Q\ x) \textit{ net}$
⟨*proof*⟩

lemma *eventually-rev-mono*:
$\textit{eventually } P \textit{ net} \Longrightarrow (\forall x.\ P\ x \longrightarrow Q\ x) \Longrightarrow \textit{eventually } Q \textit{ net}$
⟨*proof*⟩

lemma *eventually-and*: $\textit{eventually } (\lambda x.\ P\ x \wedge Q\ x) \textit{ net} \longleftrightarrow \textit{eventually } P \textit{ net} \wedge \textit{eventually } Q \textit{ net}$
⟨*proof*⟩

lemma *eventually-false*: $\textit{eventually } (\lambda x.\ \textit{False}) \textit{ net} \longleftrightarrow \textit{trivial-limit net}$
⟨*proof*⟩

lemma *not-eventually*: $(\forall x.\ \neg\ P\ x\ ) \Longrightarrow {\sim}(\textit{trivial-limit net}) ==> {\sim}(\textit{eventually } (\lambda x.\ P\ x) \textit{ net})$
⟨*proof*⟩

16.15 Limits

Notation Lim to avoid collition with lim defined in analysis

definition
$\textit{Lim} :: {'a}\ \textit{net} \Rightarrow ({'a} \Rightarrow {'b}{::}\textit{t2-space}) \Rightarrow {'b}$ **where**
$\textit{Lim net } f = (\textit{THE } l.\ (f\ {-}{-}{-}{>}\ l) \textit{ net})$

lemma *Lim*:
$(f\ {-}{-}{-}{>}\ l) \textit{ net} \longleftrightarrow$
$\textit{trivial-limit net} \vee$
$(\forall e{>}0.\ \textit{eventually } (\lambda x.\ \textit{dist } (f\ x)\ l < e) \textit{ net})$
⟨*proof*⟩

Show that they yield usual definitions in the various cases.

lemma *Lim-within-le*: $(f\ {-}{-}{-}{>}\ l)(\textit{at } a \textit{ within } S) \longleftrightarrow$
$(\forall e{>}0.\ \exists d{>}0.\ \forall x{\in}S.\ 0 < \textit{dist } x\ a\ \wedge \textit{dist } x\ a\ <= d \longrightarrow \textit{dist } (f\ x)\ l < e)$
⟨*proof*⟩

lemma *Lim-within*: $(f\ \text{---}{>}\ l)\ (at\ a\ within\ S) \longleftrightarrow$
$(\forall\, e > 0.\ \exists\, d>0.\ \forall\, x \in S.\ 0 < dist\ x\ a\ \wedge\ dist\ x\ a\ < d\ \longrightarrow dist\ (f\,x)\ l < e)$
⟨*proof*⟩

lemma *Lim-at*: $(f\ \text{---}{>}\ l)\ (at\ a) \longleftrightarrow$
$(\forall\, e > 0.\ \exists\, d>0.\ \forall\, x.\ 0 < dist\ x\ a\ \wedge\ dist\ x\ a\ < d\ \longrightarrow dist\ (f\,x)\ l < e)$
⟨*proof*⟩

lemma *Lim-at-iff-LIM*: $(f\ \text{---}{>}\ l)\ (at\ a) \longleftrightarrow f\ \text{--}\ a\ \text{--}{>}\ l$
⟨*proof*⟩

lemma *Lim-at-infinity*:
$(f\ \text{---}{>}\ l)\ at\text{-}infinity \longleftrightarrow (\forall\, e>0.\ \exists\, b.\ \forall\, x.\ norm\ x >= b \longrightarrow dist\ (f\,x)\ l < e)$
⟨*proof*⟩

lemma *Lim-sequentially*:
$(S\ \text{---}{>}\ l)\ sequentially \longleftrightarrow$
$(\forall\, e>0.\ \exists\, N.\ \forall\, n \geq N.\ dist\ (S\ n)\ l < e)$
⟨*proof*⟩

lemma *Lim-sequentially-iff-LIMSEQ*: $(S\ \text{---}{>}\ l)\ sequentially \longleftrightarrow S\ \text{----}{>}\ l$
⟨*proof*⟩

lemma *Lim-eventually*: $eventually\ (\lambda x.\ f\,x = l)\ net \Longrightarrow (f\ \text{---}{>}\ l)\ net$
⟨*proof*⟩

The expected monotonicity property.

lemma *Lim-within-empty*: $(f\ \text{---}{>}\ l)\ (net\ within\ \{\})$
⟨*proof*⟩

lemma *Lim-within-subset*: $(f\ \text{---}{>}\ l)\ (net\ within\ S) \Longrightarrow T \subseteq S \Longrightarrow (f\ \text{---}{>}\ l)\ (net\ within\ T)$
⟨*proof*⟩

lemma *Lim-Un*: **assumes** $(f\ \text{---}{>}\ l)\ (net\ within\ S)\ (f\ \text{---}{>}\ l)\ (net\ within\ T)$
shows $(f\ \text{---}{>}\ l)\ (net\ within\ (S \cup T))$
⟨*proof*⟩

lemma *Lim-Un-univ*:
$(f\ \text{---}{>}\ l)\ (net\ within\ S) \Longrightarrow (f\ \text{---}{>}\ l)\ (net\ within\ T) \Longrightarrow\ S \cup T = UNIV$
$\text{==}{>}\ (f\ \text{---}{>}\ l)\ net$
⟨*proof*⟩

Interrelations between restricted and unrestricted limits.

lemma *Lim-at-within*: $(f\ \text{---}{>}\ l)\ net\ \text{==}{>}\ (f\ \text{---}{>}\ l)(net\ within\ S)$

⟨*proof*⟩

lemma *Lim-within-open*:
fixes f :: $'a$::*topological-space* $\Rightarrow$ $'b$::*topological-space*
assumes $a \in S$ *open* S
shows $(f \dashrightarrow l)(at\ a\ within\ S) \longleftrightarrow (f \dashrightarrow l)(at\ a)$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

Another limit point characterization.

lemma *islimpt-sequential*:
fixes x :: $'a$::*metric-space*
shows x *islimpt* $S \longleftrightarrow (\exists f.\ (\forall n{::}nat.\ f\,n \in S - \{x\}) \wedge (f \dashrightarrow x)\ sequentially)$
(**is** *?lhs* = *?rhs*)
⟨*proof*⟩

Basic arithmetical combining theorems for limits.

lemma *Lim-linear*:
assumes $(f \dashrightarrow l)$ *net bounded-linear* h
shows $((\lambda x.\ h\ (f\,x)) \dashrightarrow h\ l)$ *net*
⟨*proof*⟩

lemma *Lim-ident-at*: $((\lambda x.\ x) \dashrightarrow a)\ (at\ a)$
⟨*proof*⟩

lemma *Lim-const*[*intro*]: $((\lambda x.\ a) \dashrightarrow a)$ *net* ⟨*proof*⟩

lemma *Lim-cmul*[*intro*]:
fixes f :: $'a \Rightarrow 'b$::*real-normed-vector*
shows $(f \dashrightarrow l)$ *net* ==> $((\lambda x.\ c *_R f\,x) \dashrightarrow c *_R l)$ *net*
⟨*proof*⟩

lemma *Lim-neg*:
fixes f :: $'a \Rightarrow 'b$::*real-normed-vector*
shows $(f \dashrightarrow l)$ *net* ==> $((\lambda x.\ -(f\,x)) \dashrightarrow -l)$ *net*
⟨*proof*⟩

lemma *Lim-add*: **fixes** f :: $'a \Rightarrow 'b$::*real-normed-vector* **shows**
$(f \dashrightarrow l)$ *net* $\Longrightarrow$ $(g \dashrightarrow m)$ *net* $\Longrightarrow$ $((\lambda x.\ f(x) + g(x)) \dashrightarrow l + m)$ *net*
⟨*proof*⟩

lemma *Lim-sub*:
fixes f :: $'a \Rightarrow 'b$::*real-normed-vector*
shows $(f \dashrightarrow l)$ *net* $\Longrightarrow$ $(g \dashrightarrow m)$ *net* $\Longrightarrow$ $((\lambda x.\ f(x) - g(x)) \dashrightarrow l - m)$ *net*
⟨*proof*⟩

lemma *Lim-mul*:
fixes f :: $'a \Rightarrow 'b$::*real-normed-vector*
assumes $(c \dashrightarrow d)$ *net* $(f \dashrightarrow l)$ *net*

shows $((\lambda x.\ c(x) *_R f\ x) \ {-}{-}{-}{>}\ (d *_R l))$ *net*
⟨*proof*⟩

lemma *Lim-inv*:
fixes $f :: {}'a \Rightarrow real$
assumes $(f \ {-}{-}{-}{>}\ l)\ (net{::}'a\ net)\quad l \neq 0$
shows $((inverse\ o\ f)\ {-}{-}{-}{>}\ inverse\ l)\ net$
⟨*proof*⟩

lemma *Lim-vmul*:
fixes $c :: {}'a \Rightarrow real$ **and** $v :: {}'b{::}real\text{-}normed\text{-}vector$
shows $(c \ {-}{-}{-}{>}\ d)\ net \ {=}{=}{>}\ ((\lambda x.\ c(x) *_R v)\ {-}{-}{-}{>}\ d *_R v)\ net$
⟨*proof*⟩

lemma *Lim-null*:
fixes $f :: {}'a \Rightarrow {}'b{::}real\text{-}normed\text{-}vector$
shows $(f \ {-}{-}{-}{>}\ l)\ net \longleftrightarrow ((\lambda x.\ f(x) - l)\ {-}{-}{-}{>}\ 0)\ net$ ⟨*proof*⟩

lemma *Lim-null-norm*:
fixes $f :: {}'a \Rightarrow {}'b{::}real\text{-}normed\text{-}vector$
shows $(f \ {-}{-}{-}{>}\ 0)\ net \longleftrightarrow ((\lambda x.\ norm(f\ x))\ {-}{-}{-}{>}\ 0)\ net$
⟨*proof*⟩

lemma *Lim-null-comparison*:
fixes $f :: {}'a \Rightarrow {}'b{::}real\text{-}normed\text{-}vector$
assumes $eventually\ (\lambda x.\ norm\ (f\ x) \le g\ x)\ net\ (g \ {-}{-}{-}{>}\ 0)\ net$
shows $(f \ {-}{-}{-}{>}\ 0)\ net$
⟨*proof*⟩

lemma *Lim-component*:
fixes $f :: {}'a \Rightarrow {}'b{::}metric\text{-}space\ \hat{}\ {}'n$
shows $(f \ {-}{-}{-}{>}\ l)\ net \Longrightarrow ((\lambda a.\ f\ a\ \$i)\ {-}{-}{-}{>}\ l\$i)\ net$
⟨*proof*⟩

lemma *Lim-transform-bound*:
fixes $f :: {}'a \Rightarrow {}'b{::}real\text{-}normed\text{-}vector$
fixes $g :: {}'a \Rightarrow {}'c{::}real\text{-}normed\text{-}vector$
assumes $eventually\ (\lambda n.\ norm(f\ n) <= norm(g\ n))\ net\quad (g \ {-}{-}{-}{>}\ 0)\ net$
shows $(f \ {-}{-}{-}{>}\ 0)\ net$
⟨*proof*⟩

Deducing things about the limit from the elements.

lemma *Lim-in-closed-set*:
assumes $closed\ S\ eventually\ (\lambda x.\ f(x) \in S)\ net\ \neg(trivial\text{-}limit\ net)\ (f \ {-}{-}{-}{>}\ l)\ net$
shows $l \in S$
⟨*proof*⟩

Need to prove closed(cball(x,e)) before deducing this as a corollary.

lemma *Lim-dist-ubound*:
assumes $\neg$(*trivial-limit net*) (f $--->$ l) *net eventually* (λx. *dist a* (f x) $<=$ e) *net*
shows *dist a l* $<=$ e
⟨*proof*⟩

lemma *Lim-norm-ubound*:
fixes f :: $'a \Rightarrow 'b$::*real-normed-vector*
assumes $\neg$(*trivial-limit net*) (f $--->$ l) *net eventually* (λx. *norm*(f x) $<=$ e) *net*
shows *norm*(l) $<=$ e
⟨*proof*⟩

lemma *Lim-norm-lbound*:
fixes f :: $'a \Rightarrow 'b$::*real-normed-vector*
assumes $\neg$ (*trivial-limit net*) (f $--->$ l) *net* *eventually* (λx. e $<=$ *norm*(f x)) *net*
shows $e \leq$ *norm* l
⟨*proof*⟩

Uniqueness of the limit, when nontrivial.

lemma *Lim-unique*:
fixes f :: $'a \Rightarrow 'b$::*t2-space*
assumes $\neg$ *trivial-limit net* (f $--->$ l) *net* (f $--->$ l') *net*
shows $l = l'$
⟨*proof*⟩

lemma *tendsto-Lim*:
fixes f :: $'a \Rightarrow 'b$::*t2-space*
shows ~(*trivial-limit net*) $\Longrightarrow$ (f $--->$ l) *net* $==>$ *Lim net f* $=$ l
⟨*proof*⟩

Limit under bilinear function

lemma *Lim-bilinear*:
assumes (f $--->$ l) *net* **and** (g $--->$ m) *net* **and** *bounded-bilinear h*
shows ((λx. h (f x) (g x)) $--->$ (h l m)) *net*
⟨*proof*⟩

These are special for limits out of the same vector space.

lemma *Lim-within-id*: (*id* $--->$ a) (*at a within s*)
⟨*proof*⟩

lemmas *Lim-intros* = *Lim-add Lim-const Lim-sub Lim-cmul Lim-vmul Lim-within-id*

lemma *Lim-at-id*: (*id* $--->$ a) (*at a*)
⟨*proof*⟩

lemma *Lim-at-zero*:
fixes a :: $'a$::*real-normed-vector*

fixes l :: $'b$::*topological-space*
shows $(f \;{-}{-}{-}{>}\; l)$ $(at\ a) \longleftrightarrow ((\lambda x.\ f(a + x)) \;{-}{-}{-}{>}\; l)$ $(at\ 0)$ (**is** $?lhs = ?rhs$)
⟨*proof*⟩

It's also sometimes useful to extract the limit point from the net.

definition
netlimit :: $'a$::*t2-space net* $\Rightarrow$ $'a$ **where**
netlimit net $=$ (*SOME a.* $((\lambda x.\ x) \;{-}{-}{-}{>}\; a)$ *net*)

lemma *netlimit-within*:
assumes $\neg$ *trivial-limit* (*at a within S*)
shows *netlimit* (*at a within S*) $= a$
⟨*proof*⟩

lemma *netlimit-at*:
fixes a :: $'a$::*perfect-space*
shows *netlimit* (*at a*) $= a$
⟨*proof*⟩

Transformation of limit.

lemma *Lim-transform*:
fixes $f\ g$ :: $'a$::*type* $\Rightarrow$ $'b$::*real-normed-vector*
assumes $((\lambda x.\ f\ x - g\ x) \;{-}{-}{-}{>}\; 0)$ *net* $(f \;{-}{-}{-}{>}\; l)$ *net*
shows $(g \;{-}{-}{-}{>}\; l)$ *net*
⟨*proof*⟩

lemma *Lim-transform-eventually*:
eventually $(\lambda x.\ f\ x = g\ x)$ *net* $\Longrightarrow (f \;{-}{-}{-}{>}\; l)$ *net* $\Longrightarrow (g \;{-}{-}{-}{>}\; l)$ *net*
⟨*proof*⟩

lemma *Lim-transform-within*:
assumes $0 < d$ **and** $\forall x' \in S.\ 0 < dist\ x'\ x \wedge dist\ x'\ x < d \longrightarrow f\ x' = g\ x'$
and $(f \;{-}{-}{-}{>}\; l)$ (*at x within S*)
shows $(g \;{-}{-}{-}{>}\; l)$ (*at x within S*)
⟨*proof*⟩

lemma *Lim-transform-at*:
assumes $0 < d$ **and** $\forall x'.\ 0 < dist\ x'\ x \wedge dist\ x'\ x < d \longrightarrow f\ x' = g\ x'$
and $(f \;{-}{-}{-}{>}\; l)$ (*at x*)
shows $(g \;{-}{-}{-}{>}\; l)$ (*at x*)
⟨*proof*⟩

Common case assuming being away from some crucial point like 0.

lemma *Lim-transform-away-within*:
fixes $a\ b$ :: $'a$::*t1-space*
assumes $a \neq b$ **and** $\forall x \in S.\ x \neq a \wedge x \neq b \longrightarrow f\ x = g\ x$
and $(f \;{-}{-}{-}{>}\; l)$ (*at a within S*)
shows $(g \;{-}{-}{-}{>}\; l)$ (*at a within S*)
⟨*proof*⟩

lemma *Lim-transform-away-at*:
fixes $a\ b$ :: $'a::t1\text{-}space$
assumes *ab*: $a \neq b$ **and** *fg*: $\forall x.\ x \neq a \wedge x \neq b \longrightarrow f\ x = g\ x$
and *fl*: $(f\ \text{--->}\ l)\ (at\ a)$
shows $(g\ \text{--->}\ l)\ (at\ a)$
⟨*proof*⟩

Alternatively, within an open set.

lemma *Lim-transform-within-open*:
assumes $open\ S$ **and** $a \in S$ **and** $\forall x{\in}S.\ x \neq a \longrightarrow f\ x = g\ x$
and $(f\ \text{--->}\ l)\ (at\ a)$
shows $(g\ \text{--->}\ l)\ (at\ a)$
⟨*proof*⟩

A congruence rule allowing us to transform limits assuming not at point.

lemma *Lim-cong-within*:
assumes $\bigwedge x.\ x \neq a \Longrightarrow f\ x = g\ x$
shows $((\lambda x.\ f\ x)\ \text{--->}\ l)\ (at\ a\ within\ S) \longleftrightarrow ((g\ \text{--->}\ l)\ (at\ a\ within\ S))$
⟨*proof*⟩

lemma *Lim-cong-at*:
assumes $\bigwedge x.\ x \neq a \Longrightarrow f\ x = g\ x$
shows $((\lambda x.\ f\ x)\ \text{--->}\ l)\ (at\ a) \longleftrightarrow ((g\ \text{--->}\ l)\ (at\ a))$
⟨*proof*⟩

Useful lemmas on closure and set of possible sequential limits.

lemma *closure-sequential*:
fixes l :: $'a::metric\text{-}space$
shows $l \in closure\ S \longleftrightarrow (\exists x.\ (\forall n.\ x\ n \in S) \wedge (x\ \text{--->}\ l)\ sequentially)$ (**is** $?lhs = ?rhs$)
⟨*proof*⟩

lemma *closed-sequential-limits*:
fixes S :: $'a::metric\text{-}space\ set$
shows $closed\ S \longleftrightarrow (\forall x\ l.\ (\forall n.\ x\ n \in S) \wedge (x\ \text{--->}\ l)\ sequentially \longrightarrow l \in S)$
⟨*proof*⟩

lemma *closure-approachable*:
fixes S :: $'a::metric\text{-}space\ set$
shows $x \in closure\ S \longleftrightarrow (\forall e>0.\ \exists y{\in}S.\ dist\ y\ x < e)$
⟨*proof*⟩

lemma *closed-approachable*:
fixes S :: $'a::metric\text{-}space\ set$
shows $closed\ S \text{ ==> } (\forall e>0.\ \exists y{\in}S.\ dist\ y\ x < e) \longleftrightarrow x \in S$
⟨*proof*⟩

Some other lemmas about sequences.

lemma *sequentially-offset*:
assumes *eventually* ($\lambda i.\ P\ i$) *sequentially*
shows *eventually* ($\lambda i.\ P\ (i + k)$) *sequentially*
⟨*proof*⟩

lemma *seq-offset*:
assumes (f ---> l) *sequentially*
shows (($\lambda i.\ f\ (i + k)$) ---> l) *sequentially*
⟨*proof*⟩

lemma *seq-offset-neg*:
(f ---> l) *sequentially* ==> (($\lambda i.\ f(i - k)$) ---> l) *sequentially*
⟨*proof*⟩

lemma *seq-offset-rev*:
(($\lambda i.\ f(i + k)$) ---> l) *sequentially* ==> (f ---> l) *sequentially*
⟨*proof*⟩

lemma *seq-harmonic*: (($\lambda n.$ *inverse* (*real* n)) ---> 0) *sequentially*
⟨*proof*⟩

16.16 More properties of closed balls.

lemma *closed-cball*: *closed* (*cball* x e)
⟨*proof*⟩

lemma *open-contains-cball*: *open* $S \longleftrightarrow (\forall x{\in}S.\ \exists e{>}0.$ *cball* x $e \subseteq S)$
⟨*proof*⟩

lemma *open-contains-cball-eq*: *open* S ==> $(\forall x.\ x \in S \longleftrightarrow (\exists e{>}0.$ *cball* x $e \subseteq S))$
⟨*proof*⟩

lemma *mem-interior-cball*: $x \in$ *interior* $S \longleftrightarrow (\exists e{>}0.$ *cball* x $e \subseteq S)$
⟨*proof*⟩

lemma *islimpt-ball*:
fixes x y :: $'a$::{*real-normed-vector*,*perfect-space*}
shows y *islimpt ball* x $e \longleftrightarrow 0 < e \wedge y \in$ *cball* x e (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *closure-ball-lemma*:
fixes x y :: $'a$::*real-normed-vector*
assumes $x \neq y$ **shows** y *islimpt ball* x (*dist* x y)
⟨*proof*⟩

lemma *closure-ball*:
fixes x :: $'a$::*real-normed-vector*

shows $0 < e \Longrightarrow$ *closure* (*ball x e*) = *cball x e*
⟨*proof*⟩

lemma *interior-cball*:
fixes x :: $'a$::{*real-normed-vector*, *perfect-space*}
shows *interior* (*cball x e*) = *ball x e*
⟨*proof*⟩

lemma *frontier-ball*:
fixes a :: $'a$::*real-normed-vector*
shows $0 < e$ ==> *frontier*(*ball a e*) = {*x. dist a x = e*}
⟨*proof*⟩

lemma *frontier-cball*:
fixes a :: $'a$::{*real-normed-vector*, *perfect-space*}
shows *frontier*(*cball a e*) = {*x. dist a x = e*}
⟨*proof*⟩

lemma *cball-eq-empty*: (*cball x e* = {}) $\longleftrightarrow$ $e < 0$
⟨*proof*⟩
lemma *cball-empty*: $e < 0$ ==> *cball x e* = {} ⟨*proof*⟩

lemma *cball-eq-sing*:
fixes x :: $'a$::*perfect-space*
shows (*cball x e* = {*x*}) $\longleftrightarrow$ $e = 0$
⟨*proof*⟩

lemma *cball-sing*:
fixes x :: $'a$::*metric-space*
shows $e = 0$ ==> *cball x e* = {*x*}
⟨*proof*⟩

For points in the interior, localization of limits makes no difference.

lemma *eventually-within-interior*:
assumes $x \in$ *interior S*
shows *eventually P* (*at x within S*) $\longleftrightarrow$ *eventually P* (*at x*) (**is** *?lhs = ?rhs*)
⟨*proof*⟩

lemma *lim-within-interior*:
$x \in$ *interior S* $\Longrightarrow$ (*f* −−−> *l*) (*at x within S*) $\longleftrightarrow$ (*f* −−−> *l*) (*at x*)
⟨*proof*⟩

lemma *netlimit-within-interior*:
fixes x :: $'a$::{*perfect-space*, *real-normed-vector*}

assumes $x \in$ *interior S*
shows *netlimit*(*at x within S*) = *x* (**is** *?lhs = ?rhs*)
⟨*proof*⟩

16.17 Boundedness.

definition
bounded :: $'a$::*metric-space set* $\Rightarrow$ *bool* **where**
bounded $S \longleftrightarrow (\exists\, x\ e.\ \forall\, y{\in}S.\ dist\ x\ y \le e)$

lemma *bounded-any-center*: *bounded* $S \longleftrightarrow (\exists\, e.\ \forall\, y{\in}S.\ dist\ a\ y \le e)$
⟨*proof*⟩

lemma *bounded-iff*: *bounded* $S \longleftrightarrow (\exists\, a.\ \forall\, x{\in}S.\ norm\ x \le a)$
⟨*proof*⟩

lemma *bounded-empty*[*simp*]: *bounded* {} ⟨*proof*⟩
lemma *bounded-subset*: *bounded* $T \Longrightarrow S \subseteq T$ ==> *bounded* S
⟨*proof*⟩

lemma *bounded-interior*[*intro*]: *bounded* S ==> *bounded*(*interior* S)
⟨*proof*⟩

lemma *bounded-closure*[*intro*]: **assumes** *bounded* S **shows** *bounded*(*closure* S)
⟨*proof*⟩

lemma *bounded-cball*[*simp*,*intro*]: *bounded* (*cball* x e)
⟨*proof*⟩

lemma *bounded-ball*[*simp*,*intro*]: *bounded*(*ball* x e)
⟨*proof*⟩

lemma *finite-imp-bounded*[*intro*]:
fixes S :: $'a$::*metric-space set* **assumes** *finite* S **shows** *bounded* S
⟨*proof*⟩

lemma *bounded-Un*[*simp*]: *bounded* $(S \cup T) \longleftrightarrow$ *bounded* $S \wedge$ *bounded* T
⟨*proof*⟩

lemma *bounded-Union*[*intro*]: *finite* $F \Longrightarrow (\forall\, S{\in}F.\ bounded\ S) \Longrightarrow bounded(\bigcup F)$
⟨*proof*⟩

lemma *bounded-pos*: *bounded* $S \longleftrightarrow (\exists\, b{>}0.\ \forall\, x{\in}\ S.\ norm\ x <= b)$
⟨*proof*⟩

lemma *bounded-Int*[*intro*]: *bounded* $S \vee$ *bounded* $T \Longrightarrow$ *bounded* $(S \cap T)$
⟨*proof*⟩

lemma *bounded-diff*[*intro*]: *bounded* S ==> *bounded* $(S - T)$
⟨*proof*⟩

lemma *bounded-insert*[*intro*]:*bounded*(*insert* x S) $\longleftrightarrow$ *bounded* S
⟨*proof*⟩

lemma *not-bounded-UNIV*[*simp*, *intro*]:
 $\neg$ *bounded* (*UNIV* :: *'a*::{*real-normed-vector*, *perfect-space*} *set*)
⟨*proof*⟩

lemma *bounded-linear-image*:
 assumes *bounded S bounded-linear f*
 shows *bounded*(*f* ‘ *S*)
⟨*proof*⟩

lemma *bounded-scaling*:
 fixes *S* :: *'a*::*real-normed-vector set*
 shows *bounded S* $\Longrightarrow$ *bounded* (($\lambda x.\ c *_R x$) ‘ *S*)
 ⟨*proof*⟩

lemma *bounded-translation*:
 fixes *S* :: *'a*::*real-normed-vector set*
 assumes *bounded S* **shows** *bounded* (($\lambda x.\ a + x$) ‘ *S*)
⟨*proof*⟩

Some theorems on sups and infs using the notion "bounded".

lemma *bounded-real*:
 fixes *S* :: *real set*
 shows *bounded S* $\longleftrightarrow$ ($\exists\, a.\ \forall\, x{\in}S.\ abs\ x <= a$)
 ⟨*proof*⟩

lemma *bounded-has-Sup*:
 fixes *S* :: *real set*
 assumes *bounded S S* $\neq$ {}
 shows $\forall\, x{\in}S.\ x <= Sup\ S$ **and** $\forall\, b.\ (\forall\, x{\in}S.\ x <= b) \longrightarrow Sup\ S <= b$
⟨*proof*⟩

lemma *Sup-insert*:
 fixes *S* :: *real set*
 shows *bounded S* ==> *Sup*(*insert x S*) = (*if S* = {} *then x else max x* (*Sup S*))

⟨*proof*⟩

lemma *Sup-insert-finite*:
 fixes *S* :: *real set*
 shows *finite S* $\Longrightarrow$ *Sup*(*insert x S*) = (*if S* = {} *then x else max x* (*Sup S*))
 ⟨*proof*⟩

lemma *bounded-has-Inf*:
 fixes *S* :: *real set*
 assumes *bounded S S* $\neq$ {}
 shows $\forall\, x{\in}S.\ x >= Inf\ S$ **and** $\forall\, b.\ (\forall\, x{\in}S.\ x >= b) \longrightarrow Inf\ S >= b$
⟨*proof*⟩

lemma *Inf-insert*:

fixes S :: *real set*
shows *bounded* S ==> *Inf*(*insert* x S) = (*if* S = {} *then* x *else min* x (*Inf* S))
⟨*proof*⟩
lemma *Inf-insert-finite*:
fixes S :: *real set*
shows *finite* S ==> *Inf*(*insert* x S) = (*if* S = {} *then* x *else min* x (*Inf* S))
⟨*proof*⟩

lemma *real-isGlb-unique*: [| *isGlb R S x*; *isGlb R S y* |] ==> x = (y::*real*)
⟨*proof*⟩

16.18 Equivalent versions of compactness

16.18.1 Sequential compactness

definition
compact :: *'a::metric-space set* ⇒ *bool* **where**
compact S ⟷
(∀ f. (∀ n. f n ∈ S) ⟶
(∃ l∈S. ∃ r. *subseq* r ∧ ((f *o* r) −−−> l) *sequentially*))

A metric space (or topological vector space) is said to have the Heine-Borel property if every closed and bounded subset is compact.

class *heine-borel* =
assumes *bounded-imp-convergent-subsequence*:
bounded s ⟹ ∀ n. f n ∈ s
⟹ ∃ l r. *subseq* r ∧ ((f ∘ r) −−−> l) *sequentially*

lemma *bounded-closed-imp-compact*:
fixes s::*'a::heine-borel set*
assumes *bounded* s **and** *closed* s **shows** *compact* s
⟨*proof*⟩

lemma *subseq-bigger*: **assumes** *subseq* r **shows** $n \le r\ n$
⟨*proof*⟩

lemma *eventually-subseq*:
assumes r: *subseq* r
shows *eventually* P *sequentially* ⟹ *eventually* (λn. P (r n)) *sequentially*
⟨*proof*⟩

lemma *lim-subseq*:
subseq r ⟹ (s −−−> l) *sequentially* ⟹ ((s *o* r) −−−> l) *sequentially*
⟨*proof*⟩

lemma *num-Axiom*: *EX*! g. g *0* = e ∧ (∀ n. g (*Suc* n) = f n (g n))
⟨*proof*⟩

lemma *convergent-bounded-increasing*: **fixes** s ::$nat \Rightarrow real$
assumes $incseq\ s$ **and** $\forall n.\ abs(s\ n) \leq b$
shows $\exists\ l.\ \forall e::real>0.\ \exists\ N.\ \forall n \geq N.\ \ abs(s\ n\ -\ l) < e$
⟨*proof*⟩

lemma *convergent-bounded-monotone*: **fixes** $s::nat \Rightarrow real$
assumes $\forall n.\ abs(s\ n) \leq b$ **and** $monoseq\ s$
shows $\exists l.\ \forall e::real>0.\ \exists N.\ \forall n \geq N.\ abs(s\ n\ -\ l) < e$
⟨*proof*⟩

lemma *compact-real-lemma*:
assumes $\forall n::nat.\ abs(s\ n) \leq b$
shows $\exists (l::real)\ r.\ subseq\ r \wedge ((s \circ r) \ {-}{-}{-}{>}\ l)\ sequentially$
⟨*proof*⟩

instance *real* :: *heine-borel*
⟨*proof*⟩

lemma *bounded-component*: $bounded\ s \Longrightarrow bounded\ ((\lambda x.\ x\ \$\ i)\ `\ s)$
⟨*proof*⟩

lemma *compact-lemma*:
fixes f :: $nat \Rightarrow 'a::heine\text{-}borel\ \hat{}\ 'n$
assumes $bounded\ s$ **and** $\forall n.\ f\ n \in s$
shows $\forall d.$
$\exists l\ r.\ subseq\ r \wedge$
$(\forall e>0.\ eventually\ (\lambda n.\ \forall i \in d.\ dist\ (f\ (r\ n)\ \$\ i)\ (l\ \$\ i) < e)\ sequentially)$
⟨*proof*⟩

instance *cart* :: (*heine-borel*, *finite*) *heine-borel*
⟨*proof*⟩

lemma *bounded-fst*: $bounded\ s \Longrightarrow bounded\ (fst\ `\ s)$
⟨*proof*⟩

lemma *bounded-snd*: $bounded\ s \Longrightarrow bounded\ (snd\ `\ s)$
⟨*proof*⟩

instance $*$ :: (*heine-borel*, *heine-borel*) *heine-borel*
⟨*proof*⟩

16.18.2 Completeness

lemma *cauchy-def*:
$Cauchy\ s \longleftrightarrow (\forall e>0.\ \exists N.\ \forall m\ n.\ m \geq N \wedge n \geq N\ {-}{-}{>}\ dist(s\ m)(s\ n) < e)$
⟨*proof*⟩

definition
complete :: $'a::metric\text{-}space\ set \Rightarrow bool$ **where**

complete s $\longleftrightarrow$ $(\forall f.\ (\forall n.\ f\ n \in s) \wedge \mathit{Cauchy}\ f$
$\quad --> (\exists l \in s.\ (f\ ---> l)\ \mathit{sequentially}))$

lemma *cauchy*: $\mathit{Cauchy}\ s \longleftrightarrow (\forall e{>}0.\exists\ N{::}nat.\ \forall n{\geq}N.\ dist(s\ n)(s\ N) < e)$ (**is** *?lhs = ?rhs*)
⟨*proof*⟩

lemma *convergent-imp-cauchy*:
$(s\ ---> l)$ *sequentially* $==>$ *Cauchy s*
⟨*proof*⟩

lemma *cauchy-imp-bounded*: **assumes** *Cauchy s* **shows** *bounded* (*range s*)
⟨*proof*⟩

lemma *compact-imp-complete*: **assumes** *compact s* **shows** *complete s*
⟨*proof*⟩

instance *heine-borel* < *complete-space*
⟨*proof*⟩

lemma *complete-univ*: *complete* (*UNIV* :: *'a::complete-space set*)
⟨*proof*⟩

lemma *complete-imp-closed*: **assumes** *complete s* **shows** *closed s*
⟨*proof*⟩

lemma *complete-eq-closed*:
  fixes *s* :: *'a::complete-space set*
  shows *complete s* $\longleftrightarrow$ *closed s* (**is** *?lhs = ?rhs*)
⟨*proof*⟩

lemma *convergent-eq-cauchy*:
  fixes *s* :: *nat* $\Rightarrow$ *'a::complete-space*
  shows $(\exists l.\ (s\ ---> l)\ \mathit{sequentially}) \longleftrightarrow \mathit{Cauchy}\ s$ (**is** *?lhs = ?rhs*)
⟨*proof*⟩

lemma *convergent-imp-bounded*:
  fixes *s* :: *nat* $\Rightarrow$ *'a::metric-space*
  shows $(s\ ---> l)$ *sequentially* $==>$ *bounded* (*s* ` (*UNIV*::(*nat set*)))
  ⟨*proof*⟩

16.18.3 Total boundedness

fun *helper-1*::(*'a::metric-space set*) $\Rightarrow$ *real* $\Rightarrow$ *nat* $\Rightarrow$ *'a* **where**
  helper-1 s e n = (*SOME y::'a. y* $\in$ *s* $\wedge$ $(\forall m{<}n.\ \neg$ (*dist* (*helper-1 s e m*) *y* < *e*)))
declare *helper-1.simps*[*simp del*]

lemma *compact-imp-totally-bounded*:

assumes *compact s*
shows $\forall\, e{>}0.\ \exists\, k.\ \mathit{finite}\ k \wedge k \subseteq s \wedge s \subseteq (\bigcup((\lambda x.\ \mathit{ball}\ x\ e)\ \text{‘}\ k))$
⟨*proof*⟩

16.18.4 Heine-Borel theorem

Following Burkill & Burkill vol. 2.

lemma *heine-borel-lemma*: **fixes** s::$'a$::*metric-space set*
assumes *compact s* $s \subseteq (\bigcup\ t)$ $\forall\, b \in t.\ \mathit{open}\ b$
shows $\exists\, e{>}0.\ \forall\, x \in s.\ \exists\, b \in t.\ \mathit{ball}\ x\ e \subseteq b$
⟨*proof*⟩

lemma *compact-imp-heine-borel*: *compact s* ==> $(\forall f.\ (\forall\, t \in f.\ \mathit{open}\ t) \wedge s \subseteq (\bigcup\ f)$
$\longrightarrow (\exists f'.\ f' \subseteq f \wedge \mathit{finite}\ f' \wedge s \subseteq (\bigcup\ f')))$
⟨*proof*⟩

16.18.5 Bolzano-Weierstrass property

lemma *heine-borel-imp-bolzano-weierstrass*:
assumes $\forall f.\ (\forall\, t \in f.\ \mathit{open}\ t) \wedge s \subseteq (\bigcup\ f)$ --> $(\exists f'.\ f' \subseteq f \wedge \mathit{finite}\ f' \wedge s \subseteq (\bigcup\ f'))$
infinite t $t \subseteq s$
shows $\exists\, x \in s.\ x$ *islimpt t*
⟨*proof*⟩

16.18.6 Complete the chain of compactness variants

primrec *helper-2*::(*real* ⇒ $'a$::*metric-space*) ⇒ *nat* ⇒ $'a$ **where**
helper-2 beyond 0 = beyond 0 |
helper-2 beyond (*Suc n*) = *beyond* (*dist undefined* (*helper-2 beyond n*) + *1*)

lemma *bolzano-weierstrass-imp-bounded*: **fixes** s::$'a$::*metric-space set*
assumes $\forall\, t.\ \mathit{infinite}\ t \wedge t \subseteq s$ --> $(\exists\, x \in s.\ x$ *islimpt t*)
shows *bounded s*
⟨*proof*⟩

lemma *sequence-infinite-lemma*:
fixes l :: $'a$::*metric-space*
assumes $\forall\, n$::*nat*. $(f\ n \neq l)$ $(f$ ---> $l)$ *sequentially*
shows *infinite* (*range f*)
⟨*proof*⟩

lemma *sequence-unique-limpt*:
fixes l :: $'a$::*metric-space*
assumes $\forall\, n$::*nat*. $(f\ n \neq l)$ $(f$ ---> $l)$ *sequentially* l' *islimpt* (*range f*)
shows $l' = l$
⟨*proof*⟩

lemma *bolzano-weierstrass-imp-closed*:
fixes s :: *'a::metric-space set*
assumes $\forall\, t.$ *infinite* $t \wedge t \subseteq s$ $-->$ $(\exists\, x \in s.\ x$ *islimpt* $t)$
shows *closed* s
⟨*proof*⟩

Hence express everything as an equivalence.

lemma *compact-eq-heine-borel*:
fixes s :: *'a::heine-borel set*
shows *compact* $s \longleftrightarrow$
$(\forall f.\ (\forall\, t \in f.$ *open* $t) \wedge s \subseteq (\bigcup f)$
$-->$ $(\exists f'.\ f' \subseteq f \wedge$ *finite* $f' \wedge s \subseteq (\bigcup f')))$ (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *compact-eq-bolzano-weierstrass*:
fixes s :: *'a::heine-borel set*
shows *compact* $s \longleftrightarrow (\forall\, t.$ *infinite* $t \wedge t \subseteq s$ $-->$ $(\exists\, x \in s.\ x$ *islimpt* $t))$ (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *compact-eq-bounded-closed*:
fixes s :: *'a::heine-borel set*
shows *compact* $s \longleftrightarrow$ *bounded* $s \wedge$ *closed* s (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *compact-imp-bounded*:
fixes s :: *'a::metric-space set*
shows *compact* s ==> *bounded* s
⟨*proof*⟩

lemma *compact-imp-closed*:
fixes s :: *'a::metric-space set*
shows *compact* s ==> *closed* s
⟨*proof*⟩

In particular, some common special cases.

lemma *compact-empty*[*simp*]:
compact {}
⟨*proof*⟩

lemma *compact-union*[*intro*]:
fixes s t :: *'a::heine-borel set*
shows *compact* $s \Longrightarrow$ *compact* t ==> *compact* $(s \cup t)$
⟨*proof*⟩

lemma *compact-inter*[*intro*]:

fixes *s t* :: *'a::heine-borel set*
shows *compact s* $\Longrightarrow$ *compact t* ==> *compact* $(s \cap t)$
⟨*proof*⟩

lemma *compact-inter-closed*[*intro*]:
fixes *s t* :: *'a::heine-borel set*
shows *compact s* $\Longrightarrow$ *closed t* ==> *compact* $(s \cap t)$
⟨*proof*⟩

lemma *closed-inter-compact*[*intro*]:
fixes *s t* :: *'a::heine-borel set*
shows *closed s* $\Longrightarrow$ *compact t* ==> *compact* $(s \cap t)$
⟨*proof*⟩

lemma *finite-imp-compact*:
fixes *s* :: *'a::heine-borel set*
shows *finite s* ==> *compact s*
⟨*proof*⟩

lemma *compact-sing* [*simp*]: *compact* $\{a\}$
⟨*proof*⟩

lemma *compact-cball*[*simp*]:
fixes *x* :: *'a::heine-borel*
shows *compact*(*cball x e*)
⟨*proof*⟩

lemma *compact-frontier-bounded*[*intro*]:
fixes *s* :: *'a::heine-borel set*
shows *bounded s* ==> *compact*(*frontier s*)
⟨*proof*⟩

lemma *compact-frontier*[*intro*]:
fixes *s* :: *'a::heine-borel set*
shows *compact s* ==> *compact* (*frontier s*)
⟨*proof*⟩

lemma *frontier-subset-compact*:
fixes *s* :: *'a::heine-borel set*
shows *compact s* ==> *frontier s* $\subseteq$ *s*
⟨*proof*⟩

lemma *open-delete*:
fixes *s* :: *'a::t1-space set*
shows *open s* $\Longrightarrow$ *open* $(s - \{x\})$
⟨*proof*⟩

Finite intersection property. I could make it an equivalence in fact.

lemma *compact-imp-fip*:

fixes s :: $'a$::*heine-borel set*
assumes *compact* s $\forall t \in f.$ *closed* t
$\forall f'.$ *finite* $f' \wedge f' \subseteq f$ $-->$ $(s \cap (\bigcap f') \neq \{\})$
shows $s \cap (\bigcap f) \neq \{\}$
⟨*proof*⟩

16.19 Bounded closed nest property (proof does not use Heine-Borel).

lemma *bounded-closed-nest*:
assumes $\forall n.$ *closed*$(s\ n)$ $\forall n.$ $(s\ n \neq \{\})$
$(\forall m\ n.\ m \leq n$ $-->$ $s\ n \subseteq s\ m)$ *bounded*$(s\ 0)$
shows $\exists a$::$'a$::*heine-borel*. $\forall n$::*nat*. $a \in s(n)$
⟨*proof*⟩

Decreasing case does not even need compactness, just completeness.

lemma *decreasing-closed-nest*:
assumes $\forall n.$ *closed*$(s\ n)$
$\forall n.$ $(s\ n \neq \{\})$
$\forall m\ n.\ m \leq n$ $-->$ $s\ n \subseteq s\ m$
$\forall e{>}0.\ \exists n.\ \forall x \in (s\ n).\ \forall\ y \in (s\ n).$ *dist* $x\ y < e$
shows $\exists a$::$'a$::*heine-borel*. $\forall n$::*nat*. $a \in s\ n$
⟨*proof*⟩

Strengthen it to the intersection actually being a singleton.

lemma *decreasing-closed-nest-sing*:
fixes s :: *nat* $\Rightarrow$ $'a$::*heine-borel set*
assumes $\forall n.$ *closed*$(s\ n)$
$\forall n.\ s\ n \neq \{\}$
$\forall m\ n.\ m \leq n$ $-->$ $s\ n \subseteq s\ m$
$\forall e{>}0.\ \exists n.\ \forall x \in (s\ n).\ \forall\ y{\in}(s\ n).$ *dist* $x\ y < e$
shows $\exists a.\ \bigcap($*range* $s) = \{a\}$
⟨*proof*⟩

Cauchy-type criteria for uniform convergence.

lemma *uniformly-convergent-eq-cauchy*: **fixes** s::*nat* $\Rightarrow$ $'b$ $\Rightarrow$ $'a$::*heine-borel* **shows**
$(\exists l.\ \forall e{>}0.\ \exists N.\ \forall n\ x.\ N \leq n \wedge P\ x$ $-->$ *dist*$(s\ n\ x)(l\ x) < e) \longleftrightarrow$
$(\forall e{>}0.\ \exists N.\ \forall m\ n\ x.\ N \leq m \wedge N \leq n \wedge P\ x$ $-->$ *dist* $(s\ m\ x)\ (s\ n\ x) < e)$
(**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *uniformly-cauchy-imp-uniformly-convergent*:
fixes s :: *nat* $\Rightarrow$ $'a$ $\Rightarrow$ $'b$::*heine-borel*
assumes $\forall e{>}0.\exists N.\ \forall m\ (n$::*nat*$)\ x.\ N \leq m \wedge N \leq n \wedge P\ x$ $-->$ *dist*$(s\ m\ x)(s\ n\ x) < e$
$\forall x.\ P\ x$ $-->$ $(\forall e{>}0.\ \exists N.\ \forall n.\ N \leq n$ $-->$ *dist*$(s\ n\ x)(l\ x) < e)$
shows $\forall e{>}0.\ \exists N.\ \forall n\ x.\ N \leq n \wedge P\ x$ $-->$ *dist*$(s\ n\ x)(l\ x) < e$
⟨*proof*⟩

16.20 Continuity

Define continuity over a net to take in restrictions of the set.

definition
continuous :: *'a::t2-space net* ⇒ (*'a* ⇒ *'b::topological-space*) ⇒ *bool* **where**
continuous net f ⟷ (*f* ---> *f*(*netlimit net*)) *net*

lemma *continuous-trivial-limit*:
trivial-limit net ==> *continuous net f*
⟨*proof*⟩

lemma *continuous-within*: *continuous* (*at x within s*) *f* ⟷ (*f* ---> *f*(*x*)) (*at x within s*)
⟨*proof*⟩

lemma *continuous-at*: *continuous* (*at x*) *f* ⟷ (*f* ---> *f*(*x*)) (*at x*)
⟨*proof*⟩

lemma *continuous-at-within*:
assumes *continuous* (*at x*) *f* **shows** *continuous* (*at x within s*) *f*
⟨*proof*⟩

Derive the epsilon-delta forms, which we often use as "definitions"

lemma *continuous-within-eps-delta*:
continuous (*at x within s*) *f* ⟷ (∀ *e*>*0*. ∃ *d*>*0*. ∀ *x'*∈ *s*. *dist x' x* < *d* --> *dist* (*f x'*) (*f x*) < *e*)
⟨*proof*⟩

lemma *continuous-at-eps-delta*: *continuous* (*at x*) *f* ⟷ (∀ *e*>*0*. ∃ *d*>*0*.
∀ *x'*. *dist x' x* < *d* --> *dist*(*f x'*)(*f x*) < *e*)
⟨*proof*⟩

Versions in terms of open balls.

lemma *continuous-within-ball*:
continuous (*at x within s*) *f* ⟷ (∀ *e*>*0*. ∃ *d*>*0*.
f ` (*ball x d* ∩ *s*) ⊆ *ball* (*f x*) *e*) (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *continuous-at-ball*:
continuous (*at x*) *f* ⟷ (∀ *e*>*0*. ∃ *d*>*0*. *f* ` (*ball x d*) ⊆ *ball* (*f x*) *e*) (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

Define setwise continuity in terms of limits within the set.

definition
continuous-on ::
'a set ⇒ (*'a::topological-space* ⇒ *'b::topological-space*) ⇒ *bool*
where

continuous-on s f $\longleftrightarrow$ $(\forall\, x{\in}s.\ (f\ \text{--->}\ f\ x)\ (at\ x\ within\ s))$

lemma *continuous-on-topological*:
continuous-on s f $\longleftrightarrow$
$(\forall\, x{\in}s.\ \forall\, B.\ open\ B \longrightarrow f\ x \in B \longrightarrow$
$(\exists\, A.\ open\ A \wedge x \in A \wedge (\forall\, y{\in}s.\ y \in A \longrightarrow f\ y \in B)))$
⟨*proof*⟩

lemma *continuous-on-iff*:
continuous-on s f $\longleftrightarrow$
$(\forall\, x{\in}s.\ \forall\, e{>}0.\ \exists\, d{>}0.\ \forall\, x'{\in}s.\ dist\ x'\ x < d \longrightarrow dist\ (f\ x')\ (f\ x) < e)$
⟨*proof*⟩

definition
uniformly-continuous-on ::
$'a\ set \Rightarrow ('a{::}metric\text{-}space \Rightarrow 'b{::}metric\text{-}space) \Rightarrow bool$
where
uniformly-continuous-on s f $\longleftrightarrow$
$(\forall\, e{>}0.\ \exists\, d{>}0.\ \forall\, x{\in}s.\ \forall\, x'{\in}s.\ dist\ x'\ x < d \longrightarrow dist\ (f\ x')\ (f\ x) < e)$

Some simple consequential lemmas.

lemma *uniformly-continuous-imp-continuous*:
uniformly-continuous-on s f ==> *continuous-on s f*
⟨*proof*⟩

lemma *continuous-at-imp-continuous-within*:
continuous (*at x*) *f* ==> *continuous* (*at x within s*) *f*
⟨*proof*⟩

lemma *Lim-trivial-limit*: *trivial-limit net* $\Longrightarrow$ $(f\ \text{--->}\ l)$ *net*
⟨*proof*⟩

lemma *continuous-at-imp-continuous-on*:
assumes $\forall\, x{\in}s.$ *continuous* (*at x*) *f*
shows *continuous-on s f*
⟨*proof*⟩

lemma *continuous-on-eq-continuous-within*:
continuous-on s f $\longleftrightarrow$ $(\forall\, x \in s.$ *continuous* (*at x within s*) *f*$)$
⟨*proof*⟩

lemmas *continuous-on* = *continuous-on-def* — legacy theorem name

lemma *continuous-on-eq-continuous-at*:
shows *open s* ==> (*continuous-on s f* $\longleftrightarrow$ $(\forall\, x \in s.$ *continuous* (*at x*) *f*$)$)
⟨*proof*⟩

lemma *continuous-within-subset*:
continuous (*at x within s*) *f* $\Longrightarrow$ $t \subseteq s$

==> *continuous* (*at x within t*) *f*
⟨*proof*⟩

lemma *continuous-on-subset*:
shows *continuous-on s f* $\Longrightarrow$ $t \subseteq s$ ==> *continuous-on t f*
⟨*proof*⟩

lemma *continuous-on-interior*:
shows *continuous-on s f* $\Longrightarrow$ $x \in$ *interior s* ==> *continuous* (*at x*) *f*
⟨*proof*⟩

lemma *continuous-on-eq*:
($\forall x \in s.\ f\ x = g\ x$) $\Longrightarrow$ *continuous-on s f* $\Longrightarrow$ *continuous-on s g*
⟨*proof*⟩

Characterization of various kinds of continuity in terms of sequences.

lemma *continuous-within-sequentially*:
fixes *f* :: *'a::metric-space* $\Rightarrow$ *'b::metric-space*
shows *continuous* (*at a within s*) *f* $\longleftrightarrow$
($\forall x.$ ($\forall n$::*nat*. *x n* $\in$ *s*) $\wedge$ (*x* ---> *a*) *sequentially*
--> ((*f o x*) ---> *f a*) *sequentially*) (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *continuous-at-sequentially*:
fixes *f* :: *'a::metric-space* $\Rightarrow$ *'b::metric-space*
shows *continuous* (*at a*) *f* $\longleftrightarrow$ ($\forall x.$ (*x* ---> *a*) *sequentially*
--> ((*f o x*) ---> *f a*) *sequentially*)
⟨*proof*⟩

lemma *continuous-on-sequentially*:
fixes *f* :: *'a::metric-space* $\Rightarrow$ *'b::metric-space*
shows *continuous-on s f* $\longleftrightarrow$
($\forall x.\ \forall a \in s.$ ($\forall n.\ x(n) \in s$) $\wedge$ (*x* ---> *a*) *sequentially*
--> ((*f o x*) ---> *f*(*a*)) *sequentially*) (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *uniformly-continuous-on-sequentially'*:
uniformly-continuous-on s f $\longleftrightarrow$ ($\forall x\ y.$ ($\forall n.\ x\ n \in s$) $\wedge$ ($\forall n.\ y\ n \in s$) $\wedge$
(($\lambda n.$ *dist* (*x n*) (*y n*)) ---> *0*) *sequentially*
$\longrightarrow$ (($\lambda n.$ *dist* (*f*(*x n*)) (*f*(*y n*))) ---> *0*) *sequentially*) (**is** *?lhs*
= *?rhs*)
⟨*proof*⟩

lemma *uniformly-continuous-on-sequentially*:
fixes *f* :: *'a::real-normed-vector* $\Rightarrow$ *'b::real-normed-vector*
shows *uniformly-continuous-on s f* $\longleftrightarrow$ ($\forall x\ y.$ ($\forall n.\ x\ n \in s$) $\wedge$ ($\forall n.\ y\ n \in s$) $\wedge$
(($\lambda n.$ *x n* − *y n*) ---> *0*) *sequentially*
$\longrightarrow$ (($\lambda n.$ *f*(*x n*) − *f*(*y n*)) ---> *0*) *sequentially*) (**is** *?lhs* =
?rhs)

⟨*proof*⟩

The usual transformation theorems.

lemma *continuous-transform-within*:
fixes $f\ g :: 'a::metric\text{-}space \Rightarrow 'b::topological\text{-}space$
assumes $0 < d\ x \in s\ \forall x' \in s.\ dist\ x'\ x < d\ \text{-->}\ f\ x' = g\ x'$
$continuous\ (at\ x\ within\ s)\ f$
shows $continuous\ (at\ x\ within\ s)\ g$
⟨*proof*⟩

lemma *continuous-transform-at*:
fixes $f\ g :: 'a::metric\text{-}space \Rightarrow 'b::topological\text{-}space$
assumes $0 < d\ \forall x'.\ dist\ x'\ x < d\ \text{-->}\ f\ x' = g\ x'$
$continuous\ (at\ x)\ f$
shows $continuous\ (at\ x)\ g$
⟨*proof*⟩

Combination results for pointwise continuity.

lemma *continuous-const*: $continuous\ net\ (\lambda x.\ c)$
⟨*proof*⟩

lemma *continuous-cmul*:
fixes $f :: 'a::t2\text{-}space \Rightarrow 'b::real\text{-}normed\text{-}vector$
shows $continuous\ net\ f \text{ ==> } continuous\ net\ (\lambda x.\ c *_R f\ x)$
⟨*proof*⟩

lemma *continuous-neg*:
fixes $f :: 'a::t2\text{-}space \Rightarrow 'b::real\text{-}normed\text{-}vector$
shows $continuous\ net\ f \text{ ==> } continuous\ net\ (\lambda x.\ -(f\ x))$
⟨*proof*⟩

lemma *continuous-add*:
fixes $f\ g :: 'a::t2\text{-}space \Rightarrow 'b::real\text{-}normed\text{-}vector$
shows $continuous\ net\ f \Longrightarrow continuous\ net\ g \Longrightarrow continuous\ net\ (\lambda x.\ f\ x + g\ x)$
⟨*proof*⟩

lemma *continuous-sub*:
fixes $f\ g :: 'a::t2\text{-}space \Rightarrow 'b::real\text{-}normed\text{-}vector$
shows $continuous\ net\ f \Longrightarrow continuous\ net\ g \Longrightarrow continuous\ net\ (\lambda x.\ f\ x - g\ x)$
⟨*proof*⟩

Same thing for setwise continuity.

lemma *continuous-on-const*:
$continuous\text{-}on\ s\ (\lambda x.\ c)$
⟨*proof*⟩

lemma *continuous-on-cmul*:
fixes f :: $'a$::*topological-space* ⇒ $'b$::*real-normed-vector*
shows *continuous-on* s f ⟹ *continuous-on* s $(\lambda x.\ c *_R (f\ x))$
⟨*proof*⟩

lemma *continuous-on-neg*:
fixes f :: $'a$::*topological-space* ⇒ $'b$::*real-normed-vector*
shows *continuous-on* s f ⟹ *continuous-on* s $(\lambda x.\ -\ f\ x)$
⟨*proof*⟩

lemma *continuous-on-add*:
fixes f g :: $'a$::*topological-space* ⇒ $'b$::*real-normed-vector*
shows *continuous-on* s f ⟹ *continuous-on* s g
⟹ *continuous-on* s $(\lambda x.\ f\ x + g\ x)$
⟨*proof*⟩

lemma *continuous-on-sub*:
fixes f g :: $'a$::*topological-space* ⇒ $'b$::*real-normed-vector*
shows *continuous-on* s f ⟹ *continuous-on* s g
⟹ *continuous-on* s $(\lambda x.\ f\ x - g\ x)$
⟨*proof*⟩

Same thing for uniform continuity, using sequential formulations.

lemma *uniformly-continuous-on-const*:
uniformly-continuous-on s $(\lambda x.\ c)$
⟨*proof*⟩

lemma *uniformly-continuous-on-cmul*:
fixes f :: $'a$::*metric-space* ⇒ $'b$::*real-normed-vector*
assumes *uniformly-continuous-on* s f
shows *uniformly-continuous-on* s $(\lambda x.\ c *_R f(x))$
⟨*proof*⟩

lemma *dist-minus*:
fixes x y :: $'a$::*real-normed-vector*
shows *dist* $(-\ x)$ $(-\ y)$ = *dist* x y
⟨*proof*⟩

lemma *uniformly-continuous-on-neg*:
fixes f :: $'a$::*metric-space* ⇒ $'b$::*real-normed-vector*
shows *uniformly-continuous-on* s f
==> *uniformly-continuous-on* s $(\lambda x.\ -(f\ x))$
⟨*proof*⟩

lemma *uniformly-continuous-on-add*:
fixes f g :: $'a$::*metric-space* ⇒ $'b$::*real-normed-vector*
assumes *uniformly-continuous-on* s f *uniformly-continuous-on* s g
shows *uniformly-continuous-on* s $(\lambda x.\ f\ x + g\ x)$
⟨*proof*⟩

lemma *uniformly-continuous-on-sub*:
fixes f :: $'a$::*metric-space* $\Rightarrow$ $'b$::*real-normed-vector*
shows *uniformly-continuous-on* s f $\Longrightarrow$ *uniformly-continuous-on* s g
==> *uniformly-continuous-on* s $(\lambda x.\ f\,x - g\,x)$
⟨*proof*⟩

Identity function is continuous in every sense.

lemma *continuous-within-id*:
continuous (*at* a *within* s) $(\lambda x.\ x)$
⟨*proof*⟩

lemma *continuous-at-id*:
continuous (*at* a) $(\lambda x.\ x)$
⟨*proof*⟩

lemma *continuous-on-id*:
continuous-on s $(\lambda x.\ x)$
⟨*proof*⟩

lemma *uniformly-continuous-on-id*:
uniformly-continuous-on s $(\lambda x.\ x)$
⟨*proof*⟩

Continuity of all kinds is preserved under composition.

lemma *continuous-within-topological*:
continuous (*at* x *within* s) f $\longleftrightarrow$
$(\forall B.$ *open* $B \longrightarrow f\,x \in B \longrightarrow$
$(\exists A.$ *open* $A \wedge x \in A \wedge (\forall y{\in}s.\ y \in A \longrightarrow f\,y \in B)))$
⟨*proof*⟩

lemma *continuous-within-compose*:
assumes *continuous* (*at* x *within* s) f
assumes *continuous* (*at* $(f\,x)$ *within* f ‘ s) g
shows *continuous* (*at* x *within* s) $(g\ o\ f)$
⟨*proof*⟩

lemma *continuous-at-compose*:
assumes *continuous* (*at* x) f *continuous* (*at* $(f\,x)$) g
shows *continuous* (*at* x) $(g\ o\ f)$
⟨*proof*⟩

lemma *continuous-on-compose*:
continuous-on s f $\Longrightarrow$ *continuous-on* $(f$ ‘ $s)$ g $\Longrightarrow$ *continuous-on* s $(g\ o\ f)$
⟨*proof*⟩

lemma *uniformly-continuous-on-compose*:
assumes *uniformly-continuous-on* s f *uniformly-continuous-on* $(f$ ‘ $s)$ g
shows *uniformly-continuous-on* s $(g\ o\ f)$

⟨*proof*⟩

Continuity in terms of open preimages.

lemma *continuous-at-open*:
shows $continuous\ (at\ x)\ f \longleftrightarrow (\forall\, t.\ open\ t \wedge f\, x \in t \dashrightarrow (\exists\, s.\ open\ s \wedge x \in s \wedge (\forall\, x' \in s.\ (f\, x') \in t)))$
⟨*proof*⟩

lemma *continuous-on-open*:
shows $continuous\text{-}on\ s\ f \longleftrightarrow$
$(\forall\, t.\ openin\ (subtopology\ euclidean\ (f\ `\ s))\ t$
$--> openin\ (subtopology\ euclidean\ s)\ \{x \in s.\ f\, x \in t\})$ (**is** $?lhs = ?rhs$)
⟨*proof*⟩

Similarly in terms of closed sets.

lemma *continuous-on-closed*:
shows $continuous\text{-}on\ s\ f \longleftrightarrow (\forall\, t.\ closedin\ (subtopology\ euclidean\ (f\ `\ s))\ t --> closedin\ (subtopology\ euclidean\ s)\ \{x \in s.\ f\, x \in t\})$ (**is** $?lhs = ?rhs$)
⟨*proof*⟩

Half-global and completely global cases.

lemma *continuous-open-in-preimage*:
assumes $continuous\text{-}on\ s\ f\ \ open\ t$
shows $openin\ (subtopology\ euclidean\ s)\ \{x \in s.\ f\, x \in t\}$
⟨*proof*⟩

lemma *continuous-closed-in-preimage*:
assumes $continuous\text{-}on\ s\ f\ \ closed\ t$
shows $closedin\ (subtopology\ euclidean\ s)\ \{x \in s.\ f\, x \in t\}$
⟨*proof*⟩

lemma *continuous-open-preimage*:
assumes $continuous\text{-}on\ s\ f\ open\ s\ open\ t$
shows $open\ \{x \in s.\ f\, x \in t\}$
⟨*proof*⟩

lemma *continuous-closed-preimage*:
assumes $continuous\text{-}on\ s\ f\ closed\ s\ closed\ t$
shows $closed\ \{x \in s.\ f\, x \in t\}$
⟨*proof*⟩

lemma *continuous-open-preimage-univ*:
shows $\forall\, x.\ continuous\ (at\ x)\ f \Longrightarrow open\ s \Longrightarrow open\ \{x.\ f\, x \in s\}$
⟨*proof*⟩

lemma *continuous-closed-preimage-univ*:
shows $(\forall\, x.\ continuous\ (at\ x)\ f) \Longrightarrow closed\ s ==> closed\ \{x.\ f\, x \in s\}$
⟨*proof*⟩

lemma *continuous-open-vimage*:
shows $\forall x.\ continuous\ (at\ x)\ f \Longrightarrow open\ s \Longrightarrow open\ (f\ -`\ s)$
⟨*proof*⟩

lemma *continuous-closed-vimage*:
shows $\forall x.\ continuous\ (at\ x)\ f \Longrightarrow closed\ s \Longrightarrow closed\ (f\ -`\ s)$
⟨*proof*⟩

lemma *interior-image-subset*:
assumes $\forall x.\ continuous\ (at\ x)\ f\ inj\ f$
shows $interior\ (f\ `\ s) \subseteq f\ `\ (interior\ s)$
⟨*proof*⟩

Equality of continuous functions on closure and related results.

lemma *continuous-closed-in-preimage-constant*:
fixes f :: - $\Rightarrow$ *'b::t1-space*
shows $continuous\text{-}on\ s\ f \text{ ==> } closedin\ (subtopology\ euclidean\ s)\ \{x \in s.\ f\ x = a\}$
⟨*proof*⟩

lemma *continuous-closed-preimage-constant*:
fixes f :: - $\Rightarrow$ *'b::t1-space*
shows $continuous\text{-}on\ s\ f \Longrightarrow closed\ s \text{ ==> } closed\ \{x \in s.\ f\ x = a\}$
⟨*proof*⟩

lemma *continuous-constant-on-closure*:
fixes f :: - $\Rightarrow$ *'b::t1-space*
assumes $continuous\text{-}on\ (closure\ s)\ f$
$\forall x \in s.\ f\ x = a$
shows $\forall x \in (closure\ s).\ f\ x = a$
⟨*proof*⟩

lemma *image-closure-subset*:
assumes $continuous\text{-}on\ (closure\ s)\ f\ \ closed\ t\ \ (f\ `\ s) \subseteq t$
shows $f\ `\ (closure\ s) \subseteq t$
⟨*proof*⟩

lemma *continuous-on-closure-norm-le*:
fixes f :: *'a::metric-space* $\Rightarrow$ *'b::real-normed-vector*
assumes $continuous\text{-}on\ (closure\ s)\ f\ \ \forall y \in s.\ norm(f\ y) \leq b\ \ x \in (closure\ s)$
shows $norm(f\ x) \leq b$
⟨*proof*⟩

Making a continuous function avoid some value in a neighbourhood.

lemma *continuous-within-avoid*:
fixes f :: *'a::metric-space* $\Rightarrow$ *'b::metric-space*
assumes $continuous\ (at\ x\ within\ s)\ f\ \ x \in s\ \ f\ x \neq a$
shows $\exists e>0.\ \forall y \in s.\ dist\ x\ y < e \text{ --> } f\ y \neq a$
⟨*proof*⟩

lemma *continuous-at-avoid*:
fixes $f :: 'a::metric\text{-}space \Rightarrow 'b::metric\text{-}space$
assumes $continuous\ (at\ x)\ f\ \ f\ x \neq a$
shows $\exists\, e>0.\ \forall\, y.\ dist\ x\ y < e \longrightarrow f\ y \neq a$
⟨*proof*⟩

lemma *continuous-on-avoid*:
fixes $f :: 'a::metric\text{-}space \Rightarrow 'b::metric\text{-}space$
assumes $continuous\text{-}on\ s\ f\ \ x \in s\ \ f\ x \neq a$
shows $\exists\, e>0.\ \forall\, y \in s.\ dist\ x\ y < e \longrightarrow f\ y \neq a$
⟨*proof*⟩

lemma *continuous-on-open-avoid*:
fixes $f :: 'a::metric\text{-}space \Rightarrow 'b::metric\text{-}space$
assumes $continuous\text{-}on\ s\ f\ \ open\ s\ \ x \in s\ \ f\ x \neq a$
shows $\exists\, e>0.\ \forall\, y.\ dist\ x\ y < e \longrightarrow f\ y \neq a$
⟨*proof*⟩

Proving a function is constant by proving open-ness of level set.

lemma *continuous-levelset-open-in-cases*:
fixes $f :: \text{-} \Rightarrow 'b::t1\text{-}space$
shows $connected\ s \Longrightarrow continuous\text{-}on\ s\ f \Longrightarrow$
$openin\ (subtopology\ euclidean\ s)\ \{x \in s.\ f\ x = a\}$
$==> (\forall\, x \in s.\ f\ x \neq a) \vee (\forall\, x \in s.\ f\ x = a)$
⟨*proof*⟩

lemma *continuous-levelset-open-in*:
fixes $f :: \text{-} \Rightarrow 'b::t1\text{-}space$
shows $connected\ s \Longrightarrow continuous\text{-}on\ s\ f \Longrightarrow$
$openin\ (subtopology\ euclidean\ s)\ \{x \in s.\ f\ x = a\} \Longrightarrow$
$(\exists\, x \in s.\ f\ x = a)\ \ ==> (\forall\, x \in s.\ f\ x = a)$
⟨*proof*⟩

lemma *continuous-levelset-open*:
fixes $f :: \text{-} \Rightarrow 'b::t1\text{-}space$
assumes $connected\ s\ \ continuous\text{-}on\ s\ f\ \ open\ \{x \in s.\ f\ x = a\}\ \ \exists\, x \in s.\ \ f\ x =$
a
shows $\forall\, x \in s.\ f\ x = a$
⟨*proof*⟩

Some arithmetical combinations (more to prove).

lemma *open-scaling*[*intro*]:
fixes $s :: 'a::real\text{-}normed\text{-}vector\ set$
assumes $c \neq 0\ \ open\ s$
shows $open((\lambda x.\ c *_R x)\ `\ s)$
⟨*proof*⟩

lemma *minus-image-eq-vimage*:

fixes A :: *'a::ab-group-add set*
shows $(\lambda x.\ -\ x)$ ' $A = (\lambda x.\ -\ x)$ −' A
⟨*proof*⟩

lemma *open-negations*:
fixes s :: *'a::real-normed-vector set*
shows *open s ==> open* $((\lambda\ x.\ -x)$ ' $s)$
⟨*proof*⟩

lemma *open-translation*:
fixes s :: *'a::real-normed-vector set*
assumes *open s* **shows** *open*$((\lambda x.\ a + x)$ ' $s)$
⟨*proof*⟩

lemma *open-affinity*:
fixes s :: *'a::real-normed-vector set*
assumes *open s* $c \neq 0$
shows *open* $((\lambda x.\ a + c *_R x)$ ' $s)$
⟨*proof*⟩

lemma *interior-translation*:
fixes s :: *'a::real-normed-vector set*
shows *interior* $((\lambda x.\ a + x)$ ' $s) = (\lambda x.\ a + x)$ ' (*interior* s)
⟨*proof*⟩

We can now extend limit compositions to consider the scalar multiplier.

lemma *continuous-vmul*:
fixes c :: *'a::metric-space* ⇒ *real* **and** v :: *'b::real-normed-vector*
shows *continuous net c ==> continuous net* $(\lambda x.\ c(x) *_R v)$
⟨*proof*⟩

lemma *continuous-mul*:
fixes c :: *'a::metric-space* ⇒ *real*
fixes f :: *'a::metric-space* ⇒ *'b::real-normed-vector*
shows *continuous net c* ⟹ *continuous net f*
==> *continuous net* $(\lambda x.\ c(x) *_R f\ x)$
⟨*proof*⟩

lemmas *continuous-intros = continuous-add continuous-vmul continuous-cmul continuous-const continuous-sub continuous-at-id continuous-within-id continuous-mul*

lemma *continuous-on-vmul*:
fixes c :: *'a::metric-space* ⇒ *real* **and** v :: *'b::real-normed-vector*
shows *continuous-on s c ==> continuous-on s* $(\lambda x.\ c(x) *_R v)$
⟨*proof*⟩

lemma *continuous-on-mul*:
fixes c :: *'a::metric-space* ⇒ *real*
fixes f :: *'a::metric-space* ⇒ *'b::real-normed-vector*

shows *continuous-on s c* $\Longrightarrow$ *continuous-on s f*
==> *continuous-on s* $(\lambda x.\ c(x) *_R f\ x)$
⟨*proof*⟩

lemmas *continuous-on-intros* = *continuous-on-add continuous-on-const continuous-on-id continuous-on-compose continuous-on-cmul continuous-on-neg continuous-on-sub uniformly-continuous-on-add uniformly-continuous-on-const uniformly-continuous-on-id uniformly-continuous-on-compose uniformly-continuous-on-cmul uniformly-continuous-on-neg uniformly-continuous-on-sub continuous-on-mul continuous-on-vmul*

And so we have continuity of inverse.

lemma *continuous-inv*:
fixes *f* :: *'a::metric-space* $\Rightarrow$ *real*
shows *continuous net f* $\Longrightarrow$ *f*(*netlimit net*) $\neq$ *0*
==> *continuous net* (*inverse o f*)
⟨*proof*⟩

lemma *continuous-at-within-inv*:
fixes *f* :: *'a::metric-space* $\Rightarrow$ *'b::real-normed-field*
assumes *continuous* (*at a within s*) *f f a* $\neq$ *0*
shows *continuous* (*at a within s*) (*inverse o f*)
⟨*proof*⟩

lemma *continuous-at-inv*:
fixes *f* :: *'a::metric-space* $\Rightarrow$ *'b::real-normed-field*
shows *continuous* (*at a*) *f* $\Longrightarrow$ *f a* $\neq$ *0*
==> *continuous* (*at a*) (*inverse o f*)
⟨*proof*⟩

Topological properties of linear functions.

lemma *linear-lim-0*:
assumes *bounded-linear f* **shows** (*f* ---> *0*) (*at* (*0*))
⟨*proof*⟩

lemma *linear-continuous-at*:
assumes *bounded-linear f* **shows** *continuous* (*at a*) *f*
⟨*proof*⟩

lemma *linear-continuous-within*:
shows *bounded-linear f* ==> *continuous* (*at x within s*) *f*
⟨*proof*⟩

lemma *linear-continuous-on*:
shows *bounded-linear f* ==> *continuous-on s f*
⟨*proof*⟩

Also bilinear functions, in composition form.

lemma *bilinear-continuous-at-compose*:

shows *continuous* (*at* x) $f \Longrightarrow$ *continuous* (*at* x) $g \Longrightarrow$ *bounded-bilinear* h
==> *continuous* (*at* x) ($\lambda x.$ h (f x) (g x))
⟨*proof*⟩

lemma *bilinear-continuous-within-compose*:
shows *continuous* (*at* x *within* s) $f \Longrightarrow$ *continuous* (*at* x *within* s) $g \Longrightarrow$ *bounded-bilinear* h
==> *continuous* (*at* x *within* s) ($\lambda x.$ h (f x) (g x))
⟨*proof*⟩

lemma *bilinear-continuous-on-compose*:
shows *continuous-on* s $f \Longrightarrow$ *continuous-on* s $g \Longrightarrow$ *bounded-bilinear* h
==> *continuous-on* s ($\lambda x.$ h (f x) (g x))
⟨*proof*⟩

Preservation of compactness and connectedness under continuous function.

lemma *compact-continuous-image*:
assumes *continuous-on* s f *compact* s
shows *compact*(f ‘ s)
⟨*proof*⟩

lemma *connected-continuous-image*:
assumes *continuous-on* s f *connected* s
shows *connected*(f ‘ s)
⟨*proof*⟩

Continuity implies uniform continuity on a compact domain.

lemma *compact-uniformly-continuous*:
assumes *continuous-on* s f *compact* s
shows *uniformly-continuous-on* s f
⟨*proof*⟩

Continuity of inverse function on compact domain.

lemma *continuous-on-inverse*:
fixes f :: $'a$::*heine-borel* $\Rightarrow$ $'b$::*heine-borel*

assumes *continuous-on* s f *compact* s $\forall x \in s.\ g\ (f\ x) = x$
shows *continuous-on* (f ‘ s) g
⟨*proof*⟩

A uniformly convergent limit of continuous functions is continuous.

lemma *norm-triangle-lt*:
fixes x y :: $'a$::*real-normed-vector*
shows *norm* x + *norm* $y < e \Longrightarrow$ *norm* $(x + y) < e$
⟨*proof*⟩

lemma *continuous-uniform-limit*:
fixes f :: $'a \Rightarrow$ $'b$::*metric-space* $\Rightarrow$ $'c$::*real-normed-vector*

assumes $\neg$ (*trivial-limit net*) *eventually* ($\lambda n.$ *continuous-on s* (*f n*)) *net*
$\forall\, e>0.$ *eventually* ($\lambda n. \forall\, x \in s.$ *norm*(*f n x* $-$ *g x*) $<$ *e*) *net*
shows *continuous-on s g*
⟨*proof*⟩

16.21 Topological stuff lifted from and dropped to R

lemma *open-real*:
fixes *s* :: *real set* **shows**
open s $\longleftrightarrow$
($\forall\, x \in s.\ \exists\, e>0.\ \forall\, x'.$ *abs*($x' - x$) $< e$ $-->$ $x' \in s$) (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *islimpt-approachable-real*:
fixes *s* :: *real set*
shows *x islimpt s* $\longleftrightarrow$ ($\forall\, e>0.\ \exists\, x' \in s.\ x' \neq x \wedge$ *abs*($x' - x$) $< e$)
⟨*proof*⟩

lemma *closed-real*:
fixes *s* :: *real set*
shows *closed s* $\longleftrightarrow$
($\forall\, x.$ ($\forall\, e>0.\ \exists\, x' \in s.\ x' \neq x \wedge$ *abs*($x' - x$) $< e$)
$-->$ $x \in s$)
⟨*proof*⟩

lemma *continuous-at-real-range*:
fixes *f* :: *'a::real-normed-vector* $\Rightarrow$ *real*
shows *continuous* (*at x*) *f* $\longleftrightarrow$ ($\forall\, e>0.\ \exists\, d>0.$
$\forall\, x'.$ *norm*($x' - x$) $< d$ $-->$ *abs*($f\,x' - f\,x$) $< e$)
⟨*proof*⟩

lemma *continuous-on-real-range*:
fixes *f* :: *'a::real-normed-vector* $\Rightarrow$ *real*
shows *continuous-on s f* $\longleftrightarrow$ ($\forall\, x \in s.\ \forall\, e>0.\ \exists\, d>0.$ ($\forall\, x' \in s.$ *norm*($x' - x$) $< d$ $-->$ *abs*($f\,x' - f\,x$) $< e$))
⟨*proof*⟩

lemma *continuous-at-norm*: *continuous* (*at x*) *norm*
⟨*proof*⟩

lemma *continuous-on-norm*: *continuous-on s norm*
⟨*proof*⟩

lemma *continuous-at-component*: *continuous* (*at a*) ($\lambda x.\ x\ \$\ i$)
⟨*proof*⟩

lemma *continuous-on-component*: *continuous-on s* ($\lambda x.\ x\ \$\ i$)
⟨*proof*⟩

lemma *continuous-at-infnorm*: *continuous* (*at* x) *infnorm*
⟨*proof*⟩

Hence some handy theorems on distance, diameter etc. of/from a set.

lemma *compact-attains-sup*:
fixes s :: *real set*
assumes *compact* s $s \neq \{\}$
shows $\exists\, x \in s.\ \forall\, y \in s.\ y \leq x$
⟨*proof*⟩

lemma *Inf*:
fixes S :: *real set*
shows $S \neq \{\}$ ==> ($\exists\, b.\ b$ <=* S) ==> *isGlb UNIV* S (*Inf* S)
⟨*proof*⟩

lemma *compact-attains-inf*:
fixes s :: *real set*
assumes *compact* s $s \neq \{\}$ **shows** $\exists\, x \in s.\ \forall\, y \in s.\ x \leq y$
⟨*proof*⟩

lemma *continuous-attains-sup*:
fixes f :: *'a::metric-space* $\Rightarrow$ *real*
shows *compact* $s \Longrightarrow s \neq \{\} \Longrightarrow$ *continuous-on* $s\ f$
==> ($\exists\, x \in s.\ \forall\, y \in s.\ f\,y \leq f\,x$)
⟨*proof*⟩

lemma *continuous-attains-inf*:
fixes f :: *'a::metric-space* $\Rightarrow$ *real*
shows *compact* $s \Longrightarrow s \neq \{\} \Longrightarrow$ *continuous-on* $s\ f$
$\Longrightarrow (\exists\, x \in s.\ \forall\, y \in s.\ f\,x \leq f\,y)$
⟨*proof*⟩

lemma *distance-attains-sup*:
assumes *compact* s $s \neq \{\}$
shows $\exists\, x \in s.\ \forall\, y \in s.$ *dist* $a\ y \leq$ *dist* $a\ x$
⟨*proof*⟩

For *minimal* distance, we only need closure, not compactness.

lemma *distance-attains-inf*:
fixes a :: *'a::heine-borel*
assumes *closed* s $s \neq \{\}$
shows $\exists\, x \in s.\ \forall\, y \in s.$ *dist* $a\ x \leq$ *dist* $a\ y$
⟨*proof*⟩

16.22 Pasted sets

lemma *bounded-Times*:
assumes *bounded* s *bounded* t **shows** *bounded* ($s \times t$)
⟨*proof*⟩

lemma *mem-Times-iff*: $x \in A \times B \longleftrightarrow fst\ x \in A \wedge snd\ x \in B$
⟨*proof*⟩

lemma *compact-Times*: $compact\ s \Longrightarrow compact\ t \Longrightarrow compact\ (s \times t)$
⟨*proof*⟩

Hence some useful properties follow quite easily.

lemma *compact-scaling*:
fixes s :: $'a$::*real-normed-vector set*
assumes *compact s* **shows** $compact\ ((\lambda x.\ c *_R x)\ `\ s)$
⟨*proof*⟩

lemma *compact-negations*:
fixes s :: $'a$::*real-normed-vector set*
assumes *compact s* **shows** $compact\ ((\lambda x.\ -x)\ `\ s)$
⟨*proof*⟩

lemma *compact-sums*:
fixes $s\ t$ :: $'a$::*real-normed-vector set*
assumes *compact s compact t* **shows** $compact\ \{x + y \mid x\ y.\ x \in s \wedge y \in t\}$
⟨*proof*⟩

lemma *compact-differences*:
fixes $s\ t$ :: $'a$::*real-normed-vector set*
assumes *compact s compact t* **shows** $compact\ \{x - y \mid x\ y.\ x \in s \wedge y \in t\}$
⟨*proof*⟩

lemma *compact-translation*:
fixes s :: $'a$::*real-normed-vector set*
assumes *compact s* **shows** $compact\ ((\lambda x.\ a + x)\ `\ s)$
⟨*proof*⟩

lemma *compact-affinity*:
fixes s :: $'a$::*real-normed-vector set*
assumes *compact s* **shows** $compact\ ((\lambda x.\ a + c *_R x)\ `\ s)$
⟨*proof*⟩

Hence we get the following.

lemma *compact-sup-maxdistance*:
fixes s :: $'a$::*real-normed-vector set*
assumes *compact s* $s \neq \{\}$
shows $\exists x{\in}s.\ \exists y{\in}s.\ \forall u{\in}s.\ \forall v{\in}s.\ norm(u - v) \le norm(x - y)$
⟨*proof*⟩

We can state this in terms of diameter of a set.

definition $diameter\ s = (if\ s = \{\}\ then\ 0{::}real\ else\ Sup\ \{norm(x - y) \mid x\ y.\ x \in s \wedge y \in s\})$

lemma *diameter-bounded*:
assumes *bounded s*
shows $\forall\, x{\in}s.\ \forall\, y{\in}s.\ norm(x - y) \leq diameter\ s$
$\forall\, d{>}0.\ d < diameter\ s \ {-}{-}{>}\ (\exists\, x{\in}s.\ \exists\, y{\in}s.\ norm(x - y) > d)$
⟨*proof*⟩

lemma *diameter-bounded-bound*:
$bounded\ s \Longrightarrow x \in s \Longrightarrow y \in s \ {=}{=}{>}\ norm(x - y) \leq diameter\ s$
⟨*proof*⟩

lemma *diameter-compact-attained*:
fixes s :: $'a$::*real-normed-vector set*
assumes *compact s* $s \neq \{\}$
shows $\exists\, x{\in}s.\ \exists\, y{\in}s.\ (norm(x - y) = diameter\ s)$
⟨*proof*⟩

Related results with closure as the conclusion.

lemma *closed-scaling*:
fixes s :: $'a$::*real-normed-vector set*
assumes *closed s* **shows** *closed* $((\lambda x.\ c *_R x)$ ‘ $s)$
⟨*proof*⟩

lemma *closed-negations*:
fixes s :: $'a$::*real-normed-vector set*
assumes *closed s* **shows** *closed* $((\lambda x.\ -x)$ ‘ $s)$
⟨*proof*⟩

lemma *compact-closed-sums*:
fixes s :: $'a$::*real-normed-vector set*
assumes *compact s* *closed t* **shows** *closed* $\{x + y \mid x\ y.\ x \in s \wedge y \in t\}$
⟨*proof*⟩

lemma *closed-compact-sums*:
fixes $s\ t$ :: $'a$::*real-normed-vector set*
assumes *closed s* *compact t*
shows *closed* $\{x + y \mid x\ y.\ x \in s \wedge y \in t\}$
⟨*proof*⟩

lemma *compact-closed-differences*:
fixes $s\ t$ :: $'a$::*real-normed-vector set*
assumes *compact s* *closed t*
shows *closed* $\{x - y \mid x\ y.\ x \in s \wedge y \in t\}$
⟨*proof*⟩

lemma *closed-compact-differences*:
fixes $s\ t$ :: $'a$::*real-normed-vector set*
assumes *closed s compact t*
shows *closed* $\{x - y \mid x\ y.\ x \in s \wedge y \in t\}$

⟨*proof*⟩

lemma *closed-translation*:
fixes $a :: 'a::real\text{-}normed\text{-}vector$
assumes *closed s* **shows** *closed* $((\lambda x.\ a + x)$ ‘ $s)$
⟨*proof*⟩

lemma *translation-Compl*:
fixes $a :: 'a::ab\text{-}group\text{-}add$
shows $(\lambda x.\ a + x)$ ‘ $(-\ t) = -\ ((\lambda x.\ a + x)$ ‘ $t)$
⟨*proof*⟩

lemma *translation-UNIV*:
fixes $a :: 'a::ab\text{-}group\text{-}add$ **shows** *range* $(\lambda x.\ a + x) = UNIV$
⟨*proof*⟩

lemma *translation-diff*:
fixes $a :: 'a::ab\text{-}group\text{-}add$
shows $(\lambda x.\ a + x)$ ‘ $(s - t) = ((\lambda x.\ a + x)$ ‘ $s) - ((\lambda x.\ a + x)$ ‘ $t)$
⟨*proof*⟩

lemma *closure-translation*:
fixes $a :: 'a::real\text{-}normed\text{-}vector$
shows *closure* $((\lambda x.\ a + x)$ ‘ $s) = (\lambda x.\ a + x)$ ‘ $(closure\ s)$
⟨*proof*⟩

lemma *frontier-translation*:
fixes $a :: 'a::real\text{-}normed\text{-}vector$
shows *frontier*$((\lambda x.\ a + x)$ ‘ $s) = (\lambda x.\ a + x)$ ‘ $(frontier\ s)$
⟨*proof*⟩

16.23 Separation between points and sets.

lemma *separate-point-closed*:
fixes $s :: 'a::heine\text{-}borel\ set$
shows *closed s* $\Longrightarrow a \notin s$ ==> $(\exists\, d>0.\ \forall\, x\in s.\ d \leq dist\ a\ x)$
⟨*proof*⟩

lemma *separate-compact-closed*:
fixes $s\ t :: 'a::\{heine\text{-}borel,\ real\text{-}normed\text{-}vector\}\ set$

assumes *compact s* **and** *closed t* **and** $s \cap t = \{\}$
shows $\exists\, d>0.\ \forall\, x\in s.\ \forall\, y\in t.\ d \leq dist\ x\ y$
⟨*proof*⟩

lemma *separate-closed-compact*:
fixes $s\ t :: 'a::\{heine\text{-}borel,\ real\text{-}normed\text{-}vector\}\ set$
assumes *closed s* **and** *compact t* **and** $s \cap t = \{\}$
shows $\exists\, d>0.\ \forall\, x\in s.\ \forall\, y\in t.\ d \leq dist\ x\ y$

⟨*proof*⟩

16.24 Intervals

lemma *interval*: **fixes** a :: $'a$::*ord*$\hat{}'n$ **shows**
$\{a <..< b\} = \{x::'a\hat{}'n.\ \forall i.\ a\$i < x\$i \wedge x\$i < b\$i\}$ **and**
$\{a\ ..\ b\} = \{x::'a\hat{}'n.\ \forall i.\ a\$i \le x\$i \wedge x\$i \le b\$i\}$
⟨*proof*⟩

lemma *mem-interval*: **fixes** a :: $'a$::*ord*$\hat{}'n$ **shows**
$x \in \{a<..<b\} \longleftrightarrow (\forall i.\ a\$i < x\$i \wedge x\$i < b\$i)$
$x \in \{a\ ..\ b\} \longleftrightarrow (\forall i.\ a\$i \le x\$i \wedge x\$i \le b\$i)$
⟨*proof*⟩

lemma *interval-eq-empty*: **fixes** a :: *real*$\hat{}'n$ **shows**
$(\{a <..< b\} = \{\} \longleftrightarrow (\exists i.\ b\$i \le a\$i))$ (**is** *?th1*) **and**
$(\{a\ ..\ b\} = \{\} \longleftrightarrow (\exists i.\ b\$i < a\$i))$ (**is** *?th2*)
⟨*proof*⟩

lemma *interval-ne-empty*: **fixes** a :: *real*$\hat{}'n$ **shows**
$\{a\ ..\ b\} \neq \{\} \longleftrightarrow (\forall i.\ a\$i \le b\$i)$ **and**
$\{a <..< b\} \neq \{\} \longleftrightarrow (\forall i.\ a\$i < b\$i)$
⟨*proof*⟩

lemma *subset-interval-imp*: **fixes** a :: *real*$\hat{}'n$ **shows**
$(\forall i.\ a\$i \le c\$i \wedge d\$i \le b\$i) \Longrightarrow \{c\ ..\ d\} \subseteq \{a\ ..\ b\}$ **and**
$(\forall i.\ a\$i < c\$i \wedge d\$i < b\$i) \Longrightarrow \{c\ ..\ d\} \subseteq \{a<..<b\}$ **and**
$(\forall i.\ a\$i \le c\$i \wedge d\$i \le b\$i) \Longrightarrow \{c<..<d\} \subseteq \{a\ ..\ b\}$ **and**
$(\forall i.\ a\$i \le c\$i \wedge d\$i \le b\$i) \Longrightarrow \{c<..<d\} \subseteq \{a<..<b\}$
⟨*proof*⟩

lemma *interval-sing*: **fixes** a :: $'a$::*linorder*$\hat{}'n$ **shows**
$\{a\ ..\ a\} = \{a\} \wedge \{a<..<a\} = \{\}$
⟨*proof*⟩

lemma *interval-open-subset-closed*: **fixes** a :: $'a$::*preorder*$\hat{}'n$ **shows**
$\{a<..<b\} \subseteq \{a\ ..\ b\}$
⟨*proof*⟩

lemma *subset-interval*: **fixes** a :: *real*$\hat{}'n$ **shows**
$\{c\ ..\ d\} \subseteq \{a\ ..\ b\} \longleftrightarrow (\forall i.\ c\$i \le d\$i) \dashrightarrow (\forall i.\ a\$i \le c\$i \wedge d\$i \le b\$i)$ (**is** *?th1*) **and**
$\{c\ ..\ d\} \subseteq \{a<..<b\} \longleftrightarrow (\forall i.\ c\$i \le d\$i) \dashrightarrow (\forall i.\ a\$i < c\$i \wedge d\$i < b\$i)$ (**is** *?th2*) **and**
$\{c<..<d\} \subseteq \{a\ ..\ b\} \longleftrightarrow (\forall i.\ c\$i < d\$i) \dashrightarrow (\forall i.\ a\$i \le c\$i \wedge d\$i \le b\$i)$ (**is** *?th3*) **and**
$\{c<..<d\} \subseteq \{a<..<b\} \longleftrightarrow (\forall i.\ c\$i < d\$i) \dashrightarrow (\forall i.\ a\$i \le c\$i \wedge d\$i \le b\$i)$ (**is** *?th4*)
⟨*proof*⟩

lemma *disjoint-interval*: **fixes** $a::real^{\wedge\prime}n$ **shows**
$\{a\ ..\ b\} \cap \{c\ ..\ d\} = \{\} \longleftrightarrow (\exists\, i.\ (b\$i < a\$i \vee d\$i < c\$i \vee b\$i < c\$i \vee d\$i < a\$i))$ (**is** *?th1*) **and**
$\{a\ ..\ b\} \cap \{c<..<d\} = \{\} \longleftrightarrow (\exists\, i.\ (b\$i < a\$i \vee d\$i \leq c\$i \vee b\$i \leq c\$i \vee d\$i \leq a\$i))$ (**is** *?th2*) **and**
$\{a<..<b\} \cap \{c\ ..\ d\} = \{\} \longleftrightarrow (\exists\, i.\ (b\$i \leq a\$i \vee d\$i < c\$i \vee b\$i \leq c\$i \vee d\$i \leq a\$i))$ (**is** *?th3*) **and**
$\{a<..<b\} \cap \{c<..<d\} = \{\} \longleftrightarrow (\exists\, i.\ (b\$i \leq a\$i \vee d\$i \leq c\$i \vee b\$i \leq c\$i \vee d\$i \leq a\$i))$ (**is** *?th4*)
⟨*proof*⟩

lemma *inter-interval*: **fixes** $a :: 'a::linorder^{\wedge\prime}n$ **shows**
$\{a\ ..\ b\} \cap \{c\ ..\ d\} = \{(\chi\ i.\ max\ (a\$i)\ (c\$i))\ ..\ (\chi\ i.\ min\ (b\$i)\ (d\$i))\}$
⟨*proof*⟩

lemma *open-interval-lemma*: **fixes** $x :: real$ **shows**
$a < x \Longrightarrow x < b ==> (\exists\, d{>}0.\ \forall\, x'.\ abs(x' - x) < d --> a < x' \wedge x' < b)$
⟨*proof*⟩

lemma *open-interval*[*intro*]: **fixes** $a :: real^{\wedge\prime}n$ **shows** *open* $\{a<..<b\}$
⟨*proof*⟩

lemma *open-interval-real*[*intro*]: **fixes** $a :: real$ **shows** *open* $\{a<..<b\}$
⟨*proof*⟩

lemma *closed-interval*[*intro*]: **fixes** $a :: real^{\wedge\prime}n$ **shows** *closed* $\{a\ ..\ b\}$
⟨*proof*⟩

lemma *interior-closed-interval*[*intro*]: **fixes** $a :: real^{\wedge\prime}n$ **shows**
interior $\{a\ ..\ b\} = \{a<..<b\}$ (**is** *?L = ?R*)
⟨*proof*⟩

lemma *bounded-closed-interval*: **fixes** $a :: real^{\wedge\prime}n$ **shows**
bounded $\{a\ ..\ b\}$
⟨*proof*⟩

lemma *bounded-interval*: **fixes** $a :: real^{\wedge\prime}n$ **shows**
bounded $\{a\ ..\ b\} \wedge$ *bounded* $\{a<..<b\}$
⟨*proof*⟩

lemma *not-interval-univ*: **fixes** $a :: real^{\wedge\prime}n$ **shows**
$(\{a\ ..\ b\} \neq UNIV) \wedge (\{a<..<b\} \neq UNIV)$
⟨*proof*⟩

lemma *compact-interval*: **fixes** $a :: real^{\wedge\prime}n$ **shows**
compact $\{a\ ..\ b\}$

⟨*proof*⟩

lemma *open-interval-midpoint*: **fixes** a :: *real*^'n
assumes $\{a<..<b\} \neq \{\}$ **shows** $((1/2) *_R (a + b)) \in \{a<..<b\}$
⟨*proof*⟩

lemma *open-closed-interval-convex*: **fixes** x :: *real*^'n
assumes x:$x \in \{a<..<b\}$ **and** y:$y \in \{a .. b\}$ **and** e:$0 < e$ $e \leq 1$
shows $(e *_R x + (1 - e) *_R y) \in \{a<..<b\}$
⟨*proof*⟩

lemma *closure-open-interval*: **fixes** a :: *real*^'n
assumes $\{a<..<b\} \neq \{\}$
shows *closure* $\{a<..<b\} = \{a .. b\}$
⟨*proof*⟩

lemma *bounded-subset-open-interval-symmetric*: **fixes** s::(*real*^'n) *set*
assumes *bounded* s **shows** $\exists\, a.\ s \subseteq \{-a<..<a\}$
⟨*proof*⟩

lemma *bounded-subset-open-interval*:
fixes s :: (*real* ^ 'n) *set*
shows *bounded* s ==> $(\exists\, a\ b.\ s \subseteq \{a<..<b\})$
⟨*proof*⟩

lemma *bounded-subset-closed-interval-symmetric*:
fixes s :: (*real* ^ 'n) *set*
assumes *bounded* s **shows** $\exists\, a.\ s \subseteq \{-a .. a\}$
⟨*proof*⟩

lemma *bounded-subset-closed-interval*:
fixes s :: (*real* ^ 'n) *set*
shows *bounded* s ==> $(\exists\, a\ b.\ s \subseteq \{a .. b\})$
⟨*proof*⟩

lemma *frontier-closed-interval*:
fixes a b :: *real* ^ -
shows *frontier* $\{a .. b\} = \{a .. b\} - \{a<..<b\}$
⟨*proof*⟩

lemma *frontier-open-interval*:
fixes a b :: *real* ^ -
shows *frontier* $\{a<..<b\}$ = (*if* $\{a<..<b\} = \{\}$ *then* $\{\}$ *else* $\{a .. b\} - \{a<..<b\}$)
⟨*proof*⟩

lemma *inter-interval-mixed-eq-empty*: **fixes** a :: *real*^'n
assumes $\{c<..<d\} \neq \{\}$ **shows** $\{a<..<b\} \cap \{c .. d\} = \{\} \longleftrightarrow \{a<..<b\} \cap \{c<..<d\} = \{\}$
⟨*proof*⟩

lemma *closed-interval-left*: **fixes** $b::real\hat{\ }'n$
  shows *closed* $\{x::real\hat{\ }'n.\ \forall\, i.\ x\$i \le b\$i\}$
⟨*proof*⟩

lemma *closed-interval-right*: **fixes** $a::real\hat{\ }'n$
  shows *closed* $\{x::real\hat{\ }'n.\ \forall\, i.\ a\$i \le x\$i\}$
⟨*proof*⟩

Intervals in general, including infinite and mixtures of open and closed.

definition *is-interval* $s \longleftrightarrow (\forall\, a{\in}s.\ \forall\, b{\in}s.\ \forall\, x.\ (\forall\, i.\ ((a\$i \le x\$i \wedge x\$i \le b\$i) \vee (b\$i \le x\$i \wedge x\$i \le a\$i)))\ \longrightarrow x \in s)$

lemma *is-interval-interval*: *is-interval* $\{a\ ..\ b::real\hat{\ }'n\}$ (**is** *?th1*) *is-interval* $\{a{<}..{<}b\}$ (**is** *?th2*) ⟨*proof*⟩

lemma *is-interval-empty*:
 is-interval $\{\}$
  ⟨*proof*⟩

lemma *is-interval-univ*:
 is-interval UNIV
  ⟨*proof*⟩

16.25 Closure of halfspaces and hyperplanes.

lemma *Lim-inner*:
  assumes $(f\ {-}{-}{-}{>}\ l)$ *net* **shows** $((\lambda y.\ inner\ a\ (f\ y))\ {-}{-}{-}{>}\ inner\ a\ l)$ *net*
  ⟨*proof*⟩

lemma *continuous-at-inner*: *continuous* $(at\ x)$ $(inner\ a)$
  ⟨*proof*⟩

lemma *continuous-on-inner*:
  fixes $s :: 'a::real\text{-}inner\ set$
  shows *continuous-on* s $(inner\ a)$
  ⟨*proof*⟩

lemma *closed-halfspace-le*: *closed* $\{x.\ inner\ a\ x \le b\}$
⟨*proof*⟩

lemma *closed-halfspace-ge*: *closed* $\{x.\ inner\ a\ x \ge b\}$
  ⟨*proof*⟩

lemma *closed-hyperplane*: *closed* $\{x.\ inner\ a\ x = b\}$
⟨*proof*⟩

lemma *closed-halfspace-component-le*:
shows *closed* $\{x::real^\frown'n.\ x\$i \leq a\}$
⟨*proof*⟩

lemma *closed-halfspace-component-ge*:
shows *closed* $\{x::real^\frown'n.\ x\$i \geq a\}$
⟨*proof*⟩

Openness of halfspaces.

lemma *open-halfspace-lt*: *open* $\{x.\ inner\ a\ x < b\}$
⟨*proof*⟩

lemma *open-halfspace-gt*: *open* $\{x.\ inner\ a\ x > b\}$
⟨*proof*⟩

lemma *open-halfspace-component-lt*:
shows *open* $\{x::real^\frown'n.\ x\$i < a\}$
⟨*proof*⟩

lemma *open-halfspace-component-gt*:
shows *open* $\{x::real^\frown'n.\ x\$i\ > a\}$
⟨*proof*⟩

This gives a simple derivation of limit component bounds.

lemma *Lim-component-le*: **fixes** f :: $'a \Rightarrow real^\frown'n$
assumes $(f\ {-}{-}{-}{>}\ l)\ net$ $\neg$ $(trivial\text{-}limit\ net)$ *eventually* $(\lambda x.\ f(x)\$i \leq b)\ net$
shows $l\$i \leq b$
⟨*proof*⟩

lemma *Lim-component-ge*: **fixes** f :: $'a \Rightarrow real^\frown'n$
assumes $(f\ {-}{-}{-}{>}\ l)\ net$ $\neg$ $(trivial\text{-}limit\ net)$ *eventually* $(\lambda x.\ b \leq (f\,x)\$i)\ net$
shows $b \leq l\$i$
⟨*proof*⟩

lemma *Lim-component-eq*: **fixes** f :: $'a \Rightarrow real^\frown'n$
assumes *net*:$(f\ {-}{-}{-}{>}\ l)\ net$ ~$(trivial\text{-}limit\ net)$ **and** *ev*:*eventually* $(\lambda x.\ f(x)\$i = b)\ net$
shows $l\$i = b$
⟨*proof*⟩

Limits relative to a union.

lemma *eventually-within-Un*:
eventually P $(net\ within\ (s \cup t)) \longleftrightarrow$
eventually P $(net\ within\ s) \wedge$ *eventually* P $(net\ within\ t)$
⟨*proof*⟩

lemma *Lim-within-union*:
$(f\ {-}{-}{-}{>}\ l)\ (net\ within\ (s \cup t)) \longleftrightarrow$

$(f \text{ ---> } l)$ *(net within s)* $\wedge$ $(f \text{ ---> } l)$ *(net within t)*
⟨*proof*⟩

lemma *Lim-topological*:
$(f \text{ ---> } l)$ *net* $\longleftrightarrow$
trivial-limit net $\vee$
$(\forall S.$ *open* $S \longrightarrow l \in S \longrightarrow$ *eventually* $(\lambda x.\ f\ x \in S)$ *net*$)$
⟨*proof*⟩

lemma *continuous-on-union*:
assumes *closed s closed t continuous-on s f continuous-on t f*
shows *continuous-on* $(s \cup t)$ *f*
⟨*proof*⟩

lemma *continuous-on-cases*:
assumes *closed s closed t continuous-on s f continuous-on t g*
$\forall x.\ (x{\in}s \wedge \neg\ P\ x) \vee (x \in t \wedge P\ x) \longrightarrow f\ x = g\ x$
shows *continuous-on* $(s \cup t)$ $(\lambda x.$ *if* $P\ x$ *then* $f\ x$ *else* $g\ x)$
⟨*proof*⟩

Some more convenient intermediate-value theorem formulations.

lemma *connected-ivt-hyperplane*:
assumes *connected* s $x \in s$ $y \in s$ *inner* $a\ x \le b$ $b \le$ *inner* $a\ y$
shows $\exists z \in s.$ *inner* $a\ z = b$
⟨*proof*⟩

lemma *connected-ivt-component*: **fixes** *x::real^'n* **shows**
connected $s \Longrightarrow x \in s \Longrightarrow y \in s \Longrightarrow x\$k \le a \Longrightarrow a \le y\$k \Longrightarrow (\exists z{\in}s.\ \ z\$k = a)$
⟨*proof*⟩

16.26 Homeomorphisms

definition *homeomorphism s t f g* $\equiv$
$(\forall x{\in}s.\ (g(f\ x) = x)) \wedge (f\ `\ s = t) \wedge$ *continuous-on s f* $\wedge$
$(\forall y{\in}t.\ (f(g\ y) = y)) \wedge (g\ `\ t = s) \wedge$ *continuous-on t g*

definition
homeomorphic :: *'a::metric-space set* $\Rightarrow$ *'b::metric-space set* $\Rightarrow$ *bool*
(**infixr** *homeomorphic 60*) **where**
homeomorphic-def: *s homeomorphic t* $\equiv$ $(\exists f\ g.$ *homeomorphism s t f g*$)$

lemma *homeomorphic-refl*: *s homeomorphic s*
⟨*proof*⟩

lemma *homeomorphic-sym*:
s homeomorphic t $\longleftrightarrow$ *t homeomorphic s*
⟨*proof*⟩

lemma *homeomorphic-trans*:
assumes *s homeomorphic t t homeomorphic u* **shows** *s homeomorphic u*
⟨*proof*⟩

lemma *homeomorphic-minimal*:
s *homeomorphic* $t \longleftrightarrow$
$(\exists f\ g.\ (\forall x{\in}s.\ f(x) \in t \wedge (g(f(x)) = x)) \wedge$
$(\forall y{\in}t.\ g(y) \in s \wedge (f(g(y)) = y)) \wedge$
continuous-on $s\ f\ \wedge$ *continuous-on* $t\ g)$
⟨*proof*⟩

Relatively weak hypotheses if a set is compact.

lemma *homeomorphism-compact*:
fixes f :: $'a$::*heine-borel* $\Rightarrow$ $'b$::*heine-borel*

assumes *compact s continuous-on s f* f ‘ $s = t$ *inj-on f s*
shows $\exists g.$ *homeomorphism s t f g*
⟨*proof*⟩

lemma *homeomorphic-compact*:
fixes f :: $'a$::*heine-borel* $\Rightarrow$ $'b$::*heine-borel*

shows *compact s* $\Longrightarrow$ *continuous-on s f* $\Longrightarrow$ $(f$ ‘ $s = t)$ $\Longrightarrow$ *inj-on f s*
$\Longrightarrow$ *s homeomorphic t*
⟨*proof*⟩

Preservation of topological properties.

lemma *homeomorphic-compactness*:
s homeomorphic t ==> (*compact s* $\longleftrightarrow$ *compact t*)
⟨*proof*⟩

Results on translation, scaling etc.

lemma *homeomorphic-scaling*:
fixes s :: $'a$::*real-normed-vector set*
assumes $c \neq 0$ **shows** *s homeomorphic* $((\lambda x.\ c *_R x)$ ‘ $s)$
⟨*proof*⟩

lemma *homeomorphic-translation*:
fixes s :: $'a$::*real-normed-vector set*
shows *s homeomorphic* $((\lambda x.\ a + x)$ ‘ $s)$
⟨*proof*⟩

lemma *homeomorphic-affinity*:
fixes s :: $'a$::*real-normed-vector set*
assumes $c \neq 0$ **shows** *s homeomorphic* $((\lambda x.\ a + c *_R x)$ ‘ $s)$
⟨*proof*⟩

lemma *homeomorphic-balls*:
fixes $a\ b$ ::$'a$::*real-normed-vector*

assumes $0 < d$ $0 < e$
shows $(ball\ a\ d)$ *homeomorphic* $(ball\ b\ e)$ (**is** $?th$)
$(cball\ a\ d)$ *homeomorphic* $(cball\ b\ e)$ (**is** $?cth$)
⟨*proof*⟩

"Isometry" (up to constant bounds) of injective linear map etc.

lemma *cauchy-isometric*:
fixes $x :: nat \Rightarrow real\ \hat{}\ 'n$
assumes $e{:}0 < e$ **and** $s{:}subspace\ s$ **and** $f{:}bounded\text{-}linear\ f$ **and** $normf{:}\forall x{\in}s.\ norm(f\ x) \geq e * norm(x)$ **and** $xs{:}\forall n{::}nat.\ x\ n \in s$ **and** $cf{:}Cauchy(f\ o\ x)$
shows *Cauchy x*
⟨*proof*⟩

lemma *complete-isometric-image*:
fixes $f :: real\ \hat{}\ \text{-} \Rightarrow real\ \hat{}\ \text{-}$
assumes $0 < e$ **and** $s{:}subspace\ s$ **and** $f{:}bounded\text{-}linear\ f$ **and** $normf{:}\forall x{\in}s.\ norm(f\ x) \geq e * norm(x)$ **and** $cs{:}complete\ s$
shows $complete(f\ `\ s)$
⟨*proof*⟩

lemma *dist-0-norm*:
fixes $x :: 'a{::}real\text{-}normed\text{-}vector$
shows $dist\ 0\ x = norm\ x$
⟨*proof*⟩

lemma *injective-imp-isometric*: **fixes** $f{::}real\hat{}'m \Rightarrow real\hat{}'n$
assumes $s{:}closed\ s$ $subspace\ s$ **and** $f{:}bounded\text{-}linear\ f$ $\forall x{\in}s.\ (f\ x = 0) \longrightarrow (x = 0)$
shows $\exists e{>}0.\ \forall x{\in}s.\ norm\ (f\ x) \geq e * norm(x)$
⟨*proof*⟩

lemma *closed-injective-image-subspace*:
fixes $f :: real\ \hat{}\ \text{-} \Rightarrow real\ \hat{}\ \text{-}$
assumes $subspace\ s$ $bounded\text{-}linear\ f$ $\forall x{\in}s.\ f\ x = 0 \dashrightarrow x = 0$ $closed\ s$
shows $closed(f\ `\ s)$
⟨*proof*⟩

16.27 Some properties of a canonical subspace.

lemma *subspace-substandard*:
$subspace\ \{x{::}real\hat{}\text{-}.\ (\forall i.\ P\ i \longrightarrow x\$i = 0)\}$
⟨*proof*⟩

lemma *closed-substandard*:
$closed\ \{x{::}real\hat{}'n.\ \forall i.\ P\ i \dashrightarrow x\$i = 0\}$ (**is** $closed\ ?A$)
⟨*proof*⟩

lemma *dim-substandard*:
shows $dim\ \{x{::}real\hat{}'n.\ \forall i.\ i \notin d \longrightarrow x\$i = 0\} = card\ d$ (**is** $dim\ ?A = \text{-}$)

⟨*proof*⟩

Hence closure and completeness of all subspaces.

lemma *closed-subspace-lemma*: $n \leq card\ (UNIV::'n::finite\ set) \Longrightarrow \exists\, A::'n\ set.\ card\ A = n$
⟨*proof*⟩

lemma *closed-subspace*: **fixes** s::(*real*ˆ′*n*) *set*
assumes *subspace s* **shows** *closed s*
⟨*proof*⟩

lemma *complete-subspace*:
fixes s :: (*real* ˆ -) *set* **shows** *subspace s* ==> *complete s*
⟨*proof*⟩

lemma *dim-closure*:
fixes s :: (*real* ˆ -) *set*
shows $dim(closure\ s) = dim\ s$ (**is** $?dc = ?d$)
⟨*proof*⟩

16.28 Affine transformations of intervals

lemma *affinity-inverses*:
assumes $m0$: $m \neq (0::'a::field)$
shows $(\lambda x.\ m *s\ x + c)\ o\ (\lambda x.\ inverse(m) *s\ x + (-(inverse(m) *s\ c))) = id$
$(\lambda x.\ inverse(m) *s\ x + (-(inverse(m) *s\ c)))\ o\ (\lambda x.\ m *s\ x + c) = id$
⟨*proof*⟩

lemma *real-affinity-le*:
$0 < (m::'a::linordered\text{-}field)$ ==> $(m * x + c \leq y \longleftrightarrow x \leq inverse(m) * y + -(c\ /\ m))$
⟨*proof*⟩

lemma *real-le-affinity*:
$0 < (m::'a::linordered\text{-}field)$ ==> $(y \leq m * x + c \longleftrightarrow inverse(m) * y + -(c\ /\ m) \leq x)$
⟨*proof*⟩

lemma *real-affinity-lt*:
$0 < (m::'a::linordered\text{-}field)$ ==> $(m * x + c < y \longleftrightarrow x < inverse(m) * y + -(c\ /\ m))$
⟨*proof*⟩

lemma *real-lt-affinity*:
$0 < (m::'a::linordered\text{-}field)$ ==> $(y < m * x + c \longleftrightarrow inverse(m) * y + -(c\ /\ m) < x)$
⟨*proof*⟩

lemma *real-affinity-eq*:

$(m::'a::linordered\text{-}field) \neq 0 ==> (m * x + c = y \longleftrightarrow x = inverse(m) * y + -(c / m))$
⟨proof⟩

lemma *real-eq-affinity*:
$(m::'a::linordered\text{-}field) \neq 0 ==> (y = m * x + c \longleftrightarrow inverse(m) * y + -(c / m) = x)$
⟨proof⟩

lemma *vector-affinity-eq*:
assumes $m0$: $(m::'a::field) \neq 0$
shows $m *s\ x + c = y \longleftrightarrow x = inverse\ m *s\ y + -(inverse\ m *s\ c)$
⟨proof⟩

lemma *vector-eq-affinity*:
$(m::'a::field) \neq 0 ==> (y = m *s\ x + c \longleftrightarrow inverse(m) *s\ y + -(inverse(m) *s\ c) = x)$
⟨proof⟩

lemma *image-affinity-interval*: **fixes** $m::real$
fixes $a\ b\ c :: real\hat{}'n$
shows $(\lambda x.\ m *_R x + c)$ ‘ $\{a .. b\} =$
$(if\ \{a .. b\} = \{\}\ then\ \{\}$
$else\ (if\ 0 \leq m\ then\ \{m *_R a + c .. m *_R b + c\}$
$else\ \{m *_R b + c .. m *_R a + c\}))$
⟨proof⟩

lemma *image-smult-interval*:$(\lambda x.\ m *_R (x::real\hat{}'n))$ ‘ $\{a..b\} =$
$(if\ \{a..b\} = \{\}\ then\ \{\}\ else\ if\ 0 \leq m\ then\ \{m *_R a..m *_R b\}\ else\ \{m *_R b..m *_R a\})$
⟨proof⟩

16.29 Banach fixed point theorem (not really topological...)

lemma *banach-fix*:
assumes s:*complete s* $s \neq \{\}$ **and** c:$0 \leq c$ $c < 1$ **and** f:$(f$ ‘ $s) \subseteq s$ **and**
lipschitz:$\forall x \in s.\ \forall y \in s.\ dist\ (f\ x)\ (f\ y) \leq c * dist\ x\ y$
shows $\exists!\ x \in s.\ (f\ x = x)$
⟨proof⟩

16.30 Edelstein fixed point theorem.

lemma *edelstein-fix*:
fixes s :: *'a::real-normed-vector set*
assumes s:*compact s* $s \neq \{\}$ **and** gs:$(g$ ‘ $s) \subseteq s$
and *dist*:$\forall x \in s.\ \forall y \in s.\ x \neq y \longrightarrow dist\ (g\ x)\ (g\ y) < dist\ x\ y$
shows $\exists!\ x \in s.\ g\ x = x$
⟨proof⟩

end

17 Vec1: Vectors of size 1, 2, or 3

theory *Vec1*
imports *Topology-Euclidean-Space*
begin

Some common special cases.

lemma *forall-1*[*simp*]: $(\forall\, i::1.\ P\ i) \longleftrightarrow P\ 1$
⟨*proof*⟩

lemma *ex-1*[*simp*]: $(\exists\, x::1.\ P\ x) \longleftrightarrow P\ 1$
⟨*proof*⟩

lemma *exhaust-2*:
fixes $x :: 2$ **shows** $x = 1 \lor x = 2$
⟨*proof*⟩

lemma *forall-2*: $(\forall\, i::2.\ P\ i) \longleftrightarrow P\ 1 \land P\ 2$
⟨*proof*⟩

lemma *exhaust-3*:
fixes $x :: 3$ **shows** $x = 1 \lor x = 2 \lor x = 3$
⟨*proof*⟩

lemma *forall-3*: $(\forall\, i::3.\ P\ i) \longleftrightarrow P\ 1 \land P\ 2 \land P\ 3$
⟨*proof*⟩

lemma *UNIV-1* [*simp*]: $UNIV = \{1::1\}$
⟨*proof*⟩

lemma *UNIV-2*: $UNIV = \{1::2,\ 2::2\}$
⟨*proof*⟩

lemma *UNIV-3*: $UNIV = \{1::3,\ 2::3,\ 3::3\}$
⟨*proof*⟩

lemma *setsum-1*: $setsum\ f\ (UNIV::1\ set) = f\ 1$
⟨*proof*⟩

lemma *setsum-2*: $setsum\ f\ (UNIV::2\ set) = f\ 1 + f\ 2$
⟨*proof*⟩

lemma *setsum-3*: $setsum\ f\ (UNIV::3\ set) = f\ 1 + f\ 2 + f\ 3$
⟨*proof*⟩

instantiation *num1* :: *cart-one* **begin**
instance ⟨*proof*⟩ **end**

abbreviation $vec1 :: 'a \Rightarrow 'a \mathbin{\hat{}} 1$ **where** $vec1\ x \equiv vec\ x$

abbreviation $dest\text{-}vec1 :: 'a \mathbin{\hat{}} 1 \Rightarrow 'a$
where $dest\text{-}vec1\ x \equiv (x\$1)$

lemma *vec1-component*[*simp*]: $(vec1\ x)\$1 = x$
⟨*proof*⟩

lemma *vec1-dest-vec1*: $vec1(dest\text{-}vec1\ x) = x$ $dest\text{-}vec1(vec1\ y) = y$
⟨*proof*⟩

declare *vec1-dest-vec1*(*1*) [*simp*]

lemma *forall-vec1*: $(\forall x.\ P\ x) \longleftrightarrow (\forall x.\ P\ (vec1\ x))$
⟨*proof*⟩

lemma *exists-vec1*: $(\exists x.\ P\ x) \longleftrightarrow (\exists x.\ P(vec1\ x))$
⟨*proof*⟩

lemma *vec1-eq*[*simp*]: $vec1\ x = vec1\ y \longleftrightarrow x = y$
⟨*proof*⟩

lemma *dest-vec1-eq*[*simp*]: $dest\text{-}vec1\ x = dest\text{-}vec1\ y \longleftrightarrow x = y$
⟨*proof*⟩

17.1 The collapse of the general concepts to dimension one.

lemma *vector-one*: $(x::'a \mathbin{\hat{}} 1) = (\chi\ i.\ (x\$1))$
⟨*proof*⟩

lemma *forall-one*: $(\forall (x::'a \mathbin{\hat{}} 1).\ P\ x) \longleftrightarrow (\forall x.\ P(\chi\ i.\ x))$
⟨*proof*⟩

lemma *norm-vector-1*: $norm\ (x :: \text{-} \mathbin{\hat{}} 1) = norm\ (x\$1)$
⟨*proof*⟩

lemma *norm-real*: $norm(x::real \mathbin{\hat{}} 1) = abs(x\$1)$
⟨*proof*⟩

lemma *dist-real*: $dist(x::real \mathbin{\hat{}} 1)\ y = abs((x\$1) - (y\$1))$
⟨*proof*⟩

17.2 Explicit vector construction from lists.

primrec $from\text{-}nat :: nat \Rightarrow 'a::\{monoid\text{-}add,one\}$
where $from\text{-}nat\ 0 = 0 \mid from\text{-}nat\ (Suc\ n) = 1 + from\text{-}nat\ n$

lemma *from-nat* [*simp*]: *from-nat* = *of-nat*
⟨*proof*⟩

primrec
list-fun :: *nat* ⇒ - *list* ⇒ - ⇒ -
where
list-fun n [] = ($\lambda x.$ *0*)
| *list-fun n* (*x* # *xs*) = *fun-upd* (*list-fun* (*Suc n*) *xs*) (*from-nat n*) *x*

definition *vector l* = (χ *i*. *list-fun 1 l i*)

lemma *vector-1*: (*vector*[*x*]) \$*1* = *x*
⟨*proof*⟩

lemma *vector-2*:
(*vector*[*x*,*y*]) \$*1* = *x*
(*vector*[*x*,*y*] :: '*a*^*2*)\$*2* = (*y*::'*a*::*zero*)
⟨*proof*⟩

lemma *vector-3*:
(*vector* [*x*,*y*,*z*] ::('*a*::*zero*)^*3*)\$*1* = *x*
(*vector* [*x*,*y*,*z*] ::('*a*::*zero*)^*3*)\$*2* = *y*
(*vector* [*x*,*y*,*z*] ::('*a*::*zero*)^*3*)\$*3* = *z*
⟨*proof*⟩

lemma *forall-vector-1*: (∀ *v*::'*a*::*zero*^*1*. *P v*) ⟷ (∀ *x*. *P*(*vector*[*x*]))
⟨*proof*⟩

lemma *forall-vector-2*: (∀ *v*::'*a*::*zero*^*2*. *P v*) ⟷ (∀ *x y*. *P*(*vector*[*x*, *y*]))
⟨*proof*⟩

lemma *forall-vector-3*: (∀ *v*::'*a*::*zero*^*3*. *P v*) ⟷ (∀ *x y z*. *P*(*vector*[*x*, *y*, *z*]))
⟨*proof*⟩

lemma *range-vec1*[*simp*]:*range vec1* = *UNIV* ⟨*proof*⟩

lemma *dest-vec1-lambda*: *dest-vec1*(χ *i*. *x i*) = *x 1*
⟨*proof*⟩

lemma *dest-vec1-vec*: *dest-vec1*(*vec x*) = *x*
⟨*proof*⟩

lemma *dest-vec1-sum*: **assumes** *fS*: *finite S*
shows *dest-vec1*(*setsum f S*) = *setsum* (*dest-vec1 o f*) *S*
⟨*proof*⟩

lemma *norm-vec1* [*simp*]: *norm*(*vec1 x*) = *abs*(*x*)
⟨*proof*⟩

lemma *dist-vec1*: *dist*(*vec1 x*) (*vec1 y*) = *abs*(*x* − *y*)
⟨*proof*⟩
lemma *abs-dest-vec1*: *norm x* = |*dest-vec1 x*|
⟨*proof*⟩

lemmas *vec1-dest-vec1-simps* = *forall-vec1 vec-add*[*THEN sym*] *dist-vec1 vec-sub*[*THEN sym*] *vec1-dest-vec1 norm-vec1 vector-smult-component*
vec1-eq vec-cmul[*THEN sym*] *smult-conv-scaleR*[*THEN sym*] *o-def dist-real-def norm-vec1 real-norm-def*

lemma *bounded-linear-vec1*:*bounded-linear* (*vec1*::*real*⇒*real*ˆ*1*)
⟨*proof*⟩

lemma *linear-vmul-dest-vec1*:
fixes *f*:: *real*ˆ- ⇒ *real*ˆ*1*
shows *linear f* ⟹ *linear* (*λx*. *dest-vec1*(*f x*) ∗*s v*)
⟨*proof*⟩

lemma *linear-from-scalars*:
assumes *lf*: *linear* (*f*::*real*ˆ*1* ⇒ *real*ˆ-)
shows *f* = (*λx*. *dest-vec1 x* ∗*s column 1* (*matrix f*))
⟨*proof*⟩

lemma *linear-to-scalars*: **assumes** *lf*: *linear* (*f*::*real* ˆ′*n* ⇒ *real*ˆ*1*)
shows *f* = (*λx*. *vec1*(*row 1* (*matrix f*) · *x*))
⟨*proof*⟩

lemma *dest-vec1-eq-0*: *dest-vec1 x* = *0* ⟷ *x* = *0*
⟨*proof*⟩

lemma *setsum-scalars*: **assumes** *fS*: *finite S*
shows *setsum f S* = *vec1* (*setsum* (*dest-vec1 o f*) *S*)
⟨*proof*⟩

lemma *dest-vec1-wlog-le*: (⋀(*x*::′*a*::*linorder* ˆ *1*) *y*. *P x y* ⟷ *P y x*) ⟹ (⋀*x y*. *dest-vec1 x* <= *dest-vec1 y* ==> *P x y*) ⟹ *P x y*
⟨*proof*⟩

Lifting and dropping

lemma *continuous-on-o-dest-vec1*: **fixes** *f*::*real* ⇒ ′*a*::*real-normed-vector*
assumes *continuous-on* {*a*..*b*::*real*} *f* **shows** *continuous-on* {*vec1 a*..*vec1 b*} (*f o dest-vec1*)
⟨*proof*⟩

lemma *continuous-on-o-vec1*: **fixes** *f*::*real*ˆ*1* ⇒ ′*a*::*real-normed-vector*
assumes *continuous-on* {*a*..*b*} *f* **shows** *continuous-on* {*dest-vec1 a*..*dest-vec1 b*} (*f o vec1*)
⟨*proof*⟩

lemma *continuous-on-vec1*:*continuous-on* A (*vec1*::*real*$\Rightarrow$*real*^*1*)
⟨*proof*⟩

lemma *mem-interval-1*: **fixes** x :: *real*^*1* **shows**
($x \in \{a\ ..\ b\} \longleftrightarrow$ *dest-vec1* $a \leq$ *dest-vec1* $x \wedge$ *dest-vec1* $x \leq$ *dest-vec1* b)
($x \in \{a<..<b\} \longleftrightarrow$ *dest-vec1* $a <$ *dest-vec1* $x \wedge$ *dest-vec1* $x <$ *dest-vec1* b)
⟨*proof*⟩

lemma *vec1-interval*:**fixes** a::*real* **shows**
vec1 ‘ $\{a\ ..\ b\} = \{$*vec1* a .. *vec1* $b\}$
vec1 ‘ $\{a<..<b\} = \{$*vec1* a<..<*vec1* $b\}$
⟨*proof*⟩

lemma *interval-cases-1*: **fixes** x :: *real*^*1* **shows**
$x \in \{a\ ..\ b\}$ ==> $x \in \{a<..<b\} \vee (x = a) \vee (x = b)$
⟨*proof*⟩

lemma *in-interval-1*: **fixes** x :: *real*^*1* **shows**
($x \in \{a\ ..\ b\} \longleftrightarrow$ *dest-vec1* $a \leq$ *dest-vec1* $x \wedge$ *dest-vec1* $x \leq$ *dest-vec1* b) $\wedge$
($x \in \{a<..<b\} \longleftrightarrow$ *dest-vec1* $a <$ *dest-vec1* $x \wedge$ *dest-vec1* $x <$ *dest-vec1* b)
⟨*proof*⟩

lemma *interval-eq-empty-1*: **fixes** a :: *real*^*1* **shows**
$\{a\ ..\ b\} = \{\} \longleftrightarrow$ *dest-vec1* $b <$ *dest-vec1* a
$\{a<..<b\} = \{\} \longleftrightarrow$ *dest-vec1* $b \leq$ *dest-vec1* a
⟨*proof*⟩

lemma *subset-interval-1*: **fixes** a :: *real*^*1* **shows**
($\{a\ ..\ b\} \subseteq \{c\ ..\ d\} \longleftrightarrow$ *dest-vec1* $b <$ *dest-vec1* $a \vee$
dest-vec1 $c \leq$ *dest-vec1* $a \wedge$ *dest-vec1* $a \leq$ *dest-vec1* $b \wedge$ *dest-vec1* b
$\leq$ *dest-vec1* d)
($\{a\ ..\ b\} \subseteq \{c<..<d\} \longleftrightarrow$ *dest-vec1* $b <$ *dest-vec1* $a \vee$
dest-vec1 $c <$ *dest-vec1* $a \wedge$ *dest-vec1* $a \leq$ *dest-vec1* $b \wedge$ *dest-vec1* b
$<$ *dest-vec1* d)
($\{a<..<b\} \subseteq \{c\ ..\ d\} \longleftrightarrow$ *dest-vec1* $b \leq$ *dest-vec1* $a \vee$
dest-vec1 $c \leq$ *dest-vec1* $a \wedge$ *dest-vec1* $a <$ *dest-vec1* $b \wedge$ *dest-vec1* b
$\leq$ *dest-vec1* d)
($\{a<..<b\} \subseteq \{c<..<d\} \longleftrightarrow$ *dest-vec1* $b \leq$ *dest-vec1* $a \vee$
dest-vec1 $c \leq$ *dest-vec1* $a \wedge$ *dest-vec1* $a <$ *dest-vec1* $b \wedge$ *dest-vec1* b
$\leq$ *dest-vec1* d)
⟨*proof*⟩

lemma *eq-interval-1*: **fixes** a :: *real*^*1* **shows**
$\{a\ ..\ b\} = \{c\ ..\ d\} \longleftrightarrow$
dest-vec1 $b <$ *dest-vec1* $a \wedge$ *dest-vec1* $d <$ *dest-vec1* $c \vee$
dest-vec1 $a =$ *dest-vec1* $c \wedge$ *dest-vec1* $b =$ *dest-vec1* d

$\langle proof \rangle$

lemma *disjoint-interval-1*: **fixes** a :: *real^1* **shows**
$\{a .. b\} \cap \{c .. d\} = \{\} \longleftrightarrow$ *dest-vec1* $b <$ *dest-vec1* $a \vee$ *dest-vec1* $d <$ *dest-vec1* $c \vee$ *dest-vec1* $b <$ *dest-vec1* $c \vee$ *dest-vec1* $d <$ *dest-vec1* a
$\{a .. b\} \cap \{c<..<d\} = \{\} \longleftrightarrow$ *dest-vec1* $b <$ *dest-vec1* $a \vee$ *dest-vec1* $d \leq$ *dest-vec1* $c \vee$ *dest-vec1* $b \leq$ *dest-vec1* $c \vee$ *dest-vec1* $d \leq$ *dest-vec1* a
$\{a<..<b\} \cap \{c .. d\} = \{\} \longleftrightarrow$ *dest-vec1* $b \leq$ *dest-vec1* $a \vee$ *dest-vec1* $d <$ *dest-vec1* $c \vee$ *dest-vec1* $b \leq$ *dest-vec1* $c \vee$ *dest-vec1* $d \leq$ *dest-vec1* a
$\{a<..<b\} \cap \{c<..<d\} = \{\} \longleftrightarrow$ *dest-vec1* $b \leq$ *dest-vec1* $a \vee$ *dest-vec1* $d \leq$ *dest-vec1* $c \vee$ *dest-vec1* $b \leq$ *dest-vec1* $c \vee$ *dest-vec1* $d \leq$ *dest-vec1* a
$\langle proof \rangle$

lemma *open-closed-interval-1*: **fixes** a :: *real^1* **shows**
$\{a<..<b\} = \{a .. b\} - \{a, b\}$
$\langle proof \rangle$

lemma *closed-open-interval-1*: *dest-vec1* $(a$::*real^1*$) \leq$ *dest-vec1* b ==> $\{a .. b\} = \{a<..<b\} \cup \{a,b\}$
$\langle proof \rangle$

lemma *Lim-drop-le*: **fixes** f :: *'a* $\Rightarrow$ *real^1* **shows**
$(f$ ---> $l)$ *net* $\Longrightarrow$ ~(*trivial-limit net*) $\Longrightarrow$ *eventually* $(\lambda x.$ *dest-vec1* $(f\,x) \leq b)$ *net* ==> *dest-vec1* $l \leq b$
$\langle proof \rangle$

lemma *Lim-drop-ge*: **fixes** f :: *'a* $\Rightarrow$ *real^1* **shows**
$(f$ ---> $l)$ *net* $\Longrightarrow$ ~(*trivial-limit net*) $\Longrightarrow$ *eventually* $(\lambda x.\ b \leq$ *dest-vec1* $(f\,x))$ *net* ==> $b \leq$ *dest-vec1* l
$\langle proof \rangle$

Also more convenient formulations of monotone convergence.

lemma *bounded-increasing-convergent*: **fixes** s::*nat* $\Rightarrow$ *real^1*
assumes *bounded* $\{s\ n|\ n$::*nat. True*$\}$ $\forall n.$ *dest-vec1*$(s\ n) \leq$ *dest-vec1*$(s(Suc\ n))$
shows $\exists l.\ (s$ ---> $l)$ *sequentially*
$\langle proof \rangle$

lemma *dest-vec1-simps*[*simp*]: **fixes** a::*real^1*
shows $a\$1 = 0 \longleftrightarrow a = 0$
$a \leq b \longleftrightarrow$ *dest-vec1* $a \leq$ *dest-vec1* b *dest-vec1* $(1$::*real^1*$) = 1$
$\langle proof \rangle$

lemma *dest-vec1-inverval*:
dest-vec1 ` $\{a .. b\} = \{$*dest-vec1* a .. *dest-vec1* $b\}$
dest-vec1 ` $\{a<.. b\} = \{$*dest-vec1* a<.. *dest-vec1* $b\}$
dest-vec1 ` $\{a ..<b\} = \{$*dest-vec1* a ..<*dest-vec1* $b\}$
dest-vec1 ` $\{a<..<b\} = \{$*dest-vec1* a<..<*dest-vec1* $b\}$
$\langle proof \rangle$

lemma *dest-vec1-setsum*: **assumes** *finite S*
shows *dest-vec1 (setsum f S) = setsum (λx. dest-vec1 (f x)) S*
⟨*proof*⟩

lemma *open-dest-vec1-vimage*: *open S ⟹ open (dest-vec1 −' S)*
⟨*proof*⟩

lemma *tendsto-dest-vec1* [*tendsto-intros*]:
(f −−−> l) net ⟹ ((λx. dest-vec1 (f x)) −−−> dest-vec1 l) net
⟨*proof*⟩

lemma *continuous-dest-vec1*: *continuous net f ⟹ continuous net (λx. dest-vec1 (f x))*
⟨*proof*⟩

lemma *forall-dest-vec1*: *(∀x. P x) ⟷ (∀x. P(dest-vec1 x))*
⟨*proof*⟩

lemma *forall-of-dest-vec1*: *(∀v. P (λx. dest-vec1 (v x))) ⟷ (∀x. P x)*
⟨*proof*⟩

lemma *forall-of-dest-vec1'*: *(∀v. P (dest-vec1 v)) ⟷ (∀x. P x)*
⟨*proof*⟩

lemma *dist-vec1-0*[*simp*]: *dist(vec1 (x::real)) 0 = norm x* ⟨*proof*⟩

lemma *bounded-linear-vec1-dest-vec1*: **fixes** *f::real ⇒ real*
shows *linear (vec1 ∘ f ∘ dest-vec1) = bounded-linear f* (**is** *?l = ?r*) ⟨*proof*⟩

lemma *vec1-le*[*simp*]:**fixes** *a::real* **shows** *vec1 a ≤ vec1 b ⟷ a ≤ b*
⟨*proof*⟩
lemma *vec1-less*[*simp*]:**fixes** *a::real* **shows** *vec1 a < vec1 b ⟷ a < b*
⟨*proof*⟩

end

18 Determinants: Traces, Determinant of square matrices and some properties

theory *Determinants*
imports *Euclidean-Space Permutations Vec1*
begin

18.1 First some facts about products

lemma *setprod-insert-eq*: *finite A ⟹ setprod f (insert a A) = (if a ∈ A then setprod f A else f a ∗ setprod f A)*

⟨*proof*⟩

lemma *setprod-add-split*:
assumes *mn*: $(m::nat) <= n + 1$
shows $setprod\ f\ \{m..\ n+p\} = setprod\ f\ \{m\ ..\ n\} * setprod\ f\ \{n+1..n+p\}$
⟨*proof*⟩

lemma *setprod-offset*: $setprod\ f\ \{(m::nat) + p\ ..\ n + p\} = setprod\ (\lambda i.\ f\ (i + p))\ \{m..n\}$
⟨*proof*⟩

lemma *setprod-singleton*: $setprod\ f\ \{x\} = f\ x$ ⟨*proof*⟩

lemma *setprod-singleton-nat-seg*: $setprod\ f\ \{n..n\} = f\ (n::'a::order)$ ⟨*proof*⟩

lemma *setprod-numseg*: $setprod\ f\ \{m..0\} = (if\ m=0\ then\ f\ 0\ else\ 1)$
$setprod\ f\ \{m\ ..\ Suc\ n\} = (if\ m \leq Suc\ n\ then\ f\ (Suc\ n) * setprod\ f\ \{m..n\}$
$else\ setprod\ f\ \{m..n\})$
⟨*proof*⟩

lemma *setprod-le*: **assumes** *fS*: *finite S* **and** *fg*: $\forall x{\in}S.\ f\ x \geq 0 \wedge f\ x \leq (g\ x :: 'a::linordered\text{-}idom)$
shows $setprod\ f\ S \leq setprod\ g\ S$
⟨*proof*⟩

lemma *setprod-inversef*: $finite\ A \Longrightarrow setprod\ (inverse \circ f)\ A = (inverse\ (setprod\ f\ A) :: 'a::\ field\text{-}inverse\text{-}zero)$
⟨*proof*⟩

lemma *setprod-le-1*: **assumes** *fS*: *finite S* **and** *f*: $\forall x{\in}S.\ f\ x \geq 0 \wedge f\ x \leq (1::'a::linordered\text{-}idom)$
shows $setprod\ f\ S \leq 1$
⟨*proof*⟩

18.2 Trace

definition $trace :: 'a::semiring\text{-}1\hat{\ }'n\hat{\ }'n \Rightarrow 'a$ **where**
$trace\ A = setsum\ (\lambda i.\ ((A\$i)\$i))\ (UNIV::'n\ set)$

lemma *trace-0*: $trace(mat\ 0) = 0$
⟨*proof*⟩

lemma *trace-I*: $trace(mat\ 1 :: 'a::semiring\text{-}1\hat{\ }'n\hat{\ }'n) = of\text{-}nat(CARD('n))$
⟨*proof*⟩

lemma *trace-add*: $trace\ ((A::'a::comm\text{-}semiring\text{-}1\hat{\ }'n\hat{\ }'n) + B) = trace\ A + trace\ B$
⟨*proof*⟩

lemma *trace-sub*: *trace* $((A::'a::comm\text{-}ring\text{-}1\hat{}'n\hat{}'n) - B) = trace\ A - trace\ B$
⟨*proof*⟩

lemma *trace-mul-sym*:*trace* $((A::'a::comm\text{-}semiring\text{-}1\hat{}'n\hat{}'n) ** B) = trace\ (B**A)$
⟨*proof*⟩

definition *det*:: $'a::comm\text{-}ring\text{-}1\hat{}'n\hat{}'n \Rightarrow 'a$ **where**
$det\ A = setsum\ (\lambda p.\ of\text{-}int\ (sign\ p) * setprod\ (\lambda i.\ A\$i\$p\ i)\ (UNIV :: 'n\ set))$
$\{p.\ p\ permutes\ (UNIV :: 'n\ set)\}$

lemma *setprod-permute*:
assumes *p*: *p permutes S*
shows $setprod\ f\ S = setprod\ (f\ o\ p)\ S$
⟨*proof*⟩

lemma *setproduct-permute-nat-interval*: $p\ permutes\ \{m::nat\ ..\ n\} ==> setprod\ f$
$\{m..n\} = setprod\ (f\ o\ p)\ \{m..n\}$
⟨*proof*⟩

lemma *det-transpose*: $det\ (transpose\ A) = det\ (A::'a::comm\text{-}ring\text{-}1\ \hat{}'n\hat{}'n)$
⟨*proof*⟩

lemma *det-lowerdiagonal*:
fixes $A :: 'a::comm\text{-}ring\text{-}1\hat{}('n::\{finite,wellorder\})\hat{}('n::\{finite,wellorder\})$
assumes *ld*: $\bigwedge i\ j.\ i < j \Longrightarrow A\$i\$j = 0$
shows $det\ A = setprod\ (\lambda i.\ A\$i\$i)\ (UNIV:: 'n\ set)$
⟨*proof*⟩

lemma *det-upperdiagonal*:
fixes $A :: 'a::comm\text{-}ring\text{-}1\hat{}'n::\{finite,wellorder\}\hat{}'n::\{finite,wellorder\}$
assumes *ld*: $\bigwedge i\ j.\ i > j \Longrightarrow A\$i\$j = 0$
shows $det\ A = setprod\ (\lambda i.\ A\$i\$i)\ (UNIV:: 'n\ set)$
⟨*proof*⟩

lemma *det-diagonal*:
fixes $A :: 'a::comm\text{-}ring\text{-}1\hat{}'n\hat{}'n$

assumes *ld*: $\bigwedge i\ j.\ i \neq j \Longrightarrow A\$i\$j = 0$
shows $det\ A = setprod\ (\lambda i.\ A\$i\$i)\ (UNIV::'n\ set)$
$\langle proof \rangle$

lemma *det-I*: $det\ (mat\ 1 :: 'a::comm\text{-}ring\text{-}1\hat{\ }'n\hat{\ }'n) = 1$
$\langle proof \rangle$

lemma *det-0*: $det\ (mat\ 0 :: 'a::comm\text{-}ring\text{-}1\hat{\ }'n\hat{\ }'n) = 0$
$\langle proof \rangle$

lemma *det-permute-rows*:
fixes $A :: 'a::comm\text{-}ring\text{-}1\hat{\ }'n\hat{\ }'n$
assumes *p*: p *permutes* $(UNIV :: 'n::finite\ set)$
shows $det(\chi\ i.\ A\$p\ i :: 'a\hat{\ }'n\hat{\ }'n) = of\text{-}int\ (sign\ p) * det\ A$
$\langle proof \rangle$

lemma *det-permute-columns*:
fixes $A :: 'a::comm\text{-}ring\text{-}1\hat{\ }'n\hat{\ }'n$
assumes *p*: p *permutes* $(UNIV :: 'n\ set)$
shows $det(\chi\ i\ j.\ A\$i\$\ p\ j :: 'a\hat{\ }'n\hat{\ }'n) = of\text{-}int\ (sign\ p) * det\ A$
$\langle proof \rangle$

lemma *det-identical-rows*:
fixes $A :: 'a::linordered\text{-}idom\hat{\ }'n\hat{\ }'n$
assumes *ij*: $i \neq j$
and *r*: $row\ i\ A = row\ j\ A$
shows $det\ A = 0$
$\langle proof \rangle$

lemma *det-identical-columns*:
fixes $A :: 'a::linordered\text{-}idom\hat{\ }'n\hat{\ }'n$
assumes *ij*: $i \neq j$
and *r*: $column\ i\ A = column\ j\ A$
shows $det\ A = 0$
$\langle proof \rangle$

lemma *det-zero-row*:
fixes $A :: 'a::\{idom,\ ring\text{-}char\text{-}0\}\hat{\ }'n\hat{\ }'n$
assumes *r*: $row\ i\ A = 0$
shows $det\ A = 0$
$\langle proof \rangle$

lemma *det-zero-column*:
fixes $A :: 'a::\{idom,ring\text{-}char\text{-}0\}\hat{\ }'n\hat{\ }'n$
assumes *r*: $column\ i\ A = 0$
shows $det\ A = 0$
$\langle proof \rangle$

lemma *det-row-add*:

fixes $a\ b\ c :: 'n::finite \Rightarrow\ \text{-}\ \hat{}\ 'n$
shows $det((\chi\ i.\ if\ i = k\ then\ a\ i + b\ i\ else\ c\ i)::'a::comm\text{-}ring\text{-}1\hat{}'n\hat{}'n) =$
$det((\chi\ i.\ if\ i = k\ then\ a\ i\ else\ c\ i)::'a::comm\text{-}ring\text{-}1\hat{}'n\hat{}'n) +$
$det((\chi\ i.\ if\ i = k\ then\ b\ i\ else\ c\ i)::'a::comm\text{-}ring\text{-}1\hat{}'n\hat{}'n)$
⟨*proof*⟩

lemma *det-row-mul*:
fixes $a\ b :: 'n::finite \Rightarrow\ \text{-}\ \hat{}\ 'n$
shows $det((\chi\ i.\ if\ i = k\ then\ c *s\ a\ i\ else\ b\ i)::'a::comm\text{-}ring\text{-}1\hat{}'n\hat{}'n) =$
$c*\ det((\chi\ i.\ if\ i = k\ then\ a\ i\ else\ b\ i)::'a::comm\text{-}ring\text{-}1\hat{}'n\hat{}'n)$

⟨*proof*⟩

lemma *det-row-0*:
fixes $b :: 'n::finite \Rightarrow\ \text{-}\ \hat{}\ 'n$
shows $det((\chi\ i.\ if\ i = k\ then\ 0\ else\ b\ i)::'a::comm\text{-}ring\text{-}1\hat{}'n\hat{}'n) = 0$
⟨*proof*⟩

lemma *det-row-operation*:
fixes $A :: 'a::linordered\text{-}idom\hat{}'n\hat{}'n$
assumes ij: $i \neq j$
shows $det\ (\chi\ k.\ if\ k = i\ then\ row\ i\ A + c *s\ row\ j\ A\ else\ row\ k\ A) = det\ A$
⟨*proof*⟩

lemma *det-row-span*:
fixes $A :: real\hat{}'n\hat{}'n$
assumes x: $x \in span\ \{row\ j\ A\ |j.\ j \neq i\}$
shows $det\ (\chi\ k.\ if\ k = i\ then\ row\ i\ A + x\ else\ row\ k\ A) = det\ A$
⟨*proof*⟩

lemma *det-dependent-rows*:
fixes $A:: real\hat{}'n\hat{}'n$
assumes d: $dependent\ (rows\ A)$
shows $det\ A = 0$
⟨*proof*⟩

lemma *det-dependent-columns*: **assumes** d: $dependent(columns\ (A::real\hat{}'n\hat{}'n))$
shows $det\ A = 0$
⟨*proof*⟩

lemma *Cart-lambda-cong*: $(\bigwedge x.\ f\,x = g\,x) \Longrightarrow (Cart\text{-}lambda\, f{::}'a\,\hat{}\,'n) = (Cart\text{-}lambda\ g :: 'a\,\hat{}\,'n)$
⟨*proof*⟩

lemma *det-linear-row-setsum*:
assumes *fS*: *finite S*
shows $det\ ((\chi\ i.\ if\ i = k\ then\ setsum\ (a\ i)\ S\ else\ c\ i){::}'a{::}comm\text{-}ring\text{-}1\,\hat{}\,'n\,\hat{}\,'n) = setsum\ (\lambda j.\ det\ ((\chi\ i.\ if\ i = k\ then\ a\ \ i\ j\ else\ c\ i){::}'a\,\hat{}\,'n\,\hat{}\,'n))\ S$
⟨*proof*⟩

lemma *finite-bounded-functions*:
assumes *fS*: *finite S*
shows *finite* $\{f.\ (\forall\, i \in \{1..\ (k{::}nat)\}.\ f\ i \in S) \wedge (\forall\, i.\ i \notin \{1\ ..\ k\} \longrightarrow f\ i = i)\}$
⟨*proof*⟩

lemma *eq-id-iff* [*simp*]: $(\forall\, x.\ f\ x = x) = (f = id)$ ⟨*proof*⟩

lemma *det-linear-rows-setsum-lemma*:
assumes *fS*: *finite S* **and** *fT*: *finite T*
shows $det((\chi\ i.\ if\ i \in T\ then\ setsum\ (a\ i)\ S\ else\ c\ i){::}\ 'a{::}comm\text{-}ring\text{-}1\,\hat{}\,'n\,\hat{}\,'n) =$
$setsum\ (\lambda f.\ det((\chi\ i.\ if\ i \in T\ then\ a\ i\ (f\ i)\ else\ c\ i){::}'a\,\hat{}\,'n\,\hat{}\,'n))$
$\{f.\ (\forall\, i \in T.\ f\ i \in S) \wedge (\forall\, i.\ i \notin T \longrightarrow f\ i = i)\}$
⟨*proof*⟩

lemma *det-linear-rows-setsum*:
assumes *fS*: *finite* $(S{::}'n{::}finite\ set)$
shows $det\ (\chi\ i.\ setsum\ (a\ i)\ S) = setsum\ (\lambda f.\ det\ (\chi\ i.\ a\ i\ (f\ i)\ ::\ 'a{::}comm\text{-}ring\text{-}1\ \hat{}\ 'n\,\hat{}\,'n))\ \{f.\ \forall\, i.\ f\ i \in S\}$
⟨*proof*⟩

lemma *matrix-mul-setsum-alt*:
fixes $A\ B\ ::\ 'a{::}comm\text{-}ring\text{-}1\,\hat{}\,'n\,\hat{}\,'n$
shows $A\ {*}{*}\ B = (\chi\ i.\ setsum\ (\lambda k.\ A\$i\$k\ {*}s\ B\ \$\ k)\ (UNIV\ ::\ 'n\ set))$
⟨*proof*⟩

lemma *det-rows-mul*:
$det((\chi\ i.\ c\ i\ {*}s\ a\ i){::}'a{::}comm\text{-}ring\text{-}1\,\hat{}\,'n\,\hat{}\,'n) =$
$setprod\ (\lambda i.\ c\ i)\ (UNIV{::}\ 'n\ set) * det((\chi\ i.\ a\ i){::}'a\,\hat{}\,'n\,\hat{}\,'n)$
⟨*proof*⟩

lemma *det-mul*:
fixes $A\ B\ ::\ 'a{::}linordered\text{-}idom\,\hat{}\,'n\,\hat{}\,'n$
shows $det\ (A\ {*}{*}\ B) = det\ A * det\ B$
⟨*proof*⟩

lemma *invertible-left-inverse*:
fixes $A :: real^\frown{'}n^\frown{'}n$
shows $invertible\ A \longleftrightarrow (\exists (B::real^\frown{'}n^\frown{'}n).\ B{**}\ A = mat\ 1)$
⟨*proof*⟩

lemma *invertible-righ-inverse*:
fixes $A :: real^\frown{'}n^\frown{'}n$
shows $invertible\ A \longleftrightarrow (\exists (B::real^\frown{'}n^\frown{'}n).\ A{**}\ B = mat\ 1)$
⟨*proof*⟩

lemma *invertible-det-nz*:
fixes $A::real\ ^\frown{'}n^\frown{'}n$
shows $invertible\ A \longleftrightarrow det\ A \neq 0$
⟨*proof*⟩

lemma *cramer-lemma-transpose*:
fixes $A::\ real^\frown{'}n^\frown{'}n$ **and** $x :: real^\frown{'}n$
shows $det\ ((\chi\ i.\ if\ i = k\ then\ setsum\ (\lambda i.\ x\$i\ {*s}\ row\ i\ A)\ (UNIV::'n\ set)$
$else\ row\ i\ A)::real^\frown{'}n^\frown{'}n) = x\$k * det\ A$
(**is** *?lhs = ?rhs*)
⟨*proof*⟩

lemma *cramer-lemma*:
fixes $A :: real^\frown{'}n^\frown{'}n$
shows $det((\chi\ i\ j.\ if\ j = k\ then\ (A\ {*v}\ x)\$i\ else\ A\$i\$j)::\ real^\frown{'}n^\frown{'}n) = x\$k * det$
A
⟨*proof*⟩

lemma *cramer*:
fixes $A ::real^\frown{'}n^\frown{'}n$
assumes *d0*: $det\ A \neq 0$
shows $A\ {*v}\ x = b \longleftrightarrow x = (\chi\ k.\ det(\chi\ i\ j.\ if\ j{=}k\ then\ b\$i\ else\ A\$i\$j)\ /\ det\ A)$
⟨*proof*⟩

definition $orthogonal\text{-}transformation\ f \longleftrightarrow linear\ f \wedge (\forall\ v\ w.\ f\ v \cdot f\ w = v \cdot w)$

lemma *orthogonal-transformation*: $orthogonal\text{-}transformation\ f \longleftrightarrow linear\ f \wedge$ $(\forall (v::real\ ^\frown\text{-}).\ norm\ (f\ v) = norm\ v)$
⟨*proof*⟩

definition *orthogonal-matrix* $(Q::'a::semiring\text{-}1^{\wedge\prime}n^{\wedge\prime}n) \longleftrightarrow$ *transpose* $Q ** Q =$ *mat* $1 \wedge Q **$ *transpose* $Q =$ *mat* 1

lemma *orthogonal-matrix*: *orthogonal-matrix* $(Q::\ real\ ^{\wedge\prime}n^{\wedge\prime}n) \longleftrightarrow$ *transpose* Q $** Q =$ *mat* 1
⟨*proof*⟩

lemma *orthogonal-matrix-id*: *orthogonal-matrix* (*mat* 1 :: $\text{-}^{\wedge\prime}n^{\wedge\prime}n$)
⟨*proof*⟩

lemma *orthogonal-matrix-mul*:
fixes A :: *real* $^{\wedge\prime}n^{\wedge\prime}n$
assumes *oA* : *orthogonal-matrix* A
and *oB*: *orthogonal-matrix* B
shows *orthogonal-matrix*$(A ** B)$
⟨*proof*⟩

lemma *orthogonal-transformation-matrix*:
fixes f:: *real*$^{\wedge\prime}n \Rightarrow$ *real*$^{\wedge\prime}n$
shows *orthogonal-transformation* $f \longleftrightarrow$ *linear* $f \wedge$ *orthogonal-matrix*(*matrix* f)
(**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

lemma *det-orthogonal-matrix*:
fixes Q:: $'a$::*linordered-idom*$^{\wedge\prime}n^{\wedge\prime}n$
assumes *oQ*: *orthogonal-matrix* Q
shows *det* $Q = 1 \vee$ *det* $Q = -1$
⟨*proof*⟩

lemma *scaling-linear*:
fixes f :: *real* $^{\wedge\prime}n \Rightarrow$ *real* $^{\wedge\prime}n$
assumes *f0*: $f\ 0 = 0$ **and** *fd*: $\forall x\ y.$ *dist* $(f\ x)\ (f\ y) = c\ *$ *dist* $x\ y$
shows *linear* f
⟨*proof*⟩

lemma *isometry-linear*:
$f\ (0::\ real^{\wedge\prime}n) = (0::\ real^{\wedge\prime}n) \Longrightarrow \forall x\ y.$ *dist*$(f\ x)\ (f\ y) =$ *dist* $x\ y$
$\Longrightarrow$ *linear* f
⟨*proof*⟩

lemma *orthogonal-transformation-isometry*:

orthogonal-transformation f $\longleftrightarrow$ $f(0::real^{\wedge\prime}n) = (0::real^{\wedge\prime}n) \wedge (\forall x\ y.\ dist(f\ x)\ (f\ y) = dist\ x\ y)$
⟨*proof*⟩

lemma *isometry-sphere-extend*:
fixes f:: *real* $^{\wedge\prime}n \Rightarrow$ *real* $^{\wedge\prime}n$
assumes *f1*: $\forall x.\ norm\ x = 1 \longrightarrow norm\ (f\ x) = 1$
and *fd1*: $\forall\ x\ y.\ norm\ x = 1 \longrightarrow norm\ y = 1 \longrightarrow dist\ (f\ x)\ (f\ y) = dist\ x\ y$
shows $\exists g.$ *orthogonal-transformation* $g \wedge (\forall x.\ norm\ x = 1 \longrightarrow g\ x = f\ x)$
⟨*proof*⟩

definition *rotation-matrix Q* $\longleftrightarrow$ *orthogonal-matrix Q* $\wedge$ *det Q* $= 1$
definition *rotoinversion-matrix Q* $\longleftrightarrow$ *orthogonal-matrix Q* $\wedge$ *det Q* $= -1$

lemma *orthogonal-rotation-or-rotoinversion*:
fixes Q :: *'a::linordered-idom*$^{\wedge\prime}n^{\wedge\prime}n$
shows *orthogonal-matrix Q* $\longleftrightarrow$ *rotation-matrix Q* $\vee$ *rotoinversion-matrix Q*
⟨*proof*⟩

lemma *setprod-1*: *setprod f* $\{(1::nat)..1\} = f\ 1$ ⟨*proof*⟩

lemma *setprod-2*: *setprod f* $\{(1::nat)..2\} = f\ 1 * f\ 2$
⟨*proof*⟩
lemma *setprod-3*: *setprod f* $\{(1::nat)..3\} = f\ 1 * f\ 2 * f\ 3$
⟨*proof*⟩

lemma *det-1*: *det* $(A::'a::comm\text{-}ring\text{-}1^{\wedge}1^{\wedge}1) = A\$1\$1$
⟨*proof*⟩

lemma *det-2*: *det* $(A::'a::comm\text{-}ring\text{-}1^{\wedge}2^{\wedge}2) = A\$1\$1 * A\$2\$2 - A\$1\$2 * A\$2\$1$
⟨*proof*⟩

lemma *det-3*: *det* $(A::'a::comm\text{-}ring\text{-}1^{\wedge}3^{\wedge}3) =$
$A\$1\$1 * A\$2\$2 * A\$3\$3 +$
$A\$1\$2 * A\$2\$3 * A\$3\$1 +$
$A\$1\$3 * A\$2\$1 * A\$3\$2 -$
$A\$1\$1 * A\$2\$3 * A\$3\$2 -$
$A\$1\$2 * A\$2\$1 * A\$3\$3 -$

A$1$3 ∗ A$2$2 ∗ A$3$1
⟨*proof*⟩

end

19 Convex-Euclidean-Space: Convex sets, functions and related things.

theory *Convex-Euclidean-Space*
imports *Topology-Euclidean-Space Convex*
begin

declare *vector-add-ldistrib*[*simp*] *vector-ssub-ldistrib*[*simp*] *vector-smult-assoc*[*simp*]
vector-smult-rneg[*simp*]
declare *vector-sadd-rdistrib*[*simp*] *vector-sub-rdistrib*[*simp*]

lemmas *vector-component-simps* = *vector-minus-component vector-smult-component*
vector-add-component vector-le-def Cart-lambda-beta basis-component vector-uminus-component

lemma *norm-not-0*:$(x::real^{\wedge\prime}n)\neq 0 \Longrightarrow norm\ x \neq 0$ ⟨*proof*⟩

lemma *setsum-delta-notmem*: **assumes** $x\notin s$
shows $setsum\ (\lambda y.\ if\ (y = x)\ then\ P\ x\ else\ Q\ y)\ s = setsum\ Q\ s$
$setsum\ (\lambda y.\ if\ (x = y)\ then\ P\ x\ else\ Q\ y)\ s = setsum\ Q\ s$
$setsum\ (\lambda y.\ if\ (y = x)\ then\ P\ y\ else\ Q\ y)\ s = setsum\ Q\ s$
$setsum\ (\lambda y.\ if\ (x = y)\ then\ P\ y\ else\ Q\ y)\ s = setsum\ Q\ s$
⟨*proof*⟩

lemma *setsum-delta''*:
fixes $s::'a::real\text{-}vector\ set$ **assumes** *finite s*
shows $(\sum x\in s.\ (if\ y = x\ then\ f\ x\ else\ 0) *_R x) = (if\ y\in s\ then\ (f\ y) *_R y\ else\ 0)$
⟨*proof*⟩

lemma *not-disjointI*:$x\in A \Longrightarrow x\in B \Longrightarrow A \cap B \neq \{\}$ ⟨*proof*⟩

lemma *if-smult*:$(if\ P\ then\ x\ else\ (y::real)) *_R v = (if\ P\ then\ x *_R v\ else\ y *_R v)$
⟨*proof*⟩

lemma *image-smult-interval*:$(\lambda x.\ m *_R (x::real^{\wedge\prime}n))$ ‘ $\{a..b\}$ =
$(if\ \{a..b\} = \{\}\ then\ \{\}\ else\ if\ 0 \le m\ then\ \{m *_R a..m *_R b\}\ else\ \{m *_R b..m$

$*_R$ $a\})$
  ⟨*proof*⟩

lemma *dist-triangle-eq*:
  fixes $x\ y\ z$ :: *real* ^ -
  shows $dist\ x\ z = dist\ x\ y + dist\ y\ z \longleftrightarrow norm\ (x - y) *_R (y - z) = norm\ (y - z) *_R (x - y)$
⟨*proof*⟩

lemma *norm-eqI*:$x = y \Longrightarrow norm\ x = norm\ y$ ⟨*proof*⟩
lemma *norm-minus-eqI*:$(x::real\hat{}'n) = - y \Longrightarrow norm\ x = norm\ y$ ⟨*proof*⟩

lemma *Min-grI*: **assumes** *finite A* $A \neq \{\}$ $\forall a{\in}A.\ x < a$ **shows** $x < Min\ A$
  ⟨*proof*⟩

lemma *dimindex-ge-1*:$CARD(\text{-}::finite) \geq 1$
  ⟨*proof*⟩

lemma *real-dimindex-ge-1*:$real\ (CARD('n::finite)) \geq 1$
  ⟨*proof*⟩

lemma *real-dimindex-gt-0*:$real\ (CARD('n::finite)) > 0$ ⟨*proof*⟩

19.1 Affine set and affine hull.

definition
  affine :: *'a::real-vector set* ⇒ *bool* **where**
  $affine\ s \longleftrightarrow (\forall x{\in}s.\ \forall y{\in}s.\ \forall u\ v.\ u + v = 1 \longrightarrow u *_R x + v *_R y \in s)$

lemma *affine-alt*: $affine\ s \longleftrightarrow (\forall x{\in}s.\ \forall y{\in}s.\ \forall u::real.\ (1 - u) *_R x + u *_R y \in s)$
⟨*proof*⟩

lemma *affine-empty*[*intro*]: *affine* {}
  ⟨*proof*⟩

lemma *affine-sing*[*intro*]: *affine* $\{x\}$
  ⟨*proof*⟩

lemma *affine-UNIV*[*intro*]: *affine UNIV*
  ⟨*proof*⟩

lemma *affine-Inter*: $(\forall s{\in}f.\ affine\ s) \Longrightarrow affine\ (\bigcap\ f)$
  ⟨*proof*⟩

lemma *affine-Int*: $affine\ s \Longrightarrow affine\ t \Longrightarrow affine\ (s \cap t)$
  ⟨*proof*⟩

lemma *affine-affine-hull*: *affine*(*affine hull s*)

⟨*proof*⟩

lemma *affine-hull-eq*[*simp*]: (*affine hull s* = *s*) $\longleftrightarrow$ *affine s*
⟨*proof*⟩

lemma *setsum-restrict-set''*: **assumes** *finite A*
shows *setsum f* $\{x \in A.\ P\ x\} = (\sum x{\in}A.\ \textit{if } P\ x\ \ \textit{then } f\ x\ \textit{else } 0)$
⟨*proof*⟩

19.2 Some explicit formulations (from Lars Schewe).

lemma *affine*: **fixes** V::'*a*::*real-vector set*
shows *affine V* $\longleftrightarrow$ $(\forall\, s\ u.\ \textit{finite } s \wedge s \neq \{\} \wedge s \subseteq V \wedge \textit{setsum } u\ s = 1 \longrightarrow (\textit{setsum } (\lambda x.\ (u\ x) *_R x))\ s \in V)$
⟨*proof*⟩

lemma *affine-hull-explicit*:
affine hull $p = \{y.\ \exists\, s\ u.\ \textit{finite } s \wedge s \neq \{\} \wedge s \subseteq p \wedge \textit{setsum } u\ s = 1 \wedge \textit{setsum } (\lambda v.\ (u\ v) *_R v)\ s = y\}$
⟨*proof*⟩

lemma *affine-hull-finite*:
assumes *finite s*
shows *affine hull* $s = \{y.\ \exists\, u.\ \textit{setsum } u\ s = 1 \wedge \textit{setsum } (\lambda v.\ u\ v *_R v)\ s = y\}$
⟨*proof*⟩

19.3 Stepping theorems and hence small special cases.

lemma *affine-hull-empty*[*simp*]: *affine hull* {} = {}
⟨*proof*⟩

lemma *affine-hull-finite-step*:
fixes y :: '*a*::*real-vector*
shows $(\exists\, u.\ \textit{setsum } u\ \{\} = w \wedge \textit{setsum } (\lambda x.\ u\ x *_R x)\ \{\} = y) \longleftrightarrow w = 0 \wedge y = 0$ (**is** *?th1*)
finite s $\Longrightarrow (\exists\, u.\ \textit{setsum } u\ (\textit{insert } a\ s) = w \wedge \textit{setsum } (\lambda x.\ u\ x *_R x)\ (\textit{insert } a\ s) = y) \longleftrightarrow$
$(\exists\, v\ u.\ \textit{setsum } u\ s = w - v \wedge \textit{setsum } (\lambda x.\ u\ x *_R x)\ s = y - v *_R a)$ (**is** *?as* $\Longrightarrow$ (*?lhs* = *?rhs*))
⟨*proof*⟩

lemma *affine-hull-2*:
fixes $a\ b$ :: '*a*::*real-vector*
shows *affine hull* $\{a,b\} = \{u *_R a + v *_R b \mid u\ v.\ (u + v = 1)\}$ (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *affine-hull-3*:
fixes $a\ b\ c$ :: '*a*::*real-vector*
shows *affine hull* $\{a,b,c\} = \{\ u *_R a + v *_R b + w *_R c \mid u\ v\ w.\ u + v + w = 1\}$ (**is** *?lhs* = *?rhs*)

⟨*proof*⟩

19.4 Some relations between affine hull and subspaces.

lemma *affine-hull-insert-subset-span*:
fixes $a :: real\ \hat{}\ \text{-}$
shows $affine\ hull\ (insert\ a\ s) \subseteq \{a + v|\ v\ .\ v \in span\ \{x - a\ |\ x\ .\ x \in s\}\}$
⟨*proof*⟩

lemma *affine-hull-insert-span*:
fixes $a :: real\ \hat{}\ \text{-}$
assumes $a \notin s$
shows $affine\ hull\ (insert\ a\ s) =$
$\{a + v\ |\ v\ .\ v \in span\ \{x - a\ |\ x.\ \ x \in s\}\}$
⟨*proof*⟩

lemma *affine-hull-span*:
fixes $a :: real\ \hat{}\ \text{-}$
assumes $a \in s$
shows $affine\ hull\ s = \{a + v\ |\ v.\ v \in span\ \{x - a\ |\ x.\ x \in s - \{a\}\}\}$
⟨*proof*⟩

19.5 Cones.

definition
$cone :: 'a{::}real\text{-}vector\ set \Rightarrow bool$ **where**
$cone\ s \longleftrightarrow (\forall\, x{\in}s.\ \forall\, c{\geq}0.\ (c *_R x) \in s)$

lemma *cone-empty*[*intro*, *simp*]: $cone\ \{\}$
⟨*proof*⟩

lemma *cone-univ*[*intro*, *simp*]: $cone\ UNIV$
⟨*proof*⟩

lemma *cone-Inter*[*intro*]: $(\forall\, s{\in}f.\ cone\ s) \Longrightarrow cone(\bigcap\ f)$
⟨*proof*⟩

19.6 Conic hull.

lemma *cone-cone-hull*: $cone\ (cone\ hull\ s)$
⟨*proof*⟩

lemma *cone-hull-eq*: $(cone\ hull\ s = s) \longleftrightarrow cone\ s$
⟨*proof*⟩

19.7 Affine dependence and consequential theorems (from Lars Schewe).

definition
$affine\text{-}dependent :: 'a{::}real\text{-}vector\ set \Rightarrow bool$ **where**

affine-dependent $s \longleftrightarrow (\exists\, x{\in}s.\ x \in (\mathit{affine\ hull}\ (s - \{x\})))$

lemma *affine-dependent-explicit*:
affine-dependent $p \longleftrightarrow$
$(\exists\, s\ u.\ \mathit{finite}\ s \wedge s \subseteq p \wedge \mathit{setsum}\ u\ s = 0 \wedge$
$(\exists\, v{\in}s.\ u\ v \neq 0) \wedge \mathit{setsum}\ (\lambda v.\ u\ v *_R v)\ s = 0)$
⟨*proof*⟩

lemma *affine-dependent-explicit-finite*:
fixes s :: *'a::real-vector set* **assumes** *finite s*
shows *affine-dependent* $s \longleftrightarrow (\exists\, u.\ \mathit{setsum}\ u\ s = 0 \wedge (\exists\, v{\in}s.\ u\ v \neq 0) \wedge \mathit{setsum}$ $(\lambda v.\ u\ v *_R v)\ s = 0)$
(**is** *?lhs = ?rhs*)
⟨*proof*⟩

19.8 A general lemma.

lemma *convex-connected*:
fixes s :: *'a::real-normed-vector set*
assumes *convex s* **shows** *connected s*
⟨*proof*⟩

19.9 One rather trivial consequence.

lemma *connected-UNIV*[*intro*]: *connected* (*UNIV* :: *'a::real-normed-vector set*)
⟨*proof*⟩

19.10 Balls, being convex, are connected.

lemma *convex-box*:
assumes $\bigwedge i.\ \mathit{convex}\ \{x.\ P\ i\ x\}$
shows $\mathit{convex}\ \{x.\ \forall\, i.\ P\ i\ (x\$i)\}$
⟨*proof*⟩

lemma *convex-positive-orthant*: $\mathit{convex}\ \{x{::}\mathit{real}\hat{\ }'n.\ (\forall\, i.\ 0 \leq x\$i)\}$
⟨*proof*⟩

lemma *convex-local-global-minimum*:
fixes s :: *'a::real-normed-vector set*
assumes $0<e$ *convex-on s f* $\mathit{ball}\ x\ e \subseteq s$ $\forall\, y{\in}\mathit{ball}\ x\ e.\ f\ x \leq f\ y$
shows $\forall\, y{\in}s.\ f\ x \leq f\ y$
⟨*proof*⟩

lemma *convex-ball*:
fixes x :: *'a::real-normed-vector*
shows *convex* (*ball x e*)
⟨*proof*⟩

lemma *convex-cball*:
fixes x :: *'a::real-normed-vector*

shows *convex*(*cball x e*)
⟨*proof*⟩

lemma *connected-ball*:
fixes x :: $'a$::*real-normed-vector*
shows *connected* (*ball x e*)
⟨*proof*⟩

lemma *connected-cball*:
fixes x :: $'a$::*real-normed-vector*
shows *connected*(*cball x e*)
⟨*proof*⟩

19.11 Convex hull.

lemma *convex-convex-hull*: *convex*(*convex hull s*)
⟨*proof*⟩

lemma *convex-hull-eq*: *convex hull s* = *s* $\longleftrightarrow$ *convex s*
⟨*proof*⟩

lemma *bounded-convex-hull*:
fixes s :: $'a$::*real-normed-vector set*
assumes *bounded s* **shows** *bounded*(*convex hull s*)
⟨*proof*⟩

lemma *finite-imp-bounded-convex-hull*:
fixes s :: $'a$::*real-normed-vector set*
shows *finite s* $\Longrightarrow$ *bounded*(*convex hull s*)
⟨*proof*⟩

19.12 Stepping theorems for convex hulls of finite sets.

lemma *convex-hull-empty*[*simp*]: *convex hull* {} = {}
⟨*proof*⟩

lemma *convex-hull-singleton*[*simp*]: *convex hull* {*a*} = {*a*}
⟨*proof*⟩

lemma *convex-hull-insert*:
fixes s :: $'a$::*real-vector set*
assumes $s \neq \{\}$
shows *convex hull* (*insert a s*) = $\{x.\ \exists u \geq 0.\ \exists v \geq 0.\ \exists b.\ (u + v = 1) \wedge$
$b \in (convex\ hull\ s) \wedge (x = u *_R a + v *_R b)\}$ (**is**
?xyz = *?hull*)
⟨*proof*⟩

19.13 Explicit expression for convex hull.

lemma *convex-hull-indexed*:

fixes s :: $'a$::*real-vector set*
shows *convex hull* $s = \{y.\ \exists\, k\ u\ x.\ (\forall\, i{\in}\{1{::}nat\ ..\ k\}.\ 0 \le u\ i \wedge x\ i \in s) \wedge$
$(setsum\ u\ \{1..k\} = 1) \wedge$
$(setsum\ (\lambda i.\ u\ i *_R x\ i)\ \{1..k\} = y)\}$ (**is** $?xyz = ?hull$)
⟨*proof*⟩

lemma *convex-hull-finite*:
fixes s :: $'a$::*real-vector set*
assumes *finite s*
shows *convex hull* $s = \{y.\ \exists\, u.\ (\forall\, x{\in}s.\ 0 \le u\ x) \wedge$
$setsum\ u\ s = 1 \wedge setsum\ (\lambda x.\ u\ x *_R x)\ s = y\}$ (**is** $?HULL = ?set$)
⟨*proof*⟩

19.14 Another formulation from Lars Schewe.

lemma *setsum-constant-scaleR*:
fixes y :: $'a$::*real-vector*
shows $(\sum x{\in}A.\ y) = of\text{-}nat\ (card\ A) *_R y$
⟨*proof*⟩

lemma *convex-hull-explicit*:
fixes p :: $'a$::*real-vector set*
shows *convex hull* $p = \{y.\ \exists\, s\ u.\ finite\ s \wedge s \subseteq p \wedge$
$(\forall\, x{\in}s.\ 0 \le u\ x) \wedge setsum\ u\ s = 1 \wedge setsum\ (\lambda v.\ u\ v *_R v)\ s = y\}$ (**is** $?lhs = ?rhs$)
⟨*proof*⟩

19.15 A stepping theorem for that expansion.

lemma *convex-hull-finite-step*:
fixes s :: $'a$::*real-vector set* **assumes** *finite s*
shows $(\exists\, u.\ (\forall\, x{\in}insert\ a\ s.\ 0 \le u\ x) \wedge setsum\ u\ (insert\ a\ s) = w \wedge setsum\ (\lambda x.\ u\ x *_R x)\ (insert\ a\ s) = y)$
$\longleftrightarrow (\exists\, v{\ge}0.\ \exists\, u.\ (\forall\, x{\in}s.\ 0 \le u\ x) \wedge setsum\ u\ s = w - v \wedge setsum\ (\lambda x.\ u\ x *_R x)\ s = y - v *_R a)$ (**is** $?lhs = ?rhs$)
⟨*proof*⟩

19.16 Hence some special cases.

lemma *convex-hull-2*:
convex hull $\{a,b\} = \{u *_R a + v *_R b \mid u\ v.\ 0 \le u \wedge 0 \le v \wedge u + v = 1\}$
⟨*proof*⟩

lemma *convex-hull-2-alt*: *convex hull* $\{a,b\} = \{a + u *_R (b - a) \mid u.\ 0 \le u \wedge u \le 1\}$
⟨*proof*⟩

lemma *convex-hull-3*:
convex hull $\{a,b,c\} = \{ u *_R a + v *_R b + w *_R c \mid u\ v\ w.\ 0 \le u \wedge 0 \le v \wedge 0 \le w \wedge u + v + w = 1\}$

⟨*proof*⟩

lemma *convex-hull-3-alt*:
convex hull $\{a,b,c\} = \{a + u *_R (b - a) + v *_R (c - a) \mid u\ v.\ 0 \leq u \wedge 0 \leq v \wedge u + v \leq 1\}$
⟨*proof*⟩

19.17 Relations among closure notions and corresponding hulls.

TODO: Generalize linear algebra concepts defined in *Euclidean-Space.thy* so that we can generalize these lemmas.

lemma *subspace-imp-affine*:
fixes s :: (*real* ^ -) *set* **shows** *subspace* $s \Longrightarrow$ *affine* s
⟨*proof*⟩

lemma *affine-imp-convex*: *affine* $s \Longrightarrow$ *convex* s
⟨*proof*⟩

lemma *subspace-imp-convex*:
fixes s :: (*real* ^ -) *set* **shows** *subspace* $s \Longrightarrow$ *convex* s
⟨*proof*⟩

lemma *affine-hull-subset-span*:
fixes s :: (*real* ^ -) *set* **shows** (*affine hull* s) $\subseteq$ (*span* s)
⟨*proof*⟩

lemma *convex-hull-subset-span*:
fixes s :: (*real* ^ -) *set* **shows** (*convex hull* s) $\subseteq$ (*span* s)
⟨*proof*⟩

lemma *convex-hull-subset-affine-hull*: (*convex hull* s) $\subseteq$ (*affine hull* s)
⟨*proof*⟩

lemma *affine-dependent-imp-dependent*:
fixes s :: (*real* ^ -) *set* **shows** *affine-dependent* $s \Longrightarrow$ *dependent* s
⟨*proof*⟩

lemma *dependent-imp-affine-dependent*:
fixes s :: (*real* ^ -) *set*
assumes *dependent* $\{x - a \mid x\ .\ x \in s\}$ $a \notin s$
shows *affine-dependent* (*insert* a s)
⟨*proof*⟩

lemma *convex-cone*:
convex $s \wedge$ *cone* $s \longleftrightarrow (\forall x \in s.\ \forall y \in s.\ (x + y) \in s) \wedge (\forall x \in s.\ \forall c \geq 0.\ (c *_R x) \in s)$ (**is** *?lhs* = *?rhs*)

⟨*proof*⟩

lemma *affine-dependent-biggerset*: **fixes** *s*::(*real*^'*n*) *set*
assumes *finite s card s* $\geq$ *CARD*('*n*) + *2*
shows *affine-dependent s*
⟨*proof*⟩

lemma *affine-dependent-biggerset-general*:
assumes *finite* (*s*::(*real*^'*n*) *set*) *card s* $\geq$ *dim s* + *2*
shows *affine-dependent s*
⟨*proof*⟩

19.18 Caratheodory's theorem.

lemma *convex-hull-caratheodory*: **fixes** *p*::(*real*^'*n*) *set*
shows *convex hull p* = {*y*. ∃ *s u*. *finite s* ∧ *s* ⊆ *p* ∧ *card s* ≤ *CARD*('*n*) + *1* ∧ (∀ *x*∈*s*. *0* ≤ *u x*) ∧ *setsum u s* = *1* ∧ *setsum* (λ*v*. *u v* $*_R$ *v*) *s* = *y*}
⟨*proof*⟩

lemma *caratheodory*:
convex hull p = {*x*::*real*^'*n*. ∃ *s*. *finite s* ∧ *s* ⊆ *p* ∧
card s ≤ *CARD*('*n*) + *1* ∧ *x* ∈ *convex hull s*}
⟨*proof*⟩

19.19 Openness and compactness are preserved by convex hull operation.

lemma *open-convex-hull*[*intro*]:
fixes *s* :: '*a*::*real-normed-vector set*
assumes *open s*
shows *open*(*convex hull s*)
⟨*proof*⟩

lemma *compact-real-interval*:
fixes *a b* :: *real* **shows** *compact* {*a*..*b*}
⟨*proof*⟩

lemma *compact-convex-combinations*:
fixes *s t* :: '*a*::*real-normed-vector set*
assumes *compact s compact t*
shows *compact* { (*1* − *u*) $*_R$ *x* + *u* $*_R$ *y* | *x y u*. *0* ≤ *u* ∧ *u* ≤ *1* ∧ *x* ∈ *s* ∧ *y* ∈ *t*}
⟨*proof*⟩

lemma *compact-convex-hull*: **fixes** *s*::(*real*^'*n*) *set*
assumes *compact s* **shows** *compact*(*convex hull s*)
⟨*proof*⟩

lemma *finite-imp-compact-convex-hull*:
  fixes s :: (*real* ^ -) *set*
  shows *finite* $s \Longrightarrow$ *compact*(*convex hull* s)
⟨*proof*⟩

19.20 Extremal points of a simplex are some vertices.

lemma *dist-increases-online*:
  fixes $a\ b\ d$ :: $'a$::*real-inner*
  assumes $d \neq 0$
  shows *dist* $a\ (b + d) >$ *dist* $a\ b \lor$ *dist* $a\ (b - d) >$ *dist* $a\ b$
⟨*proof*⟩

lemma *norm-increases-online*:
  fixes d :: $'a$::*real-inner*
  shows $d \neq 0 \Longrightarrow$ *norm*$(a + d) >$ *norm* $a \lor$ *norm*$(a - d) >$ *norm* a
  ⟨*proof*⟩

lemma *simplex-furthest-lt*:
  fixes s::$'a$::*real-inner set* **assumes** *finite* s
  shows $\forall x \in$ (*convex hull* s). $x \notin s \longrightarrow (\exists y \in$(*convex hull* s). *norm*$(x - a) <$ *norm*$(y - a))$
⟨*proof*⟩

lemma *simplex-furthest-le*:
  fixes s :: (*real* ^ -) *set*
  assumes *finite* s $s \neq \{\}$
  shows $\exists y \in s. \forall x \in$(*convex hull* s). *norm*$(x - a) \leq$ *norm*$(y - a)$
⟨*proof*⟩

lemma *simplex-furthest-le-exists*:
  fixes s :: (*real* ^ -) *set*
  shows *finite* $s \Longrightarrow (\forall x \in$(*convex hull* s). $\exists y \in s.$ *norm*$(x - a) \leq$ *norm*$(y - a))$
  ⟨*proof*⟩

lemma *simplex-extremal-le*:
  fixes s :: (*real* ^ -) *set*
  assumes *finite* s $s \neq \{\}$
  shows $\exists u \in s. \exists v \in s. \forall x \in$*convex hull* s. $\forall y \in$ *convex hull* s. *norm*$(x - y) \leq$ *norm*$(u - v)$
⟨*proof*⟩

lemma *simplex-extremal-le-exists*:
  fixes s :: (*real* ^ -) *set*
  shows *finite* $s \Longrightarrow x \in$ *convex hull* $s \Longrightarrow y \in$ *convex hull* s
  $\Longrightarrow (\exists u \in s. \exists v \in s.$ *norm*$(x - y) \leq$ *norm*$(u - v))$
  ⟨*proof*⟩

19.21 Closest point of a convex set is unique, with a continuous projection.

definition
closest-point :: $'a$::{*real-inner*,*heine-borel*} *set* $\Rightarrow$ $'a$ $\Rightarrow$ $'a$ **where**
closest-point s a = (*SOME* x. $x \in s \wedge (\forall y{\in}s.$ *dist* a $x \leq$ *dist* a y))

lemma *closest-point-exists*:
assumes *closed* s $s \neq \{\}$
shows *closest-point* s $a \in s$ $\forall y{\in}s.$ *dist* a (*closest-point* s a) $\leq$ *dist* a y
⟨*proof*⟩

lemma *closest-point-in-set*:
closed $s \Longrightarrow s \neq \{\} \Longrightarrow$ (*closest-point* s a) $\in s$
⟨*proof*⟩

lemma *closest-point-le*:
closed $s \Longrightarrow x \in s \Longrightarrow$ *dist* a (*closest-point* s a) $\leq$ *dist* a x
⟨*proof*⟩

lemma *closest-point-self*:
assumes $x \in s$ **shows** *closest-point* s $x = x$
⟨*proof*⟩

lemma *closest-point-refl*:
closed $s \Longrightarrow s \neq \{\} \Longrightarrow$ (*closest-point* s $x = x \longleftrightarrow x \in s$)
⟨*proof*⟩

lemma *norm-lt*: *norm* x < *norm* y $\longleftrightarrow$ *inner* x x < *inner* y y
⟨*proof*⟩

lemma *norm-le*: *norm* $x \leq$ *norm* y $\longleftrightarrow$ *inner* x $x \leq$ *inner* y y
⟨*proof*⟩

lemma *closer-points-lemma*:
assumes *inner* y $z > 0$
shows $\exists u{>}0.$ $\forall v{>}0.$ $v \leq u \longrightarrow$ *norm*($v *_R z - y$) < *norm* y
⟨*proof*⟩

lemma *closer-point-lemma*:
assumes *inner* ($y - x$) ($z - x$) > 0
shows $\exists u{>}0.$ $u \leq 1 \wedge$ *dist* ($x + u *_R (z - x)$) y < *dist* x y
⟨*proof*⟩

lemma *any-closest-point-dot*:
assumes *convex* s *closed* s $x \in s$ $y \in s$ $\forall z{\in}s.$ *dist* a $x \leq$ *dist* a z
shows *inner* ($a - x$) ($y - x$) ≤ 0
⟨*proof*⟩

lemma *any-closest-point-unique*:
fixes x :: $'a$::*real-inner*
assumes *convex* s *closed* s $x \in s$ $y \in s$
$\forall z{\in}s.$ *dist* a $x \leq$ *dist* a z $\forall z{\in}s.$ *dist* a $y \leq$ *dist* a z
shows $x = y$ ⟨*proof*⟩

lemma *closest-point-unique*:
assumes *convex* s *closed* s $x \in s$ $\forall z{\in}s.$ *dist* a $x \leq$ *dist* a z
shows $x =$ *closest-point* s a
⟨*proof*⟩

lemma *closest-point-dot*:
assumes *convex* s *closed* s $x \in s$
shows *inner* (a − *closest-point* s a) (x − *closest-point* s a) $\leq$ *0*
⟨*proof*⟩

lemma *closest-point-lt*:
assumes *convex* s *closed* s $x \in s$ $x \neq$ *closest-point* s a
shows *dist* a (*closest-point* s a) $<$ *dist* a x
⟨*proof*⟩

lemma *closest-point-lipschitz*:
assumes *convex* s *closed* s $s \neq \{\}$
shows *dist* (*closest-point* s x) (*closest-point* s y) $\leq$ *dist* x y
⟨*proof*⟩

lemma *continuous-at-closest-point*:
assumes *convex* s *closed* s $s \neq \{\}$
shows *continuous* (*at* x) (*closest-point* s)
⟨*proof*⟩

lemma *continuous-on-closest-point*:
assumes *convex* s *closed* s $s \neq \{\}$
shows *continuous-on* t (*closest-point* s)
⟨*proof*⟩

19.22 Various point-to-set separating/supporting hyperplane theorems.

lemma *supporting-hyperplane-closed-point*:
fixes z :: $'a$::{*real-inner*,*heine-borel*}
assumes *convex* s *closed* s $s \neq \{\}$ $z \notin s$
shows $\exists\, a\ b.\ \exists\, y{\in}s.$ *inner* a $z < b \wedge$ (*inner* a $y = b$) $\wedge$ ($\forall x{\in}s.$ *inner* a $x \geq b$)
⟨*proof*⟩

lemma *separating-hyperplane-closed-point*:
fixes z :: $'a$::{*real-inner*,*heine-borel*}
assumes *convex* s *closed* s $z \notin s$

shows $\exists\, a\ b.\ inner\ a\ z < b \wedge (\forall\, x{\in}s.\ inner\ a\ x > b)$
⟨*proof*⟩

lemma *separating-hyperplane-closed-0*:
assumes $convex\ (s{::}(real\hat{}'n)\ set)\ closed\ s\ 0 \notin s$
shows $\exists\, a\ b.\ a \neq 0 \wedge 0 < b \wedge (\forall\, x{\in}s.\ inner\ a\ x > b)$
⟨*proof*⟩

19.23 Now set-to-set for closed/compact sets.

lemma *separating-hyperplane-closed-compact*:
assumes $convex\ (s{::}(real\hat{}'n)\ set)\ closed\ s\ convex\ t\ compact\ t\ t \neq \{\}\ s \cap t = \{\}$
shows $\exists\, a\ b.\ (\forall\, x{\in}s.\ inner\ a\ x < b) \wedge (\forall\, x{\in}t.\ inner\ a\ x > b)$
⟨*proof*⟩

lemma *separating-hyperplane-compact-closed*:
fixes $s :: (real\ \hat{}\ \text{-})\ set$
assumes $convex\ s\ compact\ s\ s \neq \{\}\ convex\ t\ closed\ t\ s \cap t = \{\}$
shows $\exists\, a\ b.\ (\forall\, x{\in}s.\ inner\ a\ x < b) \wedge (\forall\, x{\in}t.\ inner\ a\ x > b)$
⟨*proof*⟩

19.24 General case without assuming closure and getting non-strict separation.

lemma *separating-hyperplane-set-0*:
assumes $convex\ s\ (0{::}real\hat{}'n) \notin s$
shows $\exists\, a.\ a \neq 0 \wedge (\forall\, x{\in}s.\ 0 \leq inner\ a\ x)$
⟨*proof*⟩

lemma *separating-hyperplane-sets*:
assumes $convex\ s\ convex\ (t{::}(real\hat{}'n)\ set)\ s \neq \{\}\ t \neq \{\}\ s \cap t = \{\}$
shows $\exists\, a\ b.\ a \neq 0 \wedge (\forall\, x{\in}s.\ inner\ a\ x \leq b) \wedge (\forall\, x{\in}t.\ inner\ a\ x \geq b)$
⟨*proof*⟩

19.25 More convexity generalities.

lemma *convex-closure*:
fixes $s :: 'a{::}real\text{-}normed\text{-}vector\ set$
assumes *convex s* **shows** *convex*(*closure s*)
⟨*proof*⟩

lemma *convex-interior*:
fixes $s :: 'a{::}real\text{-}normed\text{-}vector\ set$
assumes *convex s* **shows** *convex*(*interior s*)
⟨*proof*⟩

lemma *convex-hull-eq-empty*[*simp*]: $convex\ hull\ s = \{\} \longleftrightarrow s = \{\}$
⟨*proof*⟩

19.26 Moving and scaling convex hulls.

lemma *convex-hull-translation-lemma*:
  convex hull $((\lambda x.\ a + x)\ `\ s) \subseteq (\lambda x.\ a + x)\ `\ (convex\ hull\ s)$
⟨*proof*⟩

lemma *convex-hull-bilemma*: **fixes** *neg*
  assumes $(\forall s\ a.\ (convex\ hull\ (up\ a\ s)) \subseteq up\ a\ (convex\ hull\ s))$
  shows $(\forall s.\ up\ a\ (up\ (neg\ a)\ s) = s) \wedge (\forall s.\ up\ (neg\ a)\ (up\ a\ s) = s) \wedge (\forall s\ t\ a.\ s \subseteq t \longrightarrow up\ a\ s \subseteq up\ a\ t)$
  $\Longrightarrow \forall s.\ (convex\ hull\ (up\ a\ s)) = up\ a\ (convex\ hull\ s)$
  ⟨*proof*⟩

lemma *convex-hull-translation*:
  convex hull $((\lambda x.\ a + x)\ `\ s) = (\lambda x.\ a + x)\ `\ (convex\ hull\ s)$
  ⟨*proof*⟩

lemma *convex-hull-scaling-lemma*:
$(convex\ hull\ ((\lambda x.\ c *_R x)\ `\ s)) \subseteq (\lambda x.\ c *_R x)\ `\ (convex\ hull\ s)$
⟨*proof*⟩

lemma *convex-hull-scaling*:
  convex hull $((\lambda x.\ c *_R x)\ `\ s) = (\lambda x.\ c *_R x)\ `\ (convex\ hull\ s)$
  ⟨*proof*⟩

lemma *convex-hull-affinity*:
  convex hull $((\lambda x.\ a + c *_R x)\ `\ s) = (\lambda x.\ a + c *_R x)\ `\ (convex\ hull\ s)$
⟨*proof*⟩

19.27 Convex set as intersection of halfspaces.

lemma *convex-halfspace-intersection*:
  fixes $s :: (real\ \hat{}\ \text{-})\ set$
  assumes *closed s convex s*
  shows $s = \bigcap\ \{h.\ s \subseteq h \wedge (\exists a\ b.\ h = \{x.\ inner\ a\ x \leq b\})\}$
  ⟨*proof*⟩

19.28 Radon's theorem (from Lars Schewe).

lemma *radon-ex-lemma*:
  assumes *finite c affine-dependent c*
  shows $\exists u.\ setsum\ u\ c = 0 \wedge (\exists v{\in}c.\ u\ v \neq 0) \wedge setsum\ (\lambda v.\ u\ v *_R v)\ c = 0$
⟨*proof*⟩

lemma *radon-s-lemma*:
  assumes *finite s* $setsum\ f\ s = (0{::}real)$
  shows $setsum\ f\ \{x{\in}s.\ 0 < f\ x\} = -\ setsum\ f\ \{x{\in}s.\ f\ x < 0\}$
⟨*proof*⟩

lemma *radon-v-lemma*:

assumes *finite s setsum f s* = *0* ∀ *x*. *g x* = (*0*::*real*) ⟶ *f x* = (*0*::*real*^-)
shows (*setsum f* {*x*∈*s*. *0* < *g x*}) = − *setsum f* {*x*∈*s*. *g x* < *0*}
⟨*proof*⟩

lemma *radon-partition*:
assumes *finite c affine-dependent c*
shows ∃ *m p*. *m* ∩ *p* = {} ∧ *m* ∪ *p* = *c* ∧ (*convex hull m*) ∩ (*convex hull p*) ≠ {} ⟨*proof*⟩

lemma *radon*: **assumes** *affine-dependent c*
obtains *m p* **where** *m*⊆*c p*⊆*c m* ∩ *p* = {} (*convex hull m*) ∩ (*convex hull p*) ≠ {}
⟨*proof*⟩

19.29 Helly's theorem.

lemma *helly-induct*: **fixes** *f*::(*real*^′*n*) *set set*
assumes *card f* = *n n* ≥ *CARD*(′*n*) + *1*
∀ *s*∈*f*. *convex s* ∀ *t*⊆*f*. *card t* = *CARD*(′*n*) + *1* ⟶ ⋂ *t* ≠ {}
shows ⋂ *f* ≠ {}
⟨*proof*⟩

lemma *helly*: **fixes** *f*::(*real*^′*n*) *set set*
assumes *card f* ≥ *CARD*(′*n*) + *1* ∀ *s*∈*f*. *convex s*
∀ *t*⊆*f*. *card t* = *CARD*(′*n*) + *1* ⟶ ⋂ *t* ≠ {}
shows ⋂ *f* ≠{}
⟨*proof*⟩

19.30 Convex hull is "preserved" by a linear function.

lemma *convex-hull-linear-image*:
assumes *bounded-linear f*
shows *f* ‘ (*convex hull s*) = *convex hull* (*f* ‘ *s*)
⟨*proof*⟩

lemma *in-convex-hull-linear-image*:
assumes *bounded-linear f x* ∈ *convex hull s*
shows (*f x*) ∈ *convex hull* (*f* ‘ *s*)
⟨*proof*⟩

19.31 Homeomorphism of all convex compact sets with nonempty interior.

lemma *compact-frontier-line-lemma*:
fixes *s* :: (*real* ^ -) *set*
assumes *compact s 0* ∈ *s x* ≠ *0*
obtains *u* **where** *0* ≤ *u* (*u* $*_R$ *x*) ∈ *frontier s* ∀ *v*>*u*. (*v* $*_R$ *x*) ∉ *s*
⟨*proof*⟩

lemma *starlike-compact-projective*:

assumes *compact s cball* $(0::real\hat{\ }'n)$ *1* $\subseteq$ *s*
$\forall x \in s.\ \forall u.\ 0 \leq u \wedge u < 1 \longrightarrow (u *_R x) \in (s - frontier\ s)$
shows *s homeomorphic* (*cball* $(0::real\hat{\ }'n)$ *1*)
⟨*proof*⟩

lemma *homeomorphic-convex-compact-lemma*: **fixes** $s::(real\hat{\ }'n)$ *set*
assumes *convex s compact s cball 0 1* $\subseteq$ *s*
shows *s homeomorphic* (*cball* $(0::real\hat{\ }'n)$ *1*)
⟨*proof*⟩

lemma *homeomorphic-convex-compact-cball*: **fixes** *e::real* **and** $s::(real\hat{\ }'n)$ *set*
assumes *convex s compact s interior s* $\neq \{\}$ $0 < e$
shows *s homeomorphic* (*cball* $(b::real\hat{\ }'n)$ *e*)
⟨*proof*⟩

lemma *homeomorphic-convex-compact*: **fixes** $s::(real\hat{\ }'n)$ *set* **and** $t::(real\hat{\ }'n)$ *set*
assumes *convex s compact s interior s* $\neq \{\}$
convex t compact t interior t $\neq \{\}$
shows *s homeomorphic t*
⟨*proof*⟩

19.32 Epigraphs of convex functions.

definition *epigraph s* $(f::\text{-} \Rightarrow real) = \{xy.\ fst\ xy \in s \wedge f\ (fst\ xy) \leq snd\ xy\}$

lemma *mem-epigraph*: $(x, y) \in epigraph\ s\ f \longleftrightarrow x \in s \wedge f\ x \leq y$ ⟨*proof*⟩

lemma *convex-epigraph*:
$convex(epigraph\ s\ f) \longleftrightarrow convex\text{-}on\ s\ f \wedge convex\ s$
⟨*proof*⟩

lemma *convex-epigraphI*:
$convex\text{-}on\ s\ f \Longrightarrow convex\ s \Longrightarrow convex(epigraph\ s\ f)$
⟨*proof*⟩

lemma *convex-epigraph-convex*:
$convex\ s \Longrightarrow convex\text{-}on\ s\ f \longleftrightarrow convex(epigraph\ s\ f)$
⟨*proof*⟩

19.33 Use this to derive general bound property of convex function.

lemma *convex-on*:
assumes *convex s*
shows $convex\text{-}on\ s\ f \longleftrightarrow (\forall k\ u\ x.\ (\forall i \in \{1..k::nat\}.\ 0 \leq u\ i \wedge x\ i \in s) \wedge setsum\ u\ \{1..k\} = 1 \longrightarrow$
$f\ (setsum\ (\lambda i.\ u\ i *_R x\ i)\ \{1..k\}) \leq setsum\ (\lambda i.\ u\ i * f(x\ i))\ \{1..k\})$
⟨*proof*⟩

19.34 Convexity of general and special intervals.

lemma *is-interval-convex*:
fixes $s :: (real\ \hat{}\ \text{-})\ set$
assumes *is-interval s* **shows** *convex s*
⟨*proof*⟩

lemma *is-interval-connected*:
fixes $s :: (real\ \hat{}\ \text{-})\ set$
shows $\textit{is-interval}\ s \Longrightarrow \textit{connected}\ s$
⟨*proof*⟩

lemma *convex-interval*: $\textit{convex}\ \{a\ ..\ b\}\ \textit{convex}\ \{a{<}..{<}b{::}real\hat{}'n\}$
⟨*proof*⟩

19.35 Another intermediate value theorem formulation.

lemma *ivt-increasing-component-on-1*: **fixes** $f{::}real \Rightarrow real\hat{}'n$
assumes $a \le b$ $\textit{continuous-on}\ \{a\ ..\ b\}\ f\ (f\ a)\$k \le y\ y \le (f\ b)\$k$
shows $\exists x{\in}\{a..b\}.\ (f\ x)\$k = y$
⟨*proof*⟩

lemma *ivt-increasing-component-1*: **fixes** $f{::}real \Rightarrow real\hat{}'n$
shows $a \le b \Longrightarrow \forall x{\in}\{a\ ..\ b\}.\ \textit{continuous}\ (\textit{at}\ x)\ f$
$\Longrightarrow f\ a\$k \le y \Longrightarrow y \le f\ b\$k \Longrightarrow \exists x{\in}\{a..b\}.\ (f\ x)\$k = y$
⟨*proof*⟩

lemma *ivt-decreasing-component-on-1*: **fixes** $f{::}real \Rightarrow real\hat{}'n$
assumes $a \le b$ $\textit{continuous-on}\ \{a\ ..\ b\}\ f\ (f\ b)\$k \le y\ y \le (f\ a)\$k$
shows $\exists x{\in}\{a..b\}.\ (f\ x)\$k = y$
⟨*proof*⟩

lemma *ivt-decreasing-component-1*: **fixes** $f{::}real \Rightarrow real\hat{}'n$
shows $a \le b \Longrightarrow \forall x{\in}\{a\ ..\ b\}.\ \textit{continuous}\ (\textit{at}\ x)\ f$
$\Longrightarrow f\ b\$k \le y \Longrightarrow y \le f\ a\$k \Longrightarrow \exists x{\in}\{a..b\}.\ (f\ x)\$k = y$
⟨*proof*⟩

19.36 A bound within a convex hull, and so an interval.

lemma *convex-on-convex-hull-bound*:
assumes $\textit{convex-on}\ (\textit{convex hull}\ s)\ f\ \forall x{\in}s.\ f\ x \le b$
shows $\forall x{\in}\ \textit{convex hull}\ s.\ f\ x \le b$ ⟨*proof*⟩

lemma *unit-interval-convex-hull*:
$\{0{::}real\hat{}'n\ ..\ 1\} = \textit{convex hull}\ \{x.\ \forall i.\ (x\$i = 0) \lor (x\$i = 1)\}$ (**is** *?int = convex hull ?points*)
⟨*proof*⟩

19.37 And this is a finite set of vertices.

lemma *unit-cube-convex-hull*: **obtains** s **where** *finite* s $\{0 \,..\, 1::real\hat{}'n\} = convex$ $hull\ s$
⟨*proof*⟩

19.38 Hence any cube (could do any nonempty interval).

lemma *cube-convex-hull*:
assumes $0 < d$ **obtains** $s::(real\hat{}'n)\ set$ **where** *finite* s $\{x - (\chi\ i.\ d) \,..\, x + (\chi\ i.\ d)\} = convex\ hull\ s$ ⟨*proof*⟩

19.39 Bounded convex function on open set is continuous.

lemma *convex-on-bounded-continuous*:
fixes $s :: ('a::real\text{-}normed\text{-}vector)\ set$
assumes *open* s *convex-on* $s\ f$ $\forall x \in s.\ abs(f\ x) \le b$
shows *continuous-on* $s\ f$
⟨*proof*⟩

19.40 Upper bound on a ball implies upper and lower bounds.

lemma *scaleR-2*:
fixes $x :: 'a::real\text{-}vector$
shows $scaleR\ 2\ x = x + x$
⟨*proof*⟩

lemma *convex-bounds-lemma*:
fixes $x :: 'a::real\text{-}normed\text{-}vector$
assumes *convex-on* $(cball\ x\ e)\ f$ $\forall y \in cball\ x\ e.\ f\ y \le b$
shows $\forall y \in cball\ x\ e.\ abs(f\ y) \le b + 2 * abs(f\ x)$
⟨*proof*⟩

19.41 Hence a convex function on an open set is continuous.

lemma *convex-on-continuous*:
assumes *open* $(s::(real\hat{}'n)\ set)$ *convex-on* $s\ f$
shows *continuous-on* $s\ f$
⟨*proof*⟩

19.42 Line segments, Starlike Sets, etc.

definition
midpoint $:: 'a::real\text{-}vector \Rightarrow 'a \Rightarrow 'a$ **where**
$midpoint\ a\ b = (inverse\ (2::real)) *_R (a + b)$

definition
open-segment $:: 'a::real\text{-}vector \Rightarrow 'a \Rightarrow 'a\ set$ **where**
$open\text{-}segment\ a\ b = \{(1 - u) *_R a + u *_R b \mid u::real.\ \ 0 < u \land u < 1\}$

definition
closed-segment :: $'a$::*real-vector* $\Rightarrow$ $'a$ $\Rightarrow$ $'a$ *set* **where**
closed-segment a b = $\{(1 - u) *_R a + u *_R b \mid u$::*real*. $0 \leq u \wedge u \leq 1\}$

definition *between* = (λ (a,b). *closed-segment* a b)

lemmas *segment* = *open-segment-def closed-segment-def*

definition *starlike* s $\longleftrightarrow$ ($\exists$ $a$$\in$$s$. $\forall$ $x$$\in$$s$. *closed-segment* a x $\subseteq$ s)

lemma *midpoint-refl*: *midpoint* x x = x
⟨*proof*⟩

lemma *midpoint-sym*: *midpoint* a b = *midpoint* b a ⟨*proof*⟩

lemma *midpoint-eq-iff*: *midpoint* a b = c $\longleftrightarrow$ $a + b = c + c$
⟨*proof*⟩

lemma *dist-midpoint*:
fixes a b :: $'a$::*real-normed-vector* **shows**
dist a (*midpoint* a b) = (*dist* a b) / 2 (**is** *?t1*)
dist b (*midpoint* a b) = (*dist* a b) / 2 (**is** *?t2*)
dist (*midpoint* a b) a = (*dist* a b) / 2 (**is** *?t3*)
dist (*midpoint* a b) b = (*dist* a b) / 2 (**is** *?t4*)
⟨*proof*⟩

lemma *midpoint-eq-endpoint*:
midpoint a b = a $\longleftrightarrow$ $a = b$
midpoint a b = b $\longleftrightarrow$ $a = b$
⟨*proof*⟩

lemma *convex-contains-segment*:
convex s $\longleftrightarrow$ ($\forall$ $a$$\in$$s$. $\forall$ $b$$\in$$s$. *closed-segment* a b $\subseteq$ s)
⟨*proof*⟩

lemma *convex-imp-starlike*:
convex s $\Longrightarrow$ $s \neq \{\}$ $\Longrightarrow$ *starlike* s
⟨*proof*⟩

lemma *segment-convex-hull*:
closed-segment a b = *convex hull* $\{a,b\}$ ⟨*proof*⟩

lemma *convex-segment*: *convex* (*closed-segment* a b)
⟨*proof*⟩

lemma *ends-in-segment*: a $\in$ *closed-segment* a b b $\in$ *closed-segment* a b
⟨*proof*⟩

lemma *segment-furthest-le*:

fixes $a\ b\ x\ y$ :: $real$ ^ $'n$
assumes $x \in closed\text{-}segment\ a\ b$ **shows** $norm(y - x) \leq norm(y - a) \vee norm(y - x) \leq norm(y - b)$ ⟨*proof*⟩

lemma *segment-bound*:
fixes $x\ a\ b$ :: $real$ ^ $'n$
assumes $x \in closed\text{-}segment\ a\ b$
shows $norm(x - a) \leq norm(b - a)$ $norm(x - b) \leq norm(b - a)$
⟨*proof*⟩

lemma *segment-refl*:$closed\text{-}segment\ a\ a = \{a\}$ ⟨*proof*⟩

lemma *between-mem-segment*: $between\ (a,b)\ x \longleftrightarrow x \in closed\text{-}segment\ a\ b$
⟨*proof*⟩

lemma *between*:$between\ (a,b)\ (x::real\hat{}'n) \longleftrightarrow dist\ a\ b = (dist\ a\ x) + (dist\ x\ b)$
⟨*proof*⟩

lemma *between-midpoint*: **fixes** $a::real\hat{}'n$ **shows**
$between\ (a,b)\ (midpoint\ a\ b)$ (**is** *?t1*)
$between\ (b,a)\ (midpoint\ a\ b)$ (**is** *?t2*)
⟨*proof*⟩

lemma *between-mem-convex-hull*:
$between\ (a,b)\ x \longleftrightarrow x \in convex\ hull\ \{a,b\}$
⟨*proof*⟩

19.43 Shrinking towards the interior of a convex set.

lemma *mem-interior-convex-shrink*:
fixes s :: $(real$ ^ $-)\ set$
assumes $convex\ s$ $c \in interior\ s$ $x \in s$ $0 < e$ $e \leq 1$
shows $x - e *_R (x - c) \in interior\ s$
⟨*proof*⟩

lemma *mem-interior-closure-convex-shrink*:
fixes s :: $(real$ ^ $-)\ set$
assumes $convex\ s$ $c \in interior\ s$ $x \in closure\ s$ $0 < e$ $e \leq 1$
shows $x - e *_R (x - c) \in interior\ s$
⟨*proof*⟩

19.44 Some obvious but surprisingly hard simplex lemmas.

lemma *simplex*:
assumes $finite\ s$ $0 \notin s$
shows $convex\ hull\ (insert\ 0\ s) = \{\ y.\ (\exists u.\ (\forall x\in s.\ 0 \leq u\ x) \wedge setsum\ u\ s \leq 1 \wedge setsum\ (\lambda x.\ u\ x *_R x)\ s = y)\}$
⟨*proof*⟩

lemma *std-simplex*:

convex hull (*insert 0* { *basis i* | *i. i*∈*UNIV*}) =
{*x*::*real*ˆ′*n* . (∀ *i. 0* ≤ *x*\$*i*) ∧ *setsum* (λ*i. x*\$*i*) *UNIV* ≤ *1* } (**is** *convex hull* (*insert 0 ?p*) = *?s*)
⟨*proof*⟩

lemma *interior-std-simplex*:
interior (*convex hull* (*insert 0* { *basis i*| *i. i*∈*UNIV*})) =
{*x*::*real*ˆ′*n*. (∀ *i. 0* < *x*\$*i*) ∧ *setsum* (λ*i. x*\$*i*) *UNIV* < *1* }
⟨*proof*⟩

lemma *interior-std-simplex-nonempty*: **obtains** *a*::*real*ˆ′*n* **where**
a ∈ *interior*(*convex hull* (*insert 0* {*basis i* | *i* . *i* ∈ *UNIV*})) ⟨*proof*⟩

end

20 Brouwer-Fixpoint: Results connected with topological dimension.

theory *Brouwer-Fixpoint*
imports *Convex-Euclidean-Space*
begin

lemma *brouwer-compactness-lemma*:
assumes *compact s continuous-on s f* ¬ (∃ *x*∈*s*. (*f x* = (*0*::*real*ˆ′*n*)))
obtains *d* **where** *0* < *d* ∀ *x*∈*s*. *d* ≤ *norm*(*f x*) ⟨*proof*⟩

lemma *kuhn-labelling-lemma*:
assumes (∀ *x*::*real*ˆ-. *P x* ⟶ *P* (*f x*)) ∀ *x*. *P x* ⟶ (∀ *i*. *Q i* ⟶ *0* ≤ *x*\$*i* ∧ *x*\$*i* ≤ *1*)
shows ∃ *l*. (∀ *x i*. *l x i* ≤ (*1*::*nat*)) ∧
(∀ *x i*. *P x* ∧ *Q i* ∧ (*x*\$*i* = *0*) ⟶ (*l x i* = *0*)) ∧
(∀ *x i*. *P x* ∧ *Q i* ∧ (*x*\$*i* = *1*) ⟶ (*l x i* = *1*)) ∧
(∀ *x i*. *P x* ∧ *Q i* ∧ (*l x i* = *0*) ⟶ *x*\$*i* ≤ *f*(*x*)\$*i*) ∧
(∀ *x i*. *P x* ∧ *Q i* ∧ (*l x i* = *1*) ⟶ *f*(*x*)\$*i* ≤ *x*\$*i*) ⟨*proof*⟩

20.1 The key "counting" observation, somewhat abstracted.

lemma *setsum-Un-disjoint*′:**assumes** *finite A finite B A* ∩ *B* = {} *A* ∪ *B* = *C*
shows *setsum g C* = *setsum g A* + *setsum g B*
⟨*proof*⟩

lemma *kuhn-counting-lemma*: **assumes** *finite faces finite simplices*
∀ *f*∈*faces*. *bnd f* ⟶ (*card* {*s* ∈ *simplices*. *face f s*} = *1*)
∀ *f*∈*faces*. ¬ *bnd f* ⟶ (*card* {*s* ∈ *simplices*. *face f s*} = *2*)
∀ *s*∈*simplices*. *compo s* ⟶ (*card* {*f* ∈ *faces*. *face f s* ∧ *compo*′ *f*} = *1*)
∀ *s*∈*simplices*. ¬ *compo s* ⟶ (*card* {*f* ∈ *faces*. *face f s* ∧ *compo*′ *f*} = *0*) ∨
(*card* {*f* ∈ *faces*. *face f s* ∧ *compo*′ *f*} = *2*)

$odd(card\ \{f \in faces.\ compo'\ f \wedge bnd\ f\})$
shows $odd(card\ \{s \in simplices.\ compo\ s\})$ ⟨*proof*⟩

20.2 The odd/even result for faces of complete vertices, generalized.

lemma *card-1-exists*: $card\ s = 1 \longleftrightarrow (\exists!x.\ x \in s)$ ⟨*proof*⟩

lemma *card-2-exists*: $card\ s = 2 \longleftrightarrow (\exists x{\in}s.\ \exists y{\in}s.\ x \neq y \wedge (\forall z{\in}s.\ (z = x) \vee (z = y)))$ ⟨*proof*⟩

lemma *image-lemma-0*: **assumes** $card\ \{a{\in}s.\ f\ \text{‘}\ (s - \{a\}) = t - \{b\}\} = n$
shows $card\ \{s'.\ \exists a{\in}s.\ (s' = s - \{a\}) \wedge (f\ \text{‘}\ s' = t - \{b\})\} = n$ ⟨*proof*⟩

lemma *image-lemma-1*: **assumes** $finite\ s\ finite\ t\ card\ s = card\ t\ f\ \text{‘}\ s = t\ b \in t$
shows $card\ \{s'.\ \exists a{\in}s.\ s' = s - \{a\} \wedge\ f\ \text{‘}\ s' = t - \{b\}\} = 1$ ⟨*proof*⟩

lemma *image-lemma-2*: **assumes** $finite\ s\ finite\ t\ card\ s = card\ t\ f\ \text{‘}\ s \subseteq t\ f\ \text{‘}\ s \neq t\ b \in t$
shows $(card\ \{s'.\ \exists a{\in}s.\ (s' = s - \{a\}) \wedge f\ \text{‘}\ s' = t - \{b\}\} = 0) \vee$
$(card\ \{s'.\ \exists a{\in}s.\ (s' = s - \{a\}) \wedge f\ \text{‘}\ s' = t - \{b\}\} = 2)$ ⟨*proof*⟩

20.3 Combine this with the basic counting lemma.

lemma *kuhn-complete-lemma*:
assumes *finite simplices*
$\forall f\ s.\ face\ f\ s \longleftrightarrow (\exists a{\in}s.\ f = s - \{a\})\ \forall s{\in}simplices.\ card\ s = n + 2\ \forall s{\in}simplices.\ (rl\ \text{‘}\ s) \subseteq \{0..n{+}1\}$
$\forall f{\in}\ \{f.\ \exists s{\in}simplices.\ face\ f\ s\}.\ bnd\ f\ \longrightarrow (card\ \{s{\in}simplices.\ face\ f\ s\} = 1)$
$\forall f{\in}\ \{f.\ \exists s{\in}simplices.\ face\ f\ s\}.\ \neg bnd\ f\ \longrightarrow (card\ \{s{\in}simplices.\ face\ f\ s\} = 2)$
$odd(card\ \{f{\in}\{f.\ \exists s{\in}simplices.\ face\ f\ s\}.\ rl\ \text{‘}\ f = \{0..n\} \wedge bnd\ f\})$
shows $odd\ (card\ \{s{\in}simplices.\ (rl\ \text{‘}\ s = \{0..n{+}1\})\})$
⟨*proof*⟩

20.4 We use the following notion of ordering rather than pointwise indexing.

definition $kle\ n\ x\ y \longleftrightarrow (\exists k{\subseteq}\{1..n{::}nat\}.\ (\forall j.\ y(j) = x(j) + (if\ j \in k\ then\ (1{::}nat)\ else\ 0)))$

lemma *kle-refl*[*intro*]: $kle\ n\ x\ x$ ⟨*proof*⟩

lemma *kle-antisym*: $kle\ n\ x\ y \wedge kle\ n\ y\ x \longleftrightarrow (x = y)$
⟨*proof*⟩

lemma *pointwise-minimal-pointwise-maximal*: **fixes** $s{::}(nat{\Rightarrow}nat)\ set$
assumes $finite\ s\ s \neq \{\}\ \forall x{\in}s.\ \forall y{\in}s.\ (\forall j.\ x\ j \leq y\ j) \vee (\forall j.\ y\ j \leq x\ j)$
shows $\exists a{\in}s.\ \forall x{\in}s.\ \forall j.\ a\ j \leq x\ j\ \exists a{\in}s.\ \forall x{\in}s.\ \forall j.\ x\ j \leq a\ j$
⟨*proof*⟩

lemma *kle-imp-pointwise*: $kle\ n\ x\ y \Longrightarrow (\forall j.\ x\ j \leq y\ j)$ $\langle proof \rangle$

lemma *pointwise-antisym*: **fixes** $x::nat \Rightarrow nat$
shows $(\forall j.\ x\ j \leq y\ j) \wedge (\forall j.\ y\ j \leq x\ j) \longleftrightarrow (x = y)$
$\langle proof \rangle$

lemma *kle-trans*: **assumes** $kle\ n\ x\ y$ $kle\ n\ y\ z$ $kle\ n\ x\ z \vee kle\ n\ z\ x$ **shows** $kle\ n\ x\ z$
$\langle proof \rangle$

lemma *kle-strict*: **assumes** $kle\ n\ x\ y$ $x \neq y$
shows $\forall j.\ x\ j \leq y\ j$ $\exists k.\ 1 \leq k \wedge k \leq n \wedge x(k) < y(k)$
$\langle proof \rangle$

lemma *kle-minimal*: **assumes** *finite s* $s \neq \{\}$ $\forall x\in s.\ \forall y\in s.\ kle\ n\ x\ y \vee kle\ n\ y\ x$
shows $\exists a\in s.\ \forall x\in s.\ kle\ n\ a\ x$ $\langle proof \rangle$

lemma *kle-maximal*: **assumes** *finite s* $s \neq \{\}$ $\forall x\in s.\ \forall y\in s.\ kle\ n\ x\ y \vee kle\ n\ y\ x$
shows $\exists a\in s.\ \forall x\in s.\ kle\ n\ x\ a$ $\langle proof \rangle$

lemma *kle-strict-set*: **assumes** $kle\ n\ x\ y$ $x \neq y$
shows $1 \leq card\ \{k\in\{1..n\}.\ x\ k < y\ k\}$ $\langle proof \rangle$

lemma *kle-range-combine*:
assumes $kle\ n\ x\ y$ $kle\ n\ y\ z$ $kle\ n\ x\ z \vee kle\ n\ z\ x$
$m1 \leq card\ \{k\in\{1..n\}.\ x\ k < y\ k\}$
$m2 \leq card\ \{k\in\{1..n\}.\ y\ k < z\ k\}$
shows $kle\ n\ x\ z \wedge m1 + m2 \leq card\ \{k\in\{1..n\}.\ x\ k < z\ k\}$
$\langle proof \rangle$

lemma *kle-range-combine-l*:
assumes $kle\ n\ x\ y$ $kle\ n\ y\ z$ $kle\ n\ x\ z \vee kle\ n\ z\ x$ $m \leq card\ \{k\in\{1..n\}.\ y(k) < z(k)\}$
shows $kle\ n\ x\ z \wedge m \leq card\ \{k\in\{1..n\}.\ x(k) < z(k)\}$
$\langle proof \rangle$

lemma *kle-range-combine-r*:
assumes $kle\ n\ x\ y$ $kle\ n\ y\ z$ $kle\ n\ x\ z \vee kle\ n\ z\ x$ $m \leq card\ \{k\in\{1..n\}.\ x\ k < y\ k\}$
shows $kle\ n\ x\ z \wedge m \leq card\ \{k\in\{1..n\}.\ x(k) < z(k)\}$
$\langle proof \rangle$

lemma *kle-range-induct*:
assumes $card\ s = Suc\ m$ $\forall x\in s.\ \forall y\in s.\ kle\ n\ x\ y \vee kle\ n\ y\ x$
shows $\exists x\in s.\ \exists y\in s.\ kle\ n\ x\ y \wedge m \leq card\ \{k\in\{1..n\}.\ x\ k < y\ k\}$ $\langle proof \rangle$

lemma *kle-Suc*: $kle\ n\ x\ y \Longrightarrow kle\ (n + 1)\ x\ y$
$\langle proof \rangle$

lemma *kle-trans-1*: **assumes** $kle\ n\ x\ y$ **shows** $x\ j \le y\ j$ $y\ j \le x\ j + 1$
⟨*proof*⟩

lemma *kle-trans-2*: **assumes** $kle\ n\ a\ b$ $kle\ n\ b\ c$ $\forall j.\ c\ j \le a\ j + 1$ **shows** $kle\ n\ a\ c$ ⟨*proof*⟩

lemma *kle-between-r*: **assumes** $kle\ n\ a\ b$ $kle\ n\ b\ c$ $kle\ n\ a\ x$ $kle\ n\ c\ x$ **shows** $kle\ n\ b\ x$
⟨*proof*⟩

lemma *kle-between-l*: **assumes** $kle\ n\ a\ b$ $kle\ n\ b\ c$ $kle\ n\ x\ a$ $kle\ n\ x\ c$ **shows** $kle\ n\ x\ b$
⟨*proof*⟩

lemma *kle-adjacent*:
assumes $\forall j.\ b\ j = (if\ j = k\ then\ a(j) + 1\ else\ a\ j)$ $kle\ n\ a\ x$ $kle\ n\ x\ b$
shows $(x = a) \lor (x = b)$ ⟨*proof*⟩

20.5 kuhn's notion of a simplex (a reformulation to avoid so much indexing).

definition $ksimplex\ p\ n\ (s::(nat \Rightarrow nat)\ set) \longleftrightarrow$
$(card\ s = n + 1 \land$
$(\forall x\in s.\ \forall j.\ x(j) \le p) \land$
$(\forall x\in s.\ \forall j.\ j\notin\{1..n\} \longrightarrow (x\ j = p)) \land$
$(\forall x\in s.\ \forall y\in s.\ kle\ n\ x\ y \lor kle\ n\ y\ x))$

lemma *ksimplexI*: $card\ s = n + 1 \Longrightarrow \forall x\in s.\ \forall j.\ x\ j \le p \Longrightarrow \forall x\in s.\ \forall j.\ j \notin \{1..n\} \longrightarrow x\ j = p \Longrightarrow \forall x\in s.\ \forall y\in s.\ kle\ n\ x\ y \lor kle\ n\ y\ x \Longrightarrow ksimplex\ p\ n\ s$
⟨*proof*⟩

lemma *ksimplex-eq*: $ksimplex\ p\ n\ (s::(nat \Rightarrow nat)\ set) \longleftrightarrow$
$(card\ s = n + 1 \land finite\ s \land$
$(\forall x\in s.\ \forall j.\ x(j) \le p) \land$
$(\forall x\in s.\ \forall j.\ j\notin\{1..n\} \longrightarrow (x\ j = p)) \land$
$(\forall x\in s.\ \forall y\in s.\ kle\ n\ x\ y \lor kle\ n\ y\ x))$
⟨*proof*⟩

lemma *ksimplex-extrema*: **assumes** $ksimplex\ p\ n\ s$ **obtains** $a\ b$ **where** $a \in s$ $b \in s$
$\forall x\in s.\ kle\ n\ a\ x \land kle\ n\ x\ b$ $\forall i.\ b(i) = (if\ i \in \{1..n\}\ then\ a(i) + 1\ else\ a(i))$
⟨*proof*⟩

lemma *ksimplex-extrema-strong*:
assumes $ksimplex\ p\ n\ s$ $n \ne 0$ **obtains** $a\ b$ **where** $a \in s$ $b \in s$ $a \ne b$
$\forall x\in s.\ kle\ n\ a\ x \land kle\ n\ x\ b$ $\forall i.\ b(i) = (if\ i \in \{1..n\}\ then\ a(i) + 1\ else\ a(i))$
⟨*proof*⟩

lemma *ksimplexD*:
assumes *ksimplex p n s*
shows *card s* = *n* + *1 finite s card s* = *n* + *1* $\forall x \in s.\ \forall j.\ x\ j \leq p$ $\forall x \in s.\ \forall j.\ j \notin \{1..n\} \longrightarrow x\ j = p$
$\forall x \in s.\ \forall y \in s.$ *kle n x y* $\vee$ *kle n y x* ⟨*proof*⟩

lemma *ksimplex-successor*:
assumes *ksimplex p n s* $a \in s$
shows $(\forall x \in s.\ kle\ n\ x\ a) \vee (\exists y \in s.\ \exists k \in \{1..n\}.\ \forall j.\ y(j) = (if\ j = k\ then\ a(j) + 1\ else\ a(j)))$
⟨*proof*⟩

lemma *ksimplex-predecessor*:
assumes *ksimplex p n s* $a \in s$
shows $(\forall x \in s.\ kle\ n\ a\ x) \vee (\exists y \in s.\ \exists k \in \{1..n\}.\ \forall j.\ a(j) = (if\ j = k\ then\ y(j) + 1\ else\ y(j)))$
⟨*proof*⟩

20.6 The lemmas about simplices that we need.

lemma *card-funspace'*: **assumes** *finite s finite t card s* = *m card t* = *n*
shows *card* $\{f.\ (\forall x \in s.\ f\ x \in t) \wedge (\forall x \in UNIV - s.\ f\ x = d)\} = n\ \hat{}\ m$ (**is** *card* (*?M s*) = *-*)
⟨*proof*⟩

lemma *card-funspace*: **assumes** *finite s finite t*
shows *card* $\{f.\ (\forall x \in s.\ f\ x \in t) \wedge (\forall x \in UNIV - s.\ f\ x = d)\} = (card\ t)\ \hat{}\ (card\ s)$
⟨*proof*⟩

lemma *finite-funspace*: **assumes** *finite s finite t*
shows *finite* $\{f.\ (\forall x \in s.\ f\ x \in t) \wedge (\forall x \in UNIV - s.\ f\ x = d)\}$ (**is** *finite ?S*)
⟨*proof*⟩

lemma *finite-simplices*: *finite* $\{s.\ ksimplex\ p\ n\ s\}$
⟨*proof*⟩

lemma *simplex-top-face*: **assumes** $0<p$ $\forall x \in f.\ x\ (n + 1) = p$
shows $(\exists s\ a.\ ksimplex\ p\ (n + 1)\ s \wedge a \in s \wedge (f = s - \{a\})) \longleftrightarrow ksimplex\ p\ n\ f$ (**is** *?ls* = *?rs*) ⟨*proof*⟩

lemma *ksimplex-fix-plane*:
assumes $a \in s$ $j \in \{1..n::nat\}$ $\forall x \in s - \{a\}.\ x\ j = q$ $a0 \in s$ $a1 \in s$
$\forall i.\ a1\ i = ((if\ i \in \{1..n\}\ then\ a0\ i + 1\ else\ a0\ i)::nat)$
shows $(a = a0) \vee (a = a1)$ ⟨*proof*⟩

lemma *ksimplex-fix-plane-0*:
assumes $a \in s$ $j \in \{1..n::nat\}$ $\forall x \in s - \{a\}.\ x\ j = 0$ $a0 \in s$ $a1 \in s$
$\forall i.\ a1\ i = ((if\ i \in \{1..n\}\ then\ a0\ i + 1\ else\ a0\ i)::nat)$

shows $a = a1$ ⟨*proof*⟩

lemma *ksimplex-fix-plane-p*:
assumes $ksimplex\ p\ n\ s\ a \in s\ j{\in}\{1..n\}\ \forall x{\in}s - \{a\}.\ x\ j = p\ a0 \in s\ a1 \in s$
$\forall i.\ a1\ i = (if\ i{\in}\{1..n\}\ then\ a0\ i + 1\ else\ a0\ i)$
shows $a = a0$ ⟨*proof*⟩

lemma *ksimplex-replace-0*:
assumes $ksimplex\ p\ n\ s\ a \in s\ n \neq 0\ j{\in}\{1..n\}\ \forall x{\in}s - \{a\}.\ x\ j = 0$
shows $card\ \{s'.\ ksimplex\ p\ n\ s' \wedge (\exists b{\in}s'.\ s' - \{b\} = s - \{a\})\} = 1$ ⟨*proof*⟩

lemma *ksimplex-replace-1*:
assumes $ksimplex\ p\ n\ s\ a \in s\ n \neq 0\ j{\in}\{1..n\}\ \forall x{\in}s - \{a\}.\ x\ j = p$
shows $card\ \{s'.\ ksimplex\ p\ n\ s' \wedge (\exists b{\in}s'.\ s' - \{b\} = s - \{a\})\} = 1$ ⟨*proof*⟩

lemma *ksimplex-replace-2*:
assumes $ksimplex\ p\ n\ s\ a \in s\ n \neq 0\ {\sim}(\exists j{\in}\{1..n\}.\ \forall x{\in}s - \{a\}.\ x\ j = 0)$ ${\sim}(\exists j{\in}\{1..n\}.\ \forall x{\in}s - \{a\}.\ x\ j = p)$
shows $card\ \{s'.\ ksimplex\ p\ n\ s' \wedge (\exists b{\in}s'.\ s' - \{b\} = s - \{a\})\} = 2$ (**is** $card\ ?A = 2$) ⟨*proof*⟩

20.7 Hence another step towards concreteness.

lemma *kuhn-simplex-lemma*:
assumes $\forall s.\ ksimplex\ p\ (n + 1)\ s \longrightarrow (rl\ `\ s \subseteq \{0..n+1\})$
$odd\ (card\{f.\ \exists s\ a.\ ksimplex\ p\ (n + 1)\ s \wedge a \in s \wedge (f = s - \{a\}) \wedge$
$(rl\ `\ f = \{0\ ..\ n\}) \wedge ((\exists j{\in}\{1..n+1\}.\forall x{\in}f.\ x\ j = 0) \vee (\exists j{\in}\{1..n+1\}.\forall x{\in}f.\ x\ j = p))\})$
shows $odd(card\ \{s{\in}\{s.\ ksimplex\ p\ (n + 1)\ s\}.\ rl\ `\ s = \{0..n+1\}\ \})$ ⟨*proof*⟩

20.8 Reduced labelling.

definition *reduced label* $(n{::}nat)\ (x{::}nat{\Rightarrow}nat) =$
$(SOME\ k.\ k \leq n \wedge (\forall i.\ 1{\leq}i \wedge i{<}k{+}1 \longrightarrow label\ x\ i = 0) \wedge (k = n \vee label\ x\ (k + 1) \neq (0{::}nat)))$

lemma *reduced-labelling*: **shows** $reduced\ label\ n\ x \leq n$ (**is** $?t1$) **and**
$\forall i.\ 1{\leq}i \wedge i < reduced\ label\ n\ x + 1 \longrightarrow (label\ x\ i = 0)$ (**is** $?t2$)
$(reduced\ label\ n\ x = n) \vee (label\ x\ (reduced\ label\ n\ x + 1) \neq 0)$ (**is** $?t3$) ⟨*proof*⟩

lemma *reduced-labelling-unique*: **fixes** $x{::}nat \Rightarrow nat$
assumes $r \leq n\ \ \forall i.\ 1 \leq i \wedge i < r + 1 \longrightarrow (label\ x\ i = 0)\ (r = n) \vee (label\ x\ (r + 1) \neq 0)$
shows $reduced\ label\ n\ x = r$ ⟨*proof*⟩

lemma *reduced-labelling-0*: **assumes** $j{\in}\{1..n\}\ label\ x\ j = 0$ **shows** $reduced\ label\ n\ x \neq j - 1$
⟨*proof*⟩

lemma *reduced-labelling-1*: **assumes** $j \in \{1..n\}$ *label* $x\ j \neq 0$ **shows** *reduced label* $n\ x < j$
⟨*proof*⟩

lemma *reduced-labelling-Suc*:
assumes *reduced lab* $(n + 1)\ x \neq n + 1$ **shows** *reduced lab* $(n + 1)\ x =$ *reduced lab* $n\ x$
⟨*proof*⟩

lemma *complete-face-top*:
assumes $\forall x \in f.\ \forall j \in \{1..n+1\}.\ x\ j = 0 \longrightarrow lab\ x\ j = 0$
$\forall x \in f.\ \forall j \in \{1..n+1\}.\ x\ j = p \longrightarrow lab\ x\ j = 1$
shows $((reduced\ lab\ (n + 1))\ `\ f = \{0..n\}) \wedge ((\exists j \in \{1..n+1\}.\ \forall x \in f.\ x\ j = 0) \vee (\exists j \in \{1..n+1\}.\ \forall x \in f.\ x\ j = p)) \longleftrightarrow$
$((reduced\ lab\ (n + 1))\ `\ f = \{0..n\}) \wedge (\forall x \in f.\ x\ (n + 1) = p)$ (**is** $?l = ?r$)
⟨*proof*⟩

20.9 Hence we get just about the nice induction.

lemma *kuhn-induction*:
assumes $0 < p$ $\forall x.\ \forall j \in \{1..n+1\}.\ (\forall j.\ x\ j \leq p) \wedge (x\ j = 0) \longrightarrow (lab\ x\ j = 0)$
$\forall x.\ \forall j \in \{1..n+1\}.\ (\forall j.\ x\ j \leq p) \wedge (x\ j = p) \longrightarrow (lab\ x\ j = 1)$
$odd\ (card\ \{f.\ ksimplex\ p\ n\ f \wedge ((reduced\ lab\ n)\ `\ f = \{0..n\})\})$
shows $odd\ (card\ \{s.\ ksimplex\ p\ (n+1)\ s \wedge ((reduced\ lab\ (n+1))\ `\ s = \{0..n+1\})\})$
⟨*proof*⟩

lemma *kuhn-induction-Suc*:
assumes $0 < p$ $\forall x.\ \forall j \in \{1..Suc\ n\}.\ (\forall j.\ x\ j \leq p) \wedge (x\ j = 0) \longrightarrow (lab\ x\ j = 0)$
$\forall x.\ \forall j \in \{1..Suc\ n\}.\ (\forall j.\ x\ j \leq p) \wedge (x\ j = p) \longrightarrow (lab\ x\ j = 1)$
$odd\ (card\ \{f.\ ksimplex\ p\ n\ f \wedge ((reduced\ lab\ n)\ `\ f = \{0..n\})\})$
shows $odd\ (card\ \{s.\ ksimplex\ p\ (Suc\ n)\ s \wedge ((reduced\ lab\ (Suc\ n))\ `\ s = \{0..Suc\ n\})\})$
⟨*proof*⟩

20.10 And so we get the final combinatorial result.

lemma *ksimplex-0*: $ksimplex\ p\ 0\ s \longleftrightarrow s = \{(\lambda x.\ p)\}$ (**is** $?l = ?r$) ⟨*proof*⟩

lemma *reduce-labelling-0*[*simp*]: $reduced\ lab\ 0\ x = 0$ ⟨*proof*⟩

lemma *kuhn-combinatorial*:
assumes $0 < p$ $\forall x\ j.\ (\forall j.\ x(j) \leq p) \wedge 1 \leq j \wedge j \leq n \wedge (x\ j = 0) \longrightarrow (lab\ x\ j = 0)$
$\forall x\ j.\ (\forall j.\ x(j) \leq p) \wedge 1 \leq j \wedge j \leq n \wedge (x\ j = p) \longrightarrow (lab\ x\ j = 1)$
shows $odd\ (card\ \{s.\ ksimplex\ p\ n\ s \wedge ((reduced\ lab\ n)\ `\ s = \{0..n\})\})$ ⟨*proof*⟩

lemma *kuhn-lemma*: **assumes** $0 < (p::nat)$ $0 < (n::nat)$
$\forall x.\ (\forall i \in \{1..n\}.\ x\ i \leq p) \longrightarrow (\forall i \in \{1..n\}.\ (label\ x\ i = (0::nat)) \vee (label\ x\ i = 1))$

$\forall x.\ (\forall i{\in}\{1..n\}.\ x\ i \le p) \longrightarrow (\forall i{\in}\{1..n\}.\ (x\ i = 0) \longrightarrow (label\ x\ i = 0))$
$\forall x.\ (\forall i{\in}\{1..n\}.\ x\ i \le p) \longrightarrow (\forall i{\in}\{1..n\}.\ (x\ i = p) \longrightarrow (label\ x\ i = 1))$
obtains q **where** $\forall i{\in}\{1..n\}.\ q\ i < p$
$\forall i{\in}\{1..n\}.\ \exists r\ s.\ (\forall j{\in}\{1..n\}.\ q(j) \le r(j) \wedge r(j) \le q(j) + 1) \wedge$
$(\forall j{\in}\{1..n\}.\ q(j) \le s(j) \wedge s(j) \le q(j) + 1) \wedge$
$\sim(label\ r\ i = label\ s\ i)$ $\langle proof\rangle$

20.11 The main result for the unit cube.

lemma *kuhn-labelling-lemma′*:
assumes $(\forall x{::}nat{\Rightarrow}real.\ P\ x \longrightarrow P\ (f\ x))$ $\forall x.\ P\ x \longrightarrow (\forall i{::}nat.\ Q\ i \longrightarrow 0 \le x\ i \wedge x\ i \le 1)$
shows $\exists l.\ (\forall x\ i.\ l\ x\ i \le (1{::}nat)) \wedge$
$(\forall x\ i.\ P\ x \wedge Q\ i \wedge (x\ i = 0) \longrightarrow (l\ x\ i = 0)) \wedge$
$(\forall x\ i.\ P\ x \wedge Q\ i \wedge (x\ i = 1) \longrightarrow (l\ x\ i = 1)) \wedge$
$(\forall x\ i.\ P\ x \wedge Q\ i \wedge (l\ x\ i = 0) \longrightarrow x\ i \le f(x)\ i) \wedge$
$(\forall x\ i.\ P\ x \wedge Q\ i \wedge (l\ x\ i = 1) \longrightarrow f(x)\ i \le x\ i)$ $\langle proof\rangle$

lemma *brouwer-cube*: **fixes** $f{::}real\hat{\ }'n \Rightarrow real\hat{\ }'n$
assumes *continuous-on* $\{0..1\}$ f f ‘ $\{0..1\} \subseteq \{0..1\}$
shows $\exists x{\in}\{0..1\}.\ f\ x = x$ $\langle proof\rangle$

20.12 Retractions.

definition *retraction* $s\ t\ (r{::}real\hat{\ }'n{\Rightarrow}real\hat{\ }'n) \longleftrightarrow$
$t \subseteq s \wedge$ *continuous-on* $s\ r \wedge (r$ ‘ $s \subseteq t) \wedge (\forall x{\in}t.\ r\ x = x)$

definition *retract-of* (**infixl** *retract′-of 12*) **where**
$(t$ *retract-of* $s) \longleftrightarrow (\exists r.$ *retraction* $s\ t\ r)$

lemma *retraction-idempotent*: *retraction* $s\ t\ r \Longrightarrow x \in s \Longrightarrow r(r\ x) = r\ x$
$\langle proof\rangle$

20.13 preservation of fixpoints under (more general notion of) retraction.

lemma *invertible-fixpoint-property*: **fixes** $s{::}(real\hat{\ }'n)$ *set* **and** $t{::}(real\hat{\ }'m)$ *set*
assumes *continuous-on* $t\ i\ i$ ‘ $t \subseteq s$ *continuous-on* $s\ r\ r$ ‘ $s \subseteq t$ $\forall y{\in}t.\ r\ (i\ y) = y$
$\forall f.$ *continuous-on* $s\ f \wedge f$ ‘ $s \subseteq s \longrightarrow (\exists x{\in}s.\ f\ x = x)$ *continuous-on* $t\ g\ g$ ‘ $t \subseteq t$
obtains y **where** $y{\in}t$ $g\ y = y$ $\langle proof\rangle$

lemma *homeomorphic-fixpoint-property*:
fixes $s{::}(real\hat{\ }'n)$ *set* **and** $t{::}(real\hat{\ }'m)$ *set* **assumes** s *homeomorphic* t
shows $(\forall f.$ *continuous-on* $s\ f \wedge f$ ‘ $s \subseteq s \longrightarrow (\exists x{\in}s.\ f\ x = x)) \longleftrightarrow$
$(\forall g.$ *continuous-on* $t\ g \wedge g$ ‘ $t \subseteq t \longrightarrow (\exists y{\in}t.\ g\ y = y))$ $\langle proof\rangle$

lemma *retract-fixpoint-property*:

assumes *t retract-of s* $\forall f.$ *continuous-on s f* $\wedge$ *f ' s* $\subseteq$ *s* $\longrightarrow$ ($\exists x \in s.$ *f x* = *x*)
continuous-on t g g ' t $\subseteq$ *t*
obtains *y* **where** *y* $\in$ *t g y* = *y* ⟨*proof*⟩

20.14 So the Brouwer theorem for any set with nonempty interior.

lemma *brouwer-weak*: **fixes** *f*::*real^'n* $\Rightarrow$ *real^'n*
assumes *compact s convex s interior s* $\neq$ {} *continuous-on s f f ' s* $\subseteq$ *s*
obtains *x* **where** *x* $\in$ *s f x* = *x* ⟨*proof*⟩

20.15 And in particular for a closed ball.

lemma *brouwer-ball*: **fixes** *f*::*real^'n* $\Rightarrow$ *real^'n*
assumes *0* < *e continuous-on* (*cball a e*) *f f '* (*cball a e*) $\subseteq$ (*cball a e*)
obtains *x* **where** *x* $\in$ *cball a e f x* = *x*
⟨*proof*⟩

Still more general form; could derive this directly without using the rather involved *HOMEOMORPHIC-CONVEX-COMPACT* theorem, just using a scaling and translation to put the set inside the unit cube.

lemma *brouwer*: **fixes** *f*::*real^'n* $\Rightarrow$ *real^'n*
assumes *compact s convex s s* $\neq$ {} *continuous-on s f f ' s* $\subseteq$ *s*
obtains *x* **where** *x* $\in$ *s f x* = *x* ⟨*proof*⟩

So we get the no-retraction theorem.

lemma *no-retraction-cball*: **assumes** *0* < *e*
shows ¬ (*frontier*(*cball a e*) *retract-of* (*cball a e*)) ⟨*proof*⟩

20.16 Bijections between intervals.

definition *interval-bij* = (λ (*a*,*b*) (*u*,*v*) (*x*::*real^'n*).
(χ *i*. *u$i* + (*x$i* − *a$i*) / (*b$i* − *a$i*) ∗ (*v$i* − *u$i*))::*real^'n*)

lemma *interval-bij-affine*:
interval-bij (*a*,*b*) (*u*,*v*) = (λ*x*. (χ *i*. (*v$i* − *u$i*) / (*b$i* − *a$i*) ∗ *x$i*) +
(χ *i*. *u$i* − (*v$i* − *u$i*) / (*b$i* − *a$i*) ∗ *a$i*))
⟨*proof*⟩

lemma *continuous-interval-bij*:
continuous (*at x*) (*interval-bij* (*a*,*b*::*real^'n*) (*u*,*v*))
⟨*proof*⟩

lemma *continuous-on-interval-bij*: *continuous-on s* (*interval-bij* (*a*,*b*) (*u*,*v*))
⟨*proof*⟩

lemma *divide-nonneg-nonneg*:**assumes** *a* ≥ *0 b* ≥ *0* **shows** *0* ≤ *a* / (*b*::*real*)
⟨*proof*⟩

lemma *in-interval-interval-bij*: **assumes** $x \in \{a..b\}$ $\{u..v\} \neq \{\}$
shows *interval-bij* (a,b) (u,v) $x \in \{u..v::real^{\wedge\prime}n\}$
$\langle proof \rangle$

lemma *interval-bij-bij*: **assumes** $\forall i.\ a\$i < b\$i \wedge u\$i < v\$i$
shows *interval-bij* (a,b) (u,v) $($*interval-bij* (u,v) (a,b) $x) = x$
$\langle proof \rangle$

end

21 Operator-Norm: Operator Norm

theory *Operator-Norm*
imports *Euclidean-Space*
begin

definition *onorm f* $=$ *Sup* $\{norm\ (f\ x)|\ x.\ norm\ x = 1\}$

lemma *norm-bound-generalize*:
fixes f:: *real* $^{\wedge\prime}n \Rightarrow real^{\wedge\prime}m$
assumes *lf*: *linear f*
shows $(\forall x.\ norm\ x = 1 \longrightarrow norm\ (f\ x) \leq b) \longleftrightarrow (\forall x.\ norm\ (f\ x) \leq b * norm$ $x)$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
$\langle proof \rangle$

lemma *onorm*:
fixes f:: *real* $^{\wedge\prime}n \Rightarrow real\ ^{\wedge\prime}m$
assumes *lf*: *linear f*
shows *norm* $(f\ x) <= onorm\ f * norm\ x$
and $\forall x.\ norm\ (f\ x) <= b * norm\ x \Longrightarrow onorm\ f <= b$
$\langle proof \rangle$

lemma *onorm-pos-le*: **assumes** *lf*: *linear* $(f::real\ ^{\wedge\prime}n \Rightarrow real\ ^{\wedge\prime}m)$ **shows** $0 <=$ *onorm f*
$\langle proof \rangle$

lemma *onorm-eq-0*: **assumes** *lf*: *linear* $(f::real\ ^{\wedge\prime}n \Rightarrow real\ ^{\wedge\prime}m)$
shows *onorm f* $= 0 \longleftrightarrow (\forall x.\ f\ x = 0)$
$\langle proof \rangle$

lemma *onorm-const*: $onorm(\lambda x::real^{\wedge\prime}n.\ (y::real\ ^{\wedge\prime}m)) = norm\ y$
$\langle proof \rangle$

lemma *onorm-pos-lt*: **assumes** *lf*: *linear* $(f::real\ ^{\wedge}\ {}^{\prime}n \Rightarrow real\ ^{\wedge\prime}m)$
shows $0 <$ *onorm f* $\longleftrightarrow {}^{\sim}(\forall x.\ f\ x = 0)$
$\langle proof \rangle$

lemma *onorm-compose*:
  assumes *lf*: *linear* $(f::real\ \hat{}'n \Rightarrow real\ \hat{}'m)$
  and *lg*: *linear* $(g::real\hat{}'k \Rightarrow real\hat{}'n)$
  shows *onorm* $(f\ o\ g)$ <= *onorm* $f * $ *onorm* g
  ⟨*proof*⟩

lemma *onorm-neg-lemma*: **assumes** *lf*: *linear* $(f::real\ \hat{}'n \Rightarrow real\hat{}'m)$
  shows *onorm* $(\lambda x. - f\ x) \leq$ *onorm* f
  ⟨*proof*⟩

lemma *onorm-neg*: **assumes** *lf*: *linear* $(f::real\ \hat{}'n \Rightarrow real\hat{}'m)$
  shows *onorm* $(\lambda x. - f\ x) =$ *onorm* f
  ⟨*proof*⟩

lemma *onorm-triangle*:
  assumes *lf*: *linear* $(f::real\ \hat{}'n \Rightarrow real\ \hat{}'m)$ **and** *lg*: *linear* g
  shows *onorm* $(\lambda x.\ f\ x + g\ x)$ <= *onorm* f + *onorm* g
  ⟨*proof*⟩

lemma *onorm-triangle-le*: *linear* $(f::real\ \hat{}'n \Rightarrow real\ \hat{}'m) \Longrightarrow$ *linear* $g \Longrightarrow$ *onorm*(f) + *onorm*(g) <= e
  $\Longrightarrow$ *onorm*$(\lambda x.\ f\ x + g\ x)$ <= e
  ⟨*proof*⟩

lemma *onorm-triangle-lt*: *linear* $(f::real\ \hat{}'n \Rightarrow real\ \hat{}'m) \Longrightarrow$ *linear* $g \Longrightarrow$ *onorm*(f) + *onorm*(g) < e
  ==> *onorm*$(\lambda x.\ f\ x + g\ x)$ < e
  ⟨*proof*⟩

end

22 Derivative: Multivariate calculus in Euclidean space.

theory *Derivative*
imports *Brouwer-Fixpoint Vec1 RealVector Operator-Norm*
begin

lemmas *linear-linear* = *linear-conv-bounded-linear*[*THEN sym*]

22.1 Derivatives

The definition is slightly tricky since we make it work over nets of a particular form. This lets us prove theorems generally and use "at a" or "at a within s" for restriction to a set (1-sided on R etc.)

definition *has-derivative* :: $('a::real\text{-}normed\text{-}vector \Rightarrow 'b::real\text{-}normed\text{-}vector) \Rightarrow ('a \Rightarrow 'b) \Rightarrow ('a\ net \Rightarrow bool)$
(**infixl** $(has'\text{-}derivative)$ 12) **where**
$(f\ has\text{-}derivative\ f')\ net \equiv bounded\text{-}linear\ f' \wedge ((\lambda y.\ (1\ /\ (norm\ (y - netlimit\ net)))\ *_R$
$(f\ y - (f\ (netlimit\ net) + f'(y - netlimit\ net)))) ---> 0)\ net$

lemma *derivative-linear*[*dest*]:$(f\ has\text{-}derivative\ f')\ net \Longrightarrow bounded\text{-}linear\ f'$
$\langle proof \rangle$

lemma *DERIV-conv-has-derivative*:$(DERIV\ f\ x :> f') = (f\ has\text{-}derivative\ op *\ f')\ (at\ (x::real))$ (**is** $?l = ?r$) $\langle proof \rangle$

lemma *FDERIV-conv-has-derivative*:$FDERIV\ f\ (x::'a::\{real\text{-}normed\text{-}vector,perfect\text{-}space\}) :> f' = (f\ has\text{-}derivative\ f')\ (at\ x)$ (**is** $?l = ?r$) $\langle proof \rangle$

22.2 These are the only cases we'll care about, probably.

lemma *has-derivative-within*: $(f\ has\text{-}derivative\ f')\ (at\ x\ within\ s) \longleftrightarrow$
$bounded\text{-}linear\ f' \wedge ((\lambda y.\ (1\ /\ norm(y - x))\ *_R\ (f\ y - (f\ x + f'\ (y - x))))$
$---> 0)\ (at\ x\ within\ s)$
$\langle proof \rangle$

lemma *has-derivative-at*: $(f\ has\text{-}derivative\ f')\ (at\ x) \longleftrightarrow$
$bounded\text{-}linear\ f' \wedge ((\lambda y.\ (1\ /\ (norm(y - x)))\ *_R\ (f\ y - (f\ x + f'\ (y - x)))) ---> 0)\ (at\ x)$
$\langle proof \rangle$

22.3 More explicit epsilon-delta forms.

lemma *has-derivative-within'*:
$(f\ has\text{-}derivative\ f')(at\ x\ within\ s) \longleftrightarrow bounded\text{-}linear\ f' \wedge$
$(\forall e>0.\ \exists d>0.\ \forall x'\in s.\ 0 < norm(x' - x) \wedge norm(x' - x) < d$
$\longrightarrow norm(f\ x' - f\ x - f'(x' - x))\ /\ norm(x' - x) < e)$
$\langle proof \rangle$

lemma *has-derivative-at'*:
$(f\ has\text{-}derivative\ f')\ (at\ x) \longleftrightarrow bounded\text{-}linear\ f' \wedge$
$(\forall e>0.\ \exists d>0.\ \forall x'.\ 0 < norm(x' - x) \wedge norm(x' - x) < d$
$\longrightarrow norm(f\ x' - f\ x - f'(x' - x))\ /\ norm(x' - x) < e)$
$\langle proof \rangle$

lemma *has-derivative-at-within*: $(f\ has\text{-}derivative\ f')\ (at\ x) \Longrightarrow (f\ has\text{-}derivative\ f')\ (at\ x\ within\ s)$
$\langle proof \rangle$

lemma *has-derivative-within-open*:
$a \in s \Longrightarrow open\ s \Longrightarrow ((f\ has\text{-}derivative\ f')\ (at\ a\ within\ s) \longleftrightarrow (f\ has\text{-}derivative\ f')\ (at\ a))$
$\langle proof \rangle$

22.4 Derivatives on real = Derivatives on (*real*, *1*) *cart*

lemma *has-derivative-within-vec1-dest-vec1*: **fixes** *f*::*real*⇒*real* **shows**
((*vec1* ∘ *f* ∘ *dest-vec1*) *has-derivative* (*vec1* ∘ *f′* ∘ *dest-vec1*)) (*at* (*vec1 x*) *within* *vec1* ‘ *s*)
= (*f has-derivative f′*) (*at x within s*)
⟨*proof*⟩

lemma *has-derivative-at-vec1-dest-vec1*: **fixes** *f*::*real*⇒*real* **shows**
((*vec1* ∘ *f* ∘ *dest-vec1*) *has-derivative* (*vec1* ∘ *f′* ∘ *dest-vec1*)) (*at* (*vec1 x*)) = (*f has-derivative f′*) (*at x*)
⟨*proof*⟩

lemma *bounded-linear-vec1*: **fixes** *f*::′*a*::*real-normed-vector*⇒*real*
shows *bounded-linear f* = *bounded-linear* (*vec1* ∘ *f*)
⟨*proof*⟩

lemma *bounded-linear-dest-vec1*: **fixes** *f*::*real*⇒′*a*::*real-normed-vector*
shows *bounded-linear f* = *bounded-linear* (*f* ∘ *dest-vec1*)
⟨*proof*⟩

lemma *has-derivative-at-vec1*: **fixes** *f*::′*a*::*real-normed-vector*⇒*real* **shows**
(*f has-derivative f′*) (*at x*) = ((*vec1* ∘ *f*) *has-derivative* (*vec1* ∘ *f′*)) (*at x*)
⟨*proof*⟩

lemma *has-derivative-within-dest-vec1*:**fixes** *f*::*real*⇒′*a*::*real-normed-vector* **shows**
((*f* ∘ *dest-vec1*) *has-derivative* (*f′* ∘ *dest-vec1*)) (*at* (*vec1 x*) *within vec1* ‘ *s*) = (*f has-derivative f′*) (*at x within s*)
⟨*proof*⟩

lemma *has-derivative-at-dest-vec1*:**fixes** *f*::*real*⇒′*a*::*real-normed-vector* **shows**
((*f* ∘ *dest-vec1*) *has-derivative* (*f′* ∘ *dest-vec1*)) (*at* (*vec1 x*)) = (*f has-derivative f′*) (*at x*)
⟨*proof*⟩

lemma *derivative-is-linear*: **fixes** *f*::*real*^′*a* ⇒ *real*^′*b* **shows**
(*f has-derivative f′*) *net* ⟹ *linear f′*
⟨*proof*⟩

22.5 Combining theorems.

lemma (**in** *bounded-linear*) *has-derivative*: (*f has-derivative f*) *net*
⟨*proof*⟩

lemma *has-derivative-id*: ((λ*x*. *x*) *has-derivative* (λ*h*. *h*)) *net*
⟨*proof*⟩

lemma *has-derivative-const*: ((λ*x*. *c*) *has-derivative* (λ*h*. *0*)) *net*
⟨*proof*⟩

lemma (**in** *bounded-linear*) *cmul*: **shows** *bounded-linear* $(\lambda x.\ (c::real) *_R f\ x)$
⟨*proof*⟩

lemma *has-derivative-cmul*: **assumes** $(f$ *has-derivative* $f')$ *net* **shows** $((\lambda x.\ c *_R f(x))$ *has-derivative* $(\lambda h.\ c *_R f'(h)))$ *net*
⟨*proof*⟩

lemma *has-derivative-cmul-eq*: **assumes** $c \neq 0$
shows $(((\lambda x.\ c *_R f(x))$ *has-derivative* $(\lambda h.\ c *_R f'(h)))$ *net* $\longleftrightarrow$ $(f$ *has-derivative* $f')$ *net*$)$
⟨*proof*⟩

lemma *has-derivative-neg*:
$(f$ *has-derivative* $f')$ *net* $\Longrightarrow$ $((\lambda x.\ -(f\ x))$ *has-derivative* $(\lambda h.\ -(f'\ h)))$ *net*
⟨*proof*⟩

lemma *has-derivative-neg-eq*: $((\lambda x.\ -(f\ x))$ *has-derivative* $(\lambda h.\ -(f'\ h)))$ *net* $\longleftrightarrow$ $(f$ *has-derivative* $f')$ *net*
⟨*proof*⟩

lemma *has-derivative-add*: **assumes** $(f$ *has-derivative* $f')$ *net* $(g$ *has-derivative* $g')$ *net*
shows $((\lambda x.\ f(x) + g(x))$ *has-derivative* $(\lambda h.\ f'(h) + g'(h)))$ *net* ⟨*proof*⟩

lemma *has-derivative-add-const*:$(f$ *has-derivative* $f')$ *net* $\Longrightarrow$ $((\lambda x.\ f\ x + c)$ *has-derivative* $f')$ *net*
⟨*proof*⟩

lemma *has-derivative-sub*:
$(f$ *has-derivative* $f')$ *net* $\Longrightarrow$ $(g$ *has-derivative* $g')$ *net* $\Longrightarrow$ $((\lambda x.\ f(x) - g(x))$ *has-derivative* $(\lambda h.\ f'(h) - g'(h)))$ *net*
⟨*proof*⟩

lemma *has-derivative-setsum*: **assumes** *finite* s $\forall a{\in}s.\ ((f\ a)$ *has-derivative* $(f'\ a))$ *net*
shows $((\lambda x.\ setsum\ (\lambda a.\ f\ a\ x)\ s)$ *has-derivative* $(\lambda h.\ setsum\ (\lambda a.\ f'\ a\ h)\ s))$ *net*
⟨*proof*⟩

lemma *has-derivative-setsum-numseg*:
$\forall i.\ m \le i \wedge i \le n \longrightarrow ((f\ i)$ *has-derivative* $(f'\ i))$ *net* $\Longrightarrow$
$((\lambda x.\ setsum\ (\lambda i.\ f\ i\ x)\ \{m..n::nat\})$ *has-derivative* $(\lambda h.\ setsum\ (\lambda i.\ f'\ i\ h)\ \{m..n\}))$ *net*
⟨*proof*⟩

22.6 somewhat different results for derivative of scalar multiplier.

lemma *has-derivative-vmul-component*: **fixes** $c::real\hat{\ }'a \Rightarrow real\hat{\ }'b$ **and** $v::real\hat{\ }'c$

assumes $(c$ *has-derivative* $c')$ *net*
shows $((\lambda x.\ c(x)\$k *_R v)$ *has-derivative* $(\lambda x.\ (c'\ x)\$k *_R v))$ *net* ⟨*proof*⟩

lemma *has-derivative-vmul-within*: **fixes** c::*real* ⇒ *real* **and** v::*real*^'*a*
assumes $(c$ *has-derivative* $c')$ (*at x within s*)
shows $((\lambda x.\ (c\ x) *_R v)$ *has-derivative* $(\lambda x.\ (c'\ x) *_R v))$ (*at x within s*) ⟨*proof*⟩

lemma *has-derivative-vmul-at*: **fixes** c::*real* ⇒ *real* **and** v::*real*^'*a*
assumes $(c$ *has-derivative* $c')$ (*at x*)
shows $((\lambda x.\ (c\ x) *_R v)$ *has-derivative* $(\lambda x.\ (c'\ x) *_R v))$ (*at x*)
⟨*proof*⟩

lemma *has-derivative-lift-dot*:
assumes (*f has-derivative f'*) *net*
shows $((\lambda x.$ *inner v* $(f\ x))$ *has-derivative* $(\lambda t.$ *inner v* $(f'\ t)))$ *net* ⟨*proof*⟩

lemmas *has-derivative-intros* = *has-derivative-sub has-derivative-add has-derivative-cmul has-derivative-id has-derivative-const*
has-derivative-neg has-derivative-vmul-component has-derivative-vmul-at has-derivative-vmul-within has-derivative-cmul
bounded-linear.has-derivative has-derivative-lift-dot

22.7 limit transformation for derivatives.

lemma *has-derivative-transform-within*:
assumes $0 < d$ $x \in s$ $\forall x'\in s.$ *dist* x' $x < d \longrightarrow f\ x' = g\ x'$ (*f has-derivative f'*) (*at x within s*)
shows (*g has-derivative f'*) (*at x within s*)
⟨*proof*⟩

lemma *has-derivative-transform-at*:
assumes $0 < d$ $\forall x'.$ *dist* x' $x < d \longrightarrow f\ x' = g\ x'$ (*f has-derivative f'*) (*at x*)
shows (*g has-derivative f'*) (*at x*)
⟨*proof*⟩

lemma *has-derivative-transform-within-open*:
assumes *open s* $x \in s$ $\forall y\in s.$ $f\ y = g\ y$ (*f has-derivative f'*) (*at x*)
shows (*g has-derivative f'*) (*at x*)
⟨*proof*⟩

22.8 differentiability.

no-notation *Deriv.differentiable* (**infixl** *differentiable 60*)

definition *differentiable* :: ('*a*::*real-normed-vector* ⇒ '*b*::*real-normed-vector*) ⇒ '*a net* ⇒ *bool* (**infixr** *differentiable 30*) **where**
f differentiable net ≡ (∃ *f'*. (*f has-derivative f'*) *net*)

definition *differentiable-on* :: ('*a*::*real-normed-vector* ⇒ '*b*::*real-normed-vector*) ⇒ '*a set* ⇒ *bool* (**infixr** *differentiable'-on 30*) **where**

f differentiable-on s $\equiv$ ($\forall\, x{\in}s$. *f differentiable* (*at x within s*))

lemma *differentiableI*: (*f has-derivative* f') *net* $\Longrightarrow$ *f differentiable net*
⟨*proof*⟩

lemma *differentiable-at-withinI*: *f differentiable* (*at x*) $\Longrightarrow$ *f differentiable* (*at x within s*)
⟨*proof*⟩

lemma *differentiable-within-open*: **assumes** $a \in s$ *open s* **shows**
f differentiable (*at a within s*) $\longleftrightarrow$ (*f differentiable* (*at a*))
⟨*proof*⟩

lemma *differentiable-at-imp-differentiable-on*: ($\forall\, x{\in}$(*s*::(*real*^'*n*) *set*). *f differentiable at x*) $\Longrightarrow$ *f differentiable-on s*
⟨*proof*⟩

lemma *differentiable-on-eq-differentiable-at*: *open s* $\Longrightarrow$ (*f differentiable-on s* $\longleftrightarrow$ ($\forall\, x{\in}s$. *f differentiable at x*))
⟨*proof*⟩

lemma *differentiable-transform-within*:
assumes $0 < d$ $x \in s$ $\forall\, x'{\in}s$. *dist* x' $x < d \longrightarrow f\, x' = g\, x'$ *f differentiable* (*at x within s*)
shows *g differentiable* (*at x within s*)
⟨*proof*⟩

lemma *differentiable-transform-at*:
assumes $0 < d$ $\forall\, x'$. *dist* x' $x < d \longrightarrow f\, x' = g\, x'$ *f differentiable at x*
shows *g differentiable at x*
⟨*proof*⟩

22.9 Frechet derivative and Jacobian matrix.

definition *frechet-derivative f net* = (*SOME* f'. (*f has-derivative* f') *net*)

lemma *frechet-derivative-works*:
f differentiable net $\longleftrightarrow$ (*f has-derivative* (*frechet-derivative f net*)) *net*
⟨*proof*⟩

lemma *linear-frechet-derivative*: **fixes** *f*::*real*^'*a* $\Rightarrow$ *real*^'*b*
shows *f differentiable net* $\Longrightarrow$ *linear*(*frechet-derivative f net*)
⟨*proof*⟩

definition *jacobian f net* = *matrix*(*frechet-derivative f net*)

lemma *jacobian-works*: (*f*::(*real*^'*a*) $\Rightarrow$ (*real*^'*b*)) *differentiable net* $\longleftrightarrow$ (*f has-derivative* (λh. (*jacobian f net*) *v *h*)) *net*
⟨*proof*⟩

22.10 Differentiability implies continuity.

lemma *Lim-mul-norm-within*: **fixes** $f::'a::real\text{-}normed\text{-}vector \Rightarrow 'b::real\text{-}normed\text{-}vector$
shows $(f \text{ ---> } 0)$ $(at\ a\ within\ s) \Longrightarrow ((\lambda x.\ norm(x - a) *_R f(x)) \text{ ---> } 0)$ $(at\ a\ within\ s)$
⟨*proof*⟩

lemma *differentiable-imp-continuous-within*: **assumes** f *differentiable* $(at\ x\ within\ s)$
shows *continuous* $(at\ x\ within\ s)$ f ⟨*proof*⟩

lemma *differentiable-imp-continuous-at*: f *differentiable at* $x \Longrightarrow$ *continuous* $(at\ x)$ f
⟨*proof*⟩

lemma *differentiable-imp-continuous-on*: f *differentiable-on* $s \Longrightarrow$ *continuous-on* s f
⟨*proof*⟩

lemma *has-derivative-within-subset*:
$(f$ *has-derivative* $f')$ $(at\ x\ within\ s) \Longrightarrow t \subseteq s \Longrightarrow (f$ *has-derivative* $f')$ $(at\ x\ within\ t)$
⟨*proof*⟩

lemma *differentiable-within-subset*:
f *differentiable* $(at\ x\ within\ t) \Longrightarrow s \subseteq t \Longrightarrow f$ *differentiable* $(at\ x\ within\ s)$
⟨*proof*⟩

lemma *differentiable-on-subset*: f *differentiable-on* $t \Longrightarrow s \subseteq t \Longrightarrow f$ *differentiable-on* s
⟨*proof*⟩

lemma *differentiable-on-empty*: f *differentiable-on* $\{\}$
⟨*proof*⟩

22.11 Several results are easier using a "multiplied-out" variant. *) (* (I got this idea from Dieudonne's proof of the chain rule).

lemma *has-derivative-within-alt*:
$(f$ *has-derivative* $f')$ $(at\ x\ within\ s) \longleftrightarrow$ *bounded-linear* $f' \wedge$
$(\forall e>0.\ \exists d>0.\ \forall y\in s.\ norm(y - x) < d \longrightarrow norm(f(y) - f(x) - f'(y - x)) \leq e * norm(y - x))$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

lemma *has-derivative-at-alt*:
$(f$ *has-derivative* $f')$ $(at\ x) \longleftrightarrow$ *bounded-linear* $f' \wedge$
$(\forall e>0.\ \exists d>0.\ \forall y.\ norm(y - x) < d \longrightarrow norm(f\ y - f\ x - f'(y - x)) \leq e * norm(y - x))$
⟨*proof*⟩

22.12 The chain rule.

lemma *diff-chain-within*:
assumes $(f$ *has-derivative* $f')$ $(at\ x\ within\ s)$ $(g$ *has-derivative* $g')$ $(at\ (f\ x)\ within\ (f\ `\ s))$
shows $((g\ o\ f)$ *has-derivative* $(g'\ o\ f'))(at\ x\ within\ s)$
⟨*proof*⟩

lemma *diff-chain-at*:
$(f$ *has-derivative* $f')$ $(at\ x) \Longrightarrow (g$ *has-derivative* $g')$ $(at\ (f\ x)) \Longrightarrow ((g\ o\ f)$ *has-derivative* $(g'\ o\ f'))$ $(at\ x)$
⟨*proof*⟩

22.13 Composition rules stated just for differentiability.

lemma *differentiable-const*[*intro*]: $(\lambda z.\ c)$ *differentiable* $(net::'a::real\text{-}normed\text{-}vector\ net)$
⟨*proof*⟩

lemma *differentiable-id*[*intro*]: $(\lambda z.\ z)$ *differentiable* $(net::'a::real\text{-}normed\text{-}vector\ net)$
⟨*proof*⟩

lemma *differentiable-cmul*[*intro*]: f *differentiable net* $\Longrightarrow (\lambda x.\ c *_R f(x))$ *differentiable* $(net::'a::real\text{-}normed\text{-}vector\ net)$
⟨*proof*⟩

lemma *differentiable-neg*[*intro*]: f *differentiable net* $\Longrightarrow (\lambda z.\ -(f\ z))$ *differentiable* $(net::'a::real\text{-}normed\text{-}vector\ net)$
⟨*proof*⟩

lemma *differentiable-add*: f *differentiable net* $\Longrightarrow g$ *differentiable net*
$\Longrightarrow (\lambda z.\ f\ z + g\ z)$ *differentiable* $(net::'a::real\text{-}normed\text{-}vector\ net)$
⟨*proof*⟩

lemma *differentiable-sub*: f *differentiable net* $\Longrightarrow g$ *differentiable net*
$\Longrightarrow (\lambda z.\ f\ z - g\ z)$ *differentiable* $(net::'a::real\text{-}normed\text{-}vector\ net)$
⟨*proof*⟩

lemma *differentiable-setsum*: **fixes** $f::'a \Rightarrow (real\hat{\ }'n \Rightarrow real\hat{\ }'n)$
assumes *finite* s $\forall\, a{\in}s.\ (f\ a)$ *differentiable net*
shows $(\lambda x.\ setsum\ (\lambda a.\ f\ a\ x)\ s)$ *differentiable net* ⟨*proof*⟩

lemma *differentiable-setsum-numseg*: **fixes** $f::\text{-} \Rightarrow (real\hat{\ }'n \Rightarrow real\hat{\ }'n)$
shows $\forall\, i.\ m \le i \wedge i \le n \longrightarrow (f\ i)$ *differentiable net* $\Longrightarrow (\lambda x.\ setsum\ (\lambda a.\ f\ a\ x)\ \{m::nat..n\})$ *differentiable net*
⟨*proof*⟩

lemma *differentiable-chain-at*:
f *differentiable* $(at\ x) \Longrightarrow g$ *differentiable* $(at(f\ x)) \Longrightarrow (g\ o\ f)$ *differentiable* $(at$

x)
⟨proof⟩

lemma *differentiable-chain-within*:
f *differentiable* (*at* x *within* s) $\Longrightarrow$ g *differentiable* (*at*(f x) *within* (f ‘ s))
$\Longrightarrow$ (g *o* f) *differentiable* (*at* x *within* s)
⟨proof⟩

22.14 Uniqueness of derivative. *) (* *) (* The general result is a bit messy because we need approachability of the *) (* limit point from any direction. But OK for nontrivial intervals etc.

lemma *frechet-derivative-unique-within*: **fixes** f::*real*^′*a* ⇒ *real*^′*b*
assumes (f *has-derivative* f') (*at* x *within* s) (f *has-derivative* f'') (*at* x *within* s)
($\forall$ *i*::′*a*::*finite*. $\forall$ *e*>*0*. $\exists$ *d*. *0* < *abs*(*d*) $\wedge$ *abs*(*d*) < *e* $\wedge$ (x + d $*_R$ *basis* i) $\in$ s)
shows $f' = f''$ ⟨proof⟩

lemma *frechet-derivative-unique-at*: **fixes** f::*real*^′*a* ⇒ *real*^′*b*
shows (f *has-derivative* f') (*at* x) $\Longrightarrow$ (f *has-derivative* f'') (*at* x) $\Longrightarrow$ $f' = f''$
⟨proof⟩

lemma *isCont* f x = *continuous* (*at* x) f ⟨proof⟩

lemma *frechet-derivative-unique-within-closed-interval*: **fixes** f::*real*^′*a* ⇒ *real*^′*b*
assumes $\forall$ *i*. *a*\$*i* < *b*\$*i* x $\in$ {*a*..*b*} (**is** x∈*?I*) **and**
(f *has-derivative* f') (*at* x *within* {*a*..*b*}) **and**
(f *has-derivative* f'') (*at* x *within* {*a*..*b*})
shows $f' = f''$ ⟨proof⟩

lemma *frechet-derivative-unique-within-open-interval*: **fixes** f::*real*^′*a* ⇒ *real*^′*b*
assumes x $\in$ {*a*<..<*b*} (f *has-derivative* f') (*at* x *within* {*a*<..<*b*})
(f *has-derivative* f'') (*at* x *within* {*a*<..<*b*})
shows $f' = f''$ ⟨proof⟩

lemma *frechet-derivative-at*: **fixes** f::*real*^′*a* ⇒ *real*^′*b*
shows (f *has-derivative* f') (*at* x) $\Longrightarrow$ (f' = *frechet-derivative* f (*at* x))
⟨proof⟩

lemma *frechet-derivative-within-closed-interval*: **fixes** f::*real*^′*a* ⇒ *real*^′*b*
assumes $\forall$ *i*. *a*\$*i* < *b*\$*i* x $\in$ {*a*..*b*} (f *has-derivative* f') (*at* x *within* {*a*.. *b*})
shows *frechet-derivative* f (*at* x *within* {*a*.. *b*}) = f'
⟨proof⟩

22.15 Component of the differential must be zero if it exists at a local *) (* maximum or minimum for that corresponding component.

lemma *differential-zero-maxmin-component*: **fixes** f::*real*^′*a* ⇒ *real*^′*b*

assumes $0 < e$ $((\forall\, y \in ball\ x\ e.\ (f\ y)\$k \leq (f\ x)\$k) \vee (\forall\, y{\in}ball\ x\ e.\ (f\ x)\$k \leq (f\ y)\$k))$
f *differentiable* $(at\ x)$ **shows** *jacobian* f $(at\ x)$ \$ $k = 0$ ⟨*proof*⟩

22.16 In particular if we have a mapping into $(real,\ 1)$ *cart*.

lemma *differential-zero-maxmin*: **fixes** $f{::}real\hat{\ }'a \Rightarrow real$
assumes $x \in s$ *open* s $(f$ *has-derivative* $f')$ $(at\ x)$
$(\forall\, y{\in}s.\ f\ y \leq f\ x) \vee (\forall\, y{\in}s.\ f\ x \leq f\ y)$
shows $f' = (\lambda v.\ 0)$ ⟨*proof*⟩

22.17 The traditional Rolle theorem in one dimension.

lemma *rolle*: **fixes** $f{::}real{\Rightarrow}real$
assumes $a < b$ $f\ a = f\ b$ *continuous-on* $\{a..b\}$ f
$\forall\, x{\in}\{a{<}..{<}b\}.\ (f$ *has-derivative* $f'(x))$ $(at\ x)$
shows $\exists\, x{\in}\{a{<}..{<}b\}.\ f'\ x = (\lambda v.\ 0)$ ⟨*proof*⟩

22.18 One-dimensional mean value theorem.

lemma *mvt*: **fixes** $f{::}real \Rightarrow real$
assumes $a < b$ *continuous-on* $\{a\ ..\ b\}$ f $\forall\, x{\in}\{a{<}..{<}b\}.\ (f$ *has-derivative* $(f'\ x))$ $(at\ x)$
shows $\exists\, x{\in}\{a{<}..{<}b\}.\ (f\ b - f\ a = (f'\ x)\ (b - a))$ ⟨*proof*⟩

lemma *mvt-simple*: **fixes** $f{::}real \Rightarrow real$
assumes $a{<}b$ $\forall\, x{\in}\{a..b\}.\ (f$ *has-derivative* $f'\ x)$ $(at\ x\ within\ \{a..b\})$
shows $\exists\, x{\in}\{a{<}..{<}b\}.\ f\ b - f\ a = f'\ x\ (b - a)$
⟨*proof*⟩

lemma *mvt-very-simple*: **fixes** $f{::}real \Rightarrow real$
assumes $a \leq b$ $\forall\, x{\in}\{a..b\}.\ (f$ *has-derivative* $f'(x))$ $(at\ x\ within\ \{a..b\})$
shows $\exists\, x{\in}\{a..b\}.\ f\ b - f\ a = f'\ x\ (b - a)$ ⟨*proof*⟩

22.19 A nice generalization (see Havin's proof of 5.19 from Rudin's book).

lemma *mvt-general*: **fixes** $f{::}real{\Rightarrow}real\hat{\ }'n$
assumes $a{<}b$ *continuous-on* $\{a..b\}$ f $\forall\, x{\in}\{a{<}..{<}b\}.\ (f$ *has-derivative* $f'(x))$ $(at\ x)$
shows $\exists\, x{\in}\{a{<}..{<}b\}.\ norm(f\ b - f\ a) \leq norm(f'(x)\ (b - a))$ ⟨*proof*⟩

22.20 Still more general bound theorem.

lemma *differentiable-bound*: **fixes** $f{::}real\hat{\ }'a \Rightarrow real\hat{\ }'b$
assumes *convex* s $\forall\, x{\in}s.\ (f$ *has-derivative* $f'(x))$ $(at\ x\ within\ s)$ $\forall\, x{\in}s.\ onorm(f'\ x) \leq B$ **and** $x{:}x{\in}s$ **and** $y{:}y{\in}s$
shows $norm(f\ x - f\ y) \leq B * norm(x - y)$ ⟨*proof*⟩

lemma *onorm-vec1*: **fixes** $f{::}real \Rightarrow real$

shows *onorm* $(\lambda x.\ vec1\ (f\ (dest\text{-}vec1\ x))) = onorm\ f$ ⟨*proof*⟩

lemma *convex-vec1*:*convex* (*vec1* ' *s*) = *convex* (*s*::*real set*)
⟨*proof*⟩

lemma *differentiable-bound-real*: **fixes** *f*::*real* ⇒ *real*
assumes *convex s* ∀ *x*∈*s*. (*f has-derivative f′ x*) (*at x within s*) ∀ *x*∈*s*. *onorm*(*f′ x*) ≤ *B* **and** *x*:*x*∈*s* **and** *y*:*y*∈*s*
shows *norm*(*f x* − *f y*) ≤ *B* ∗ *norm*(*x* − *y*)
⟨*proof*⟩

22.21 In particular.

lemma *has-derivative-zero-constant*: **fixes** *f*::*real*⇒*real*
assumes *convex s* ∀ *x*∈*s*. (*f has-derivative* (λ*h*. *0*)) (*at x within s*)
shows ∃ *c*. ∀ *x*∈*s*. *f x* = *c* ⟨*proof*⟩

lemma *has-derivative-zero-unique*: **fixes** *f*::*real*⇒*real*
assumes *convex s a* ∈ *s f a* = *c* ∀ *x*∈*s*. (*f has-derivative* (λ*h*. *0*)) (*at x within s*) *x*∈*s*
shows *f x* = *c* ⟨*proof*⟩

22.22 Differentiability of inverse function (most basic form).

lemma *has-derivative-inverse-basic*: **fixes** *f*::*real*^′*b* ⇒ *real*^′*c*
assumes (*f has-derivative f′*) (*at* (*g y*)) *bounded-linear g′ g′* ∘ *f′* = *id continuous* (*at y*) *g*
open t y ∈ *t* ∀ *z*∈*t*. *f*(*g z*) = *z*
shows (*g has-derivative g′*) (*at y*) ⟨*proof*⟩

22.23 Simply rewrite that based on the domain point x.

lemma *has-derivative-inverse-basic-x*: **fixes** *f*::*real*^′*b* ⇒ *real*^′*c*
assumes (*f has-derivative f′*) (*at x*) *bounded-linear g′ g′ o f′* = *id*
continuous (*at* (*f x*)) *g g*(*f x*) = *x open t f x* ∈ *t* ∀ *y*∈*t*. *f*(*g y*) = *y*
shows (*g has-derivative g′*) (*at* (*f*(*x*)))
⟨*proof*⟩

22.24 This is the version in Dieudonne', assuming continuity of f and g.

lemma *has-derivative-inverse-dieudonne*: **fixes** *f*::*real*^′*a* ⇒ *real*^′*b*
assumes *open s open* (*f* ' *s*) *continuous-on s f continuous-on* (*f* ' *s*) *g* ∀ *x*∈*s*. *g*(*f x*) = *x*
x∈*s* (*f has-derivative f′*) (*at x*) *bounded-linear g′ g′ o f′* = *id*
shows (*g has-derivative g′*) (*at* (*f x*))
⟨*proof*⟩

22.25 Here's the simplest way of not assuming much about g.

lemma *has-derivative-inverse*: **fixes** $f::real\hat{\ }'a \Rightarrow real\hat{\ }'b$
assumes $compact\ s\ x \in s\ f\ x \in interior(f\ `\ s)\ continuous\text{-}on\ s\ f$
$\forall y \in s.\ g(f\ y) = y\ (f\ has\text{-}derivative\ f')\ (at\ x)\ bounded\text{-}linear\ g'\ g' \circ f' = id$
shows $(g\ has\text{-}derivative\ g')\ (at\ (f\ x))$ $\langle proof \rangle$

22.26 Proving surjectivity via Brouwer fixpoint theorem.

lemma *brouwer-surjective*: **fixes** $f::real\hat{\ }'n \Rightarrow real\hat{\ }'n$
assumes $compact\ t\ convex\ t\ \ t \neq \{\}\ continuous\text{-}on\ t\ f$
$\forall x \in s.\ \forall y \in t.\ x + (y - f\ y) \in t\ x \in s$
shows $\exists y \in t.\ f\ y = x$ $\langle proof \rangle$

lemma *brouwer-surjective-cball*: **fixes** $f::real\hat{\ }'n \Rightarrow real\hat{\ }'n$
assumes $0 < e\ continuous\text{-}on\ (cball\ a\ e)\ f$
$\forall x \in s.\ \forall y \in cball\ a\ e.\ x + (y - f\ y) \in cball\ a\ e\ x \in s$
shows $\exists y \in cball\ a\ e.\ f\ y = x$ $\langle proof \rangle$

See Sussmann: "Multidifferential calculus", Theorem 2.1.1

lemma *sussmann-open-mapping*: **fixes** $f::real\hat{\ }'a \Rightarrow real\hat{\ }'b$
assumes $open\ s\ continuous\text{-}on\ s\ f\ x \in s$
$(f\ has\text{-}derivative\ f')\ (at\ x)\ bounded\text{-}linear\ g'\ f' \circ g' = id$
$t \subseteq s\ x \in interior\ t$
shows $f\ x \in interior\ (f\ `\ t)$ $\langle proof \rangle$

Hence the following eccentric variant of the inverse function theorem. *) (* This has no continuity assumptions, but we do need the inverse function. *) (* We could put f' o g = I but this happens to fit with the minimal linear *) (* algebra theory I've set up so far.

lemma *has-derivative-inverse-strong*: **fixes** $f::real\hat{\ }'n \Rightarrow real\hat{\ }'n$
assumes $open\ s\ x \in s\ continuous\text{-}on\ s\ f$
$\forall x \in s.\ g(f\ x) = x\ (f\ has\text{-}derivative\ f')\ (at\ x)\ f'\ o\ g' = id$
shows $(g\ has\text{-}derivative\ g')\ (at\ (f\ x))$ $\langle proof \rangle$

22.27 A rewrite based on the other domain.

lemma *has-derivative-inverse-strong-x*: **fixes** $f::real\hat{\ }'n \Rightarrow real\hat{\ }'n$
assumes $open\ s\ g\ y \in s\ continuous\text{-}on\ s\ f$
$\forall x \in s.\ g(f\ x) = x\ (f\ has\text{-}derivative\ f')\ (at\ (g\ y))\ f'\ o\ g' = id\ f(g\ y) = y$
shows $(g\ has\text{-}derivative\ g')\ (at\ y)$
$\langle proof \rangle$

22.28 On a region.

lemma *has-derivative-inverse-on*: **fixes** $f::real\hat{\ }'n \Rightarrow real\hat{\ }'n$
assumes $open\ s\ \forall x \in s.\ (f\ has\text{-}derivative\ f'(x))\ (at\ x)\ \forall x \in s.\ g(f\ x) = x\ f'(x)\ o\ g'(x) = id\ x \in s$

shows $(g\ \textit{has-derivative}\ g'(x))\ (\textit{at}\ (f\ x))$
$\langle \textit{proof} \rangle$

22.29 Invertible derivative continous at a point implies local injectivity. *) (* It's only for this we need continuity of the derivative, except of course *) (* if we want the fact that the inverse derivative is also continuous. So if *) (* we know for some other reason that the inverse function exists, it's OK.

lemma *bounded-linear-sub*: $\textit{bounded-linear}\ f \Longrightarrow \textit{bounded-linear}\ g ==> \textit{bounded-linear}$ $(\lambda x.\ f\ x - g\ x)$
$\langle \textit{proof} \rangle$

lemma *has-derivative-locally-injective*: **fixes** $f{::}\textit{real}\hat{~}'n \Rightarrow \textit{real}\hat{~}'m$
assumes $a \in s$ $\textit{open}\ s$ $\textit{bounded-linear}\ g'$ $g'\ o\ f'(a) = \textit{id}$
$\forall x{\in}s.\ (f\ \textit{has-derivative}\ f'(x))\ (\textit{at}\ x)$
$\forall e{>}0.\ \exists d{>}0.\ \forall x.\ \textit{dist}\ a\ x < d \longrightarrow \textit{onorm}(\lambda v.\ f'\ x\ v - f'\ a\ v) < e$
obtains t **where** $a \in t$ $\textit{open}\ t$ $\forall x{\in}t.\ \forall x'{\in}t.\ (f\ x' = f\ x) \longrightarrow (x' = x)$ $\langle \textit{proof} \rangle$

22.30 Uniformly convergent sequence of derivatives.

lemma *has-derivative-sequence-lipschitz-lemma*: **fixes** $f{::}\textit{nat} \Rightarrow \textit{real}\hat{~}'m \Rightarrow \textit{real}\hat{~}'n$
assumes $\textit{convex}\ s$ $\forall n.\ \forall x{\in}s.\ ((f\ n)\ \textit{has-derivative}\ (f'\ n\ x))\ (\textit{at}\ x\ \textit{within}\ s)$
$\forall n{\geq}N.\ \forall x{\in}s.\ \forall h.\ \textit{norm}(f'\ n\ x\ h - g'\ x\ h) \leq e * \textit{norm}(h)$
shows $\forall m{\geq}N.\ \forall n{\geq}N.\ \forall x{\in}s.\ \forall y{\in}s.\ \textit{norm}((f\ m\ x - f\ n\ x) - (f\ m\ y - f\ n\ y)) \leq 2 * e * \textit{norm}(x - y)$ $\langle \textit{proof} \rangle$

lemma *has-derivative-sequence-lipschitz*: **fixes** $f{::}\textit{nat} \Rightarrow \textit{real}\hat{~}'m \Rightarrow \textit{real}\hat{~}'n$
assumes $\textit{convex}\ s$ $\forall n.\ \forall x{\in}s.\ ((f\ n)\ \textit{has-derivative}\ (f'\ n\ x))\ (\textit{at}\ x\ \textit{within}\ s)$
$\forall e{>}0.\ \exists N.\ \forall n{\geq}N.\ \forall x{\in}s.\ \forall h.\ \textit{norm}(f'\ n\ x\ h - g'\ x\ h) \leq e * \textit{norm}(h)$ $0 < e$
shows $\forall e{>}0.\ \exists N.\ \forall m{\geq}N.\ \forall n{\geq}N.\ \forall x{\in}s.\ \forall y{\in}s.\ \textit{norm}((f\ m\ x - f\ n\ x) - (f\ m\ y - f\ n\ y)) \leq e * \textit{norm}(x - y)$ $\langle \textit{proof} \rangle$

lemma *has-derivative-sequence*: **fixes** $f{::}\textit{nat}{\Rightarrow}\textit{real}\hat{~}'m{\Rightarrow}\textit{real}\hat{~}'n$
assumes $\textit{convex}\ s$ $\forall n.\ \forall x{\in}s.\ ((f\ n)\ \textit{has-derivative}\ (f'\ n\ x))\ (\textit{at}\ x\ \textit{within}\ s)$
$\forall e{>}0.\ \exists N.\ \forall n{\geq}N.\ \forall x{\in}s.\ \forall h.\ \textit{norm}(f'\ n\ x\ h - g'\ x\ h) \leq e * \textit{norm}(h)$
$x0 \in s$ $((\lambda n.\ f\ n\ x0) ---> l)\ \textit{sequentially}$
shows $\exists g.\ \forall x{\in}s.\ ((\lambda n.\ f\ n\ x) ---> g\ x)\ \textit{sequentially} \wedge (g\ \textit{has-derivative}\ g'(x))\ (\textit{at}\ x\ \textit{within}\ s)$ $\langle \textit{proof} \rangle$

22.31 Can choose to line up antiderivatives if we want.

lemma *has-antiderivative-sequence*: **fixes** $f{::}\textit{nat}{\Rightarrow}\ \textit{real}\hat{~}'m \Rightarrow \textit{real}\hat{~}'n$
assumes $\textit{convex}\ s$ $\forall n.\ \forall x{\in}s.\ ((f\ n)\ \textit{has-derivative}\ (f'\ n\ x))\ (\textit{at}\ x\ \textit{within}\ s)$
$\forall e{>}0.\ \exists N.\ \forall n{\geq}N.\ \forall x{\in}s.\ \forall h.\ \textit{norm}(f'\ n\ x\ h - g'\ x\ h) \leq e * \textit{norm}\ h$
shows $\exists g.\ \forall x{\in}s.\ (g\ \textit{has-derivative}\ g'(x))\ (\textit{at}\ x\ \textit{within}\ s)$ $\langle \textit{proof} \rangle$

lemma *has-antiderivative-limit*: **fixes** $g'{::}\textit{real}\hat{~}'m \Rightarrow \textit{real}\hat{~}'m \Rightarrow \textit{real}\hat{~}'n$

assumes *convex s* $\forall e>0.\ \exists f\, f'.\ \forall x\in s.\ (f$ *has-derivative* $(f'\ x))$ $(at\ x\ within\ s)$ $\wedge$ $(\forall h.\ norm(f'\ x\ h\ -\ g'\ x\ h) \le e * norm(h))$
shows $\exists g.\ \forall x\in s.\ (g$ *has-derivative* $g'(x))$ $(at\ x\ within\ s)$ $\langle proof\rangle$

22.32 Differentiation of a series.

definition *sums-seq* :: $(nat \Rightarrow 'a{::}real\text{-}normed\text{-}vector) \Rightarrow 'a \Rightarrow (nat\ set) \Rightarrow bool$ (**infixl** *sums'-seq 12*) **where** $(f$ *sums-seq* $l)\ s \equiv ((\lambda n.\ setsum\ f\ (s \cap \{0..n\}))\ {-}{-}{-}{>}\ l)$ *sequentially*

lemma *has-derivative-series*: **fixes** $f{::}nat \Rightarrow real\hat{\ }'m \Rightarrow real\hat{\ }'n$
assumes *convex s* $\forall n.\ \forall x\in s.\ ((f\ n)$ *has-derivative* $(f'\ n\ x))$ $(at\ x\ within\ s)$
$\forall e>0.\ \exists N.\ \forall n\ge N.\ \forall x\in s.\ \forall h.\ norm(setsum\ (\lambda i.\ f'\ i\ x\ h)\ (k \cap \{0..n\}) - g'\ x\ h) \le e * norm(h)$
$x\in s$ $((\lambda n.\ f\ n\ x)$ *sums-seq* $l)\ k$
shows $\exists g.\ \forall x\in s.\ ((\lambda n.\ f\ n\ x)$ *sums-seq* $(g\ x))\ k \wedge (g$ *has-derivative* $g'(x))$ $(at\ x\ within\ s)$
$\langle proof\rangle$

22.33 Derivative with composed bilinear function.

lemma *has-derivative-bilinear-within*: **fixes** $h{::}real\hat{\ }'m \Rightarrow real\hat{\ }'n \Rightarrow real\hat{\ }'p$ **and** $f{::}real\hat{\ }'q \Rightarrow real\hat{\ }'m$
assumes $(f$ *has-derivative* $f')$ $(at\ x\ within\ s)$ $(g$ *has-derivative* $g')$ $(at\ x\ within\ s)$ *bounded-bilinear h*
shows $((\lambda x.\ h\ (f\ x)\ (g\ x))$ *has-derivative* $(\lambda d.\ h\ (f\ x)\ (g'\ d) + h\ (f'\ d)\ (g\ x)))$ $(at\ x\ within\ s)$ $\langle proof\rangle$

lemma *has-derivative-bilinear-at*: **fixes** $h{::}real\hat{\ }'m \Rightarrow real\hat{\ }'n \Rightarrow real\hat{\ }'p$ **and** $f{::}real\hat{\ }'p \Rightarrow real\hat{\ }'m$
assumes $(f$ *has-derivative* $f')$ $(at\ x)$ $(g$ *has-derivative* $g')$ $(at\ x)$ *bounded-bilinear h*
shows $((\lambda x.\ h\ (f\ x)\ (g\ x))$ *has-derivative* $(\lambda d.\ h\ (f\ x)\ (g'\ d) + h\ (f'\ d)\ (g\ x)))$ $(at\ x)$
$\langle proof\rangle$

22.34 Considering derivative $(real,\ 1)\ cart \Rightarrow (real,\ 'n)\ cart$ as a vector.

definition *has-vector-derivative* :: $(real \Rightarrow 'b{::}real\text{-}normed\text{-}vector) \Rightarrow ('b) \Rightarrow (real\ net \Rightarrow bool)$
(**infixl** *has'-vector'-derivative 12*) **where**
$(f$ *has-vector-derivative* $f')\ net \equiv (f$ *has-derivative* $(\lambda x.\ x *_R f'))\ net$

definition *vector-derivative f net* $\equiv (SOME\ f'.\ (f$ *has-vector-derivative* $f')\ net)$

lemma *vector-derivative-works*: **fixes** $f{::}real \Rightarrow 'a{::}real\text{-}normed\text{-}vector$
shows f *differentiable net* $\longleftrightarrow (f$ *has-vector-derivative* $(vector\text{-}derivative\ f\ net))$ *net* (**is** $?l = ?r$)
$\langle proof\rangle$

lemma *vector-derivative-unique-at*: **fixes** f::$real \Rightarrow real\hat{\ }'n$
assumes (f *has-vector-derivative* f') (*at* x) (f *has-vector-derivative* f'') (*at* x)
shows $f' = f''$ ⟨*proof*⟩

lemma *vector-derivative-unique-within-closed-interval*: **fixes** f::$real \Rightarrow real\hat{\ }'n$
assumes $a < b$ $x \in \{a..b\}$
(f *has-vector-derivative* f') (*at* x *within* $\{a..b\}$)
(f *has-vector-derivative* f'') (*at* x *within* $\{a..b\}$) **shows** $f' = f''$ ⟨*proof*⟩

lemma *vector-derivative-at*: **fixes** f::$real \Rightarrow real\hat{\ }'a$ **shows**
(f *has-vector-derivative* f') (*at* x) $\Longrightarrow$ *vector-derivative* f (*at* x) $= f'$
⟨*proof*⟩

lemma *vector-derivative-within-closed-interval*: **fixes** f::$real \Rightarrow real\hat{\ }'a$
assumes $a < b$ $x \in \{a..b\}$ (f *has-vector-derivative* f') (*at* x *within* $\{a..b\}$)
shows *vector-derivative* f (*at* x *within* $\{a..b\}$) $= f'$
⟨*proof*⟩

lemma *has-vector-derivative-within-subset*:
(f *has-vector-derivative* f') (*at* x *within* s) $\Longrightarrow t \subseteq s \Longrightarrow$ (f *has-vector-derivative* f') (*at* x *within* t)
⟨*proof*⟩

lemma *has-vector-derivative-const*:
(($\lambda x.\ c$) *has-vector-derivative* *0*) *net*
⟨*proof*⟩

lemma *has-vector-derivative-id*: ((λx::*real*. x) *has-vector-derivative* *1*) *net*
⟨*proof*⟩

lemma *has-vector-derivative-cmul*: (f *has-vector-derivative* f') *net* $\Longrightarrow$ (($\lambda x.\ c *_R f\,x$) *has-vector-derivative* ($c *_R f'$)) *net*
⟨*proof*⟩

lemma *has-vector-derivative-cmul-eq*: **assumes** $c \neq 0$
shows ((($\lambda x.\ c *_R f\,x$) *has-vector-derivative* ($c *_R f'$)) *net* $\longleftrightarrow$ (f *has-vector-derivative* f') *net*)
⟨*proof*⟩

lemma *has-vector-derivative-neg*:
(f *has-vector-derivative* f') *net* $\Longrightarrow$ (($\lambda x.\ -(f\,x)$) *has-vector-derivative* $(-\ f')$) *net*
⟨*proof*⟩

lemma *has-vector-derivative-add*:
assumes (f *has-vector-derivative* f') *net* (g *has-vector-derivative* g') *net*
shows (($\lambda x.\ f(x) + g(x)$) *has-vector-derivative* $(f' + g')$) *net*
⟨*proof*⟩

lemma *has-vector-derivative-sub*:
assumes (f *has-vector-derivative* f') *net* (g *has-vector-derivative* g') *net*
shows (($\lambda x.\ f(x) - g(x)$) *has-vector-derivative* ($f' - g'$)) *net*
⟨*proof*⟩

lemma *has-vector-derivative-bilinear-within*: **fixes** h::*real*ˆ′*m* ⇒ *real*ˆ′*n* ⇒ *real*ˆ′*p*
assumes (f *has-vector-derivative* f') (*at* x *within* s) (g *has-vector-derivative* g') (*at* x *within* s) *bounded-bilinear* h
shows (($\lambda x.\ h\ (f\ x)\ (g\ x)$) *has-vector-derivative* ($h\ (f\ x)\ g' + h\ f'\ (g\ x)$)) (*at* x *within* s) ⟨*proof*⟩

lemma *has-vector-derivative-bilinear-at*: **fixes** h::*real*ˆ′*m* ⇒ *real*ˆ′*n* ⇒ *real*ˆ′*p*
assumes (f *has-vector-derivative* f') (*at* x) (g *has-vector-derivative* g') (*at* x) *bounded-bilinear* h
shows (($\lambda x.\ h\ (f\ x)\ (g\ x)$) *has-vector-derivative* ($h\ (f\ x)\ g' + h\ f'\ (g\ x)$)) (*at* x)
⟨*proof*⟩

lemma *has-vector-derivative-at-within*: (f *has-vector-derivative* f') (*at* x) $\Longrightarrow$ (f *has-vector-derivative* f') (*at* x *within* s)
⟨*proof*⟩

lemma *has-vector-derivative-transform-within*:
assumes $0 < d$ $x \in s$ $\forall x' \in s.\ dist\ x'\ x < d \longrightarrow f\ x' = g\ x'$ (f *has-vector-derivative* f') (*at* x *within* s)
shows (g *has-vector-derivative* f') (*at* x *within* s)
⟨*proof*⟩

lemma *has-vector-derivative-transform-at*:
assumes $0 < d$ $\forall x'.\ dist\ x'\ x < d \longrightarrow f\ x' = g\ x'$ (f *has-vector-derivative* f') (*at* x)
shows (g *has-vector-derivative* f') (*at* x)
⟨*proof*⟩

lemma *has-vector-derivative-transform-within-open*:
assumes *open* s $x \in s$ $\forall y \in s.\ f\ y = g\ y$ (f *has-vector-derivative* f') (*at* x)
shows (g *has-vector-derivative* f') (*at* x)
⟨*proof*⟩

lemma *vector-diff-chain-at*:
assumes (f *has-vector-derivative* f') (*at* x) (g *has-vector-derivative* g') (*at* ($f\ x$))
shows (($g \circ f$) *has-vector-derivative* ($f' *_R g'$)) (*at* x)
⟨*proof*⟩

lemma *vector-diff-chain-within*:
assumes (f *has-vector-derivative* f') (*at* x *within* s) (g *has-vector-derivative* g') (*at* ($f\ x$) *within* f ‘ s)
shows ((g *o* f) *has-vector-derivative* ($f' *_R g'$)) (*at* x *within* s)
⟨*proof*⟩

end

23 Integration: Kurzweil-Henstock gauge integration in many dimensions.

theory *Integration*
 imports *Derivative ~~/src/HOL/Decision-Procs/Dense-Linear-Order*
begin

declare [[*smt-certificates=~~/src/HOL/Multivariate-Analysis/Integration.certs*]]
declare [[*smt-fixed=true*]]
declare [[*z3-proofs=true*]]

⟨*ML*⟩

23.1 Sundries

lemma *conjunctD2*: **assumes** $a \wedge b$ **shows** a b ⟨*proof*⟩
lemma *conjunctD3*: **assumes** $a \wedge b \wedge c$ **shows** a b c ⟨*proof*⟩
lemma *conjunctD4*: **assumes** $a \wedge b \wedge c \wedge d$ **shows** a b c d ⟨*proof*⟩
lemma *conjunctD5*: **assumes** $a \wedge b \wedge c \wedge d \wedge e$ **shows** a b c d e ⟨*proof*⟩

declare *smult-conv-scaleR*[*simp*]

lemma *simple-image*: $\{f\, x\ |x\ .\ x \in s\} = f\ `\ s$ ⟨*proof*⟩

lemma *linear-simps*: **assumes** *bounded-linear f*
 shows $f\ (a + b) = f\, a + f\, b$ $f\ (a - b) = f\, a - f\, b$ $f\ 0 = 0$ $f\ (-\ a) = -\ f\, a$ $f\ (s *_R v) = s *_R (f\, v)$
 ⟨*proof*⟩

lemma *bounded-linearI*:**assumes** $\bigwedge x\ y.\ f\ (x + y) = f\, x + f\, y$
 $\bigwedge r\ x.\ f\ (r *_R x) = r *_R f\, x$ $\bigwedge x.\ norm\ (f\, x) \le norm\ x * K$
 shows *bounded-linear f*
 ⟨*proof*⟩

lemma *real-le-inf-subset*:
 assumes $t \neq \{\}$ $t \subseteq s$ $\exists b.\ b <=* s$ **shows** $Inf\ s <= Inf\ (t::real\ set)$
 ⟨*proof*⟩

lemma *real-ge-sup-subset*:
 assumes $t \neq \{\}$ $t \subseteq s$ $\exists b.\ s *<= b$ **shows** $Sup\ s >= Sup\ (t::real\ set)$
 ⟨*proof*⟩

lemma *dist-trans*[*simp*]:$dist\ (vec1\ x)\ (vec1\ y) = dist\ x\ (y::real)$
 ⟨*proof*⟩

lemma *Lim-trans*[*simp*]: **fixes** $f::'a \Rightarrow real$
shows $((\lambda x.\ vec1\ (f\ x))\ {-}{-}{-}{>}\ vec1\ l)\ net \longleftrightarrow (f\ {-}{-}{-}{>}\ l)\ net$
⟨*proof*⟩

lemma *bounded-linear-component*[*intro*]: $bounded\text{-}linear\ (\lambda x::real\hat{~}'n.\ x\ \$\ k)$
⟨*proof*⟩

lemma *bounded-vec1*[*intro*]: $bounded\ s \Longrightarrow bounded\ (vec1\ \text{`}\ (s::real\ set))$
⟨*proof*⟩

lemma *transitive-stepwise-lt-eq*:
assumes $(\bigwedge x\ y\ z::nat.\ R\ x\ y \Longrightarrow R\ y\ z \Longrightarrow R\ x\ z)$
shows $((\forall m.\ \forall n>m.\ R\ m\ n) \longleftrightarrow (\forall n.\ R\ n\ (Suc\ n)))$ (**is** $?l = ?r$)
⟨*proof*⟩

lemma *transitive-stepwise-gt*:
assumes $\bigwedge x\ y\ z.\ R\ x\ y \Longrightarrow R\ y\ z \Longrightarrow R\ x\ z\ \bigwedge n.\ R\ n\ (Suc\ n)$
shows $\forall n>m.\ R\ m\ n$
⟨*proof*⟩

lemma *transitive-stepwise-le-eq*:
assumes $\bigwedge x.\ R\ x\ x\ \bigwedge x\ y\ z.\ R\ x\ y \Longrightarrow R\ y\ z \Longrightarrow R\ x\ z$
shows $(\forall m.\ \forall n\geq m.\ R\ m\ n) \longleftrightarrow (\forall n.\ R\ n\ (Suc\ n))$ (**is** $?l = ?r$)
⟨*proof*⟩

lemma *transitive-stepwise-le*:
assumes $\bigwedge x.\ R\ x\ x\ \bigwedge x\ y\ z.\ R\ x\ y \Longrightarrow R\ y\ z \Longrightarrow R\ x\ z\ \bigwedge n.\ R\ n\ (Suc\ n)$
shows $\forall n\geq m.\ R\ m\ n$
⟨*proof*⟩

23.2 Some useful lemmas about intervals.

lemma *empty-as-interval*: $\{\} = \{1..0::real\hat{~}'n\}$
⟨*proof*⟩

lemma *interior-subset-union-intervals*:
assumes $i = \{a..b::real\hat{~}'n\}\ j = \{c..d\}\ interior\ j \neq \{\}\ i \subseteq j \cup s\ interior(i) \cap interior(j) = \{\}$
shows $interior\ i \subseteq interior\ s$ ⟨*proof*⟩

lemma *inter-interior-unions-intervals*: **fixes** $f::(real\hat{~}'n)\ set\ set$
assumes $finite\ f\ open\ s\ \forall t\in f.\ \exists a\ b.\ t = \{a..b\}\ \forall t\in f.\ s \cap (interior\ t) = \{\}$
shows $s \cap interior(\bigcup f) = \{\}$ ⟨*proof*⟩

23.3 Bounds on intervals where they exist.

definition $interval\text{-}upperbound\ (s::(real\hat{~}'n)\ set) = (\chi\ i.\ Sup\ \{a.\ \exists x\in s.\ x\$i = a\})$

definition $interval\text{-}lowerbound\ (s::(real\hat{~}'n)\ set) = (\chi\ i.\ Inf\ \{a.\ \exists x\in s.\ x\$i = a\})$

lemma *interval-upperbound*[*simp*]: **assumes** $\forall\, i.\ a\$i \leq b\$i$ **shows** *interval-upperbound* $\{a..b\} = b$
⟨*proof*⟩

lemma *interval-lowerbound*[*simp*]: **assumes** $\forall\, i.\ a\$i \leq b\$i$ **shows** *interval-lowerbound* $\{a..b\} = a$
⟨*proof*⟩

lemmas *interval-bounds* = *interval-upperbound interval-lowerbound*

lemma *interval-bounds'*[*simp*]: **assumes** $\{a..b\}\neq\{\}$ **shows** *interval-upperbound* $\{a..b\} = b$ *interval-lowerbound* $\{a..b\} = a$
⟨*proof*⟩

lemma *interval-upperbound-1*[*simp*]: *dest-vec1* $a \leq$ *dest-vec1* $b \Longrightarrow$ *interval-upperbound* $\{a..b\} = (b::real\hat{}1)$
⟨*proof*⟩

lemma *interval-lowerbound-1*[*simp*]: *dest-vec1* $a \leq$ *dest-vec1* $b \Longrightarrow$ *interval-lowerbound* $\{a..b\} = (a::real\hat{}1)$
⟨*proof*⟩

lemmas *interval-bound-1* = *interval-upperbound-1 interval-lowerbound-1*

23.4 Content (length, area, volume...) of an interval.

definition *content* $(s::(real\hat{}'n)\ set) =$
(*if* $s = \{\}$ *then* 0 *else* $(\prod i{\in}UNIV.\ (interval\text{-}upperbound\ s)\$i - (interval\text{-}lowerbound\ s)\$i))$

lemma *interval-not-empty*: $\forall\, i.\ a\$i \leq b\$i \Longrightarrow \{a..b::real\hat{}'n\} \neq \{\}$
⟨*proof*⟩

lemma *content-closed-interval*: **assumes** $\forall\, i.\ a\$i \leq b\$i$
shows *content* $\{a..b\} = (\prod i{\in}UNIV.\ b\$i - a\$i)$
⟨*proof*⟩

lemma *content-closed-interval'*: **assumes** $\{a..b\}\neq\{\}$ **shows** *content* $\{a..b\} = (\prod i{\in}UNIV.\ b\$i - a\$i)$
⟨*proof*⟩

lemma *content-1*: *dest-vec1* $a \leq$ *dest-vec1* $b \Longrightarrow$ *content* $\{a..b\} =$ *dest-vec1* $b -$ *dest-vec1* a
⟨*proof*⟩

lemma *content-1'*: $a \leq b \Longrightarrow$ *content* $\{vec1\ a..vec1\ b\} = b - a$ ⟨*proof*⟩

lemma *content-unit*[*intro*]: *content*$\{0..1::real\hat{}'n\} = 1$ ⟨*proof*⟩

lemma *content-pos-le*[*intro*]: $0 \leq content\ \{a..b\}$ ⟨*proof*⟩

lemma *content-pos-lt*: **assumes** $\forall\, i.\ a\$i < b\$i$ **shows** $0 < content\ \{a..b\}$ ⟨*proof*⟩

lemma *content-pos-lt-1*: $dest\text{-}vec1\ a < dest\text{-}vec1\ b \Longrightarrow 0 < content(\{a..b\})$ ⟨*proof*⟩

lemma *content-eq-0*: $content(\{a..b{::}real\hat{\ }'n\}) = 0 \longleftrightarrow (\exists\, i.\ b\$i \leq a\$i)$ ⟨*proof*⟩

lemma *cond-cases*:$(P \Longrightarrow Q\ x) \Longrightarrow (\neg\ P \Longrightarrow Q\ y) \Longrightarrow Q\ (if\ P\ then\ x\ else\ y)$ ⟨*proof*⟩

lemma *content-closed-interval-cases*:
$content\ \{a..b\} = (if\ \forall\, i.\ a\$i \leq b\$i\ then\ setprod\ (\lambda i.\ b\$i - a\$i)\ UNIV\ else\ 0)$
⟨*proof*⟩

lemma *content-eq-0-interior*: $content\ \{a..b\} = 0 \longleftrightarrow interior(\{a..b\}) = \{\}$ ⟨*proof*⟩

lemma *content-eq-0-1*: $content\ \{a..b{::}real\hat{\ }1\} = 0 \longleftrightarrow dest\text{-}vec1\ b \leq dest\text{-}vec1\ a$ ⟨*proof*⟩

lemma *content-pos-lt-eq*: $0 < content\ \{a..b\} \longleftrightarrow (\forall\, i.\ a\$i < b\$i)$ ⟨*proof*⟩

lemma *content-empty*[*simp*]: $content\ \{\} = 0$ ⟨*proof*⟩

lemma *content-subset*: **assumes** $\{a..b\} \subseteq \{c..d\}$ **shows** $content\ \{a..b{::}real\hat{\ }'n\} \leq content\ \{c..d\}$ ⟨*proof*⟩

lemma *content-lt-nz*: $0 < content\ \{a..b\} \longleftrightarrow content\ \{a..b\} \neq 0$ ⟨*proof*⟩

23.5 The notion of a gauge — simply an open set containing the point.

definition *gauge* **where** $gauge\ d \longleftrightarrow (\forall\, x.\ x{\in}(d\ x) \wedge open(d\ x))$

lemma *gaugeI*:**assumes** $\bigwedge x.\ x{\in}g\ x\ \bigwedge x.\ open\ (g\ x)$ **shows** $gauge\ g$ ⟨*proof*⟩

lemma *gaugeD*[*dest*]: **assumes** $gauge\ d$ **shows** $x{\in}d\ x\ open\ (d\ x)$ ⟨*proof*⟩

lemma *gauge-ball-dependent*: $\forall\, x.\ 0 < e\ x \Longrightarrow gauge\ (\lambda x.\ ball\ x\ (e\ x))$ ⟨*proof*⟩

lemma *gauge-ball*[*intro*]: $0 < e \Longrightarrow gauge\ (\lambda x.\ ball\ x\ e)$ ⟨*proof*⟩

lemma *gauge-trivial*[*intro*]: *gauge* $(\lambda x.\ ball\ x\ 1)$ ⟨*proof*⟩

lemma *gauge-inter*[*intro*]: *gauge d1* $\Longrightarrow$ *gauge d2* $\Longrightarrow$ *gauge* $(\lambda x.\ (d1\ x) \cap (d2\ x))$
⟨*proof*⟩

lemma *gauge-inters*: **assumes** *finite s* $\forall d{\in}s.\ gauge\ (f\ d)$ **shows** $gauge(\lambda x.\ \bigcap\ \{f\ d\ x \mid d.\ d \in s\})$ ⟨*proof*⟩

lemma *gauge-existence-lemma*: $(\forall x.\ \exists d{::}real.\ p\ x \longrightarrow 0 < d \wedge q\ d\ x) \longleftrightarrow (\forall x.\ \exists d{>}0.\ p\ x \longrightarrow q\ d\ x)$ ⟨*proof*⟩

23.6 Divisions.

definition *division-of* (**infixl** *division′-of 40*) **where**
s division-of i $\equiv$
finite s $\wedge$
$(\forall k{\in}s.\ k \subseteq i \wedge k \neq \{\} \wedge (\exists a\ b.\ k = \{a..b\})) \wedge$
$(\forall k1{\in}s.\ \forall k2{\in}s.\ k1 \neq k2 \longrightarrow interior(k1) \cap interior(k2) = \{\}) \wedge$
$(\bigcup s = i)$

lemma *division-ofD*[*dest*]: **assumes** *s division-of i*
shows*finite s* $\bigwedge k.\ k{\in}s \Longrightarrow k \subseteq i$ $\bigwedge k.\ k{\in}s \Longrightarrow k \neq \{\}$ $\bigwedge k.\ k{\in}s \Longrightarrow (\exists a\ b.\ k = \{a..b\})$
$\bigwedge k1\ k2.\ [\![k1{\in}s;\ k2{\in}s;\ k1 \neq k2]\!] \Longrightarrow interior(k1) \cap interior(k2) = \{\}$ $\bigcup s = i$
⟨*proof*⟩

lemma *division-ofI*:
assumes *finite s* $\bigwedge k.\ k{\in}s \Longrightarrow k \subseteq i$ $\bigwedge k.\ k{\in}s \Longrightarrow k \neq \{\}$ $\bigwedge k.\ k{\in}s \Longrightarrow (\exists a\ b.\ k = \{a..b\})$
$\bigwedge k1\ k2.\ [\![k1{\in}s;\ k2{\in}s;\ k1 \neq k2]\!] \Longrightarrow interior(k1) \cap interior(k2) = \{\}$ $\bigcup s = i$
shows *s division-of i* ⟨*proof*⟩

lemma *division-of-finite*: *s division-of i* $\Longrightarrow$ *finite s*
⟨*proof*⟩

lemma *division-of-self*[*intro*]: $\{a..b\} \neq \{\} \Longrightarrow \{\{a..b\}\}$ *division-of* $\{a..b\}$
⟨*proof*⟩

lemma *division-of-trivial*[*simp*]: *s division-of* $\{\} \longleftrightarrow s = \{\}$ ⟨*proof*⟩

lemma *division-of-sing*[*simp*]: *s division-of* $\{a..a{::}real\hat{\ }'n\} \longleftrightarrow s = \{\{a..a\}\}$ (**is** *?l = ?r*) ⟨*proof*⟩

lemma *elementary-empty*: **obtains** *p* **where** *p division-of* $\{\}$
⟨*proof*⟩

lemma *elementary-interval*: **obtains** *p* **where** *p division-of* $\{a..b\}$
⟨*proof*⟩

lemma *division-contains*: s *division-of* $i \Longrightarrow \forall x{\in}i.\ \exists k{\in}s.\ x \in k$
⟨*proof*⟩

lemma *forall-in-division*:
d *division-of* $i \Longrightarrow ((\forall x{\in}d.\ P\ x) \longleftrightarrow (\forall a\ b.\ \{a..b\} \in d \longrightarrow P\ \{a..b\}))$
⟨*proof*⟩

lemma *division-of-subset*: **assumes** p *division-of* $(\bigcup p)$ $q \subseteq p$ **shows** q *division-of* $(\bigcup q)$
⟨*proof*⟩

lemma *division-of-union-self*[*intro*]: p *division-of* $s \Longrightarrow p$ *division-of* $(\bigcup p)$ ⟨*proof*⟩

lemma *division-of-content-0*: **assumes** *content* $\{a..b\} = 0$ d *division-of* $\{a..b\}$ **shows** $\forall k{\in}d.$ *content* $k = 0$
⟨*proof*⟩

lemma *division-inter*: **assumes** *p1 division-of s1 p2 division-of* $(s2{::}(real\hat{\ }'a)\ set)$
shows $\{k1 \cap k2 \mid k1\ k2\ .k1 \in p1 \wedge k2 \in p2 \wedge k1 \cap k2 \neq \{\}\}$ *division-of* $(s1 \cap s2)$ (**is** $?A'$ *division-of* -) ⟨*proof*⟩

lemma *division-inter-1*: **assumes** d *division-of* i $\{a..b{::}real\hat{\ }'n\} \subseteq i$
shows $\{\ \{a..b\} \cap k\ |k.\ k \in d \wedge \{a..b\} \cap k \neq \{\}\ \}$ *division-of* $\{a..b\}$ ⟨*proof*⟩

lemma *elementary-inter*: **assumes** *p1 division-of s p2 division-of* $(t{::}(real\hat{\ }'n)\ set)$
shows $\exists p.\ p$ *division-of* $(s \cap t)$
⟨*proof*⟩

lemma *elementary-inters*: **assumes** *finite* f $f{\neq}\{\}$ $\forall s{\in}f.\ \exists p.\ p$ *division-of* $(s{::}(real\hat{\ }'n)\ set)$
shows $\exists p.\ p$ *division-of* $(\bigcap f)$ ⟨*proof*⟩

lemma *division-disjoint-union*:
assumes *p1 division-of s1 p2 division-of s2 interior s1* $\cap$ *interior s2* $= \{\}$
shows $(p1 \cup p2)$ *division-of* $(s1 \cup s2)$ ⟨*proof*⟩

lemma *partial-division-extend-1*:
assumes $\{c..d\} \subseteq \{a..b{::}real\hat{\ }'n\}$ $\{c..d\} \neq \{\}$
obtains p **where** p *division-of* $\{a..b\}$ $\{c..d\} \in p$
⟨*proof*⟩

lemma *partial-division-extend-interval*: **assumes** p *division-of* $(\bigcup p)$ $(\bigcup p) \subseteq \{a..b\}$
obtains q **where** $p \subseteq q$ q *division-of* $\{a..b{::}real\hat{\ }'n\}$ ⟨*proof*⟩

lemma *elementary-bounded*[*dest*]: p *division-of* $s \Longrightarrow$ *bounded* $(s{::}(real\hat{\ }'n)\ set)$
⟨*proof*⟩

lemma *elementary-subset-interval*: p *division-of* $s \Longrightarrow \exists a\ b.\ s \subseteq \{a..b{::}real\hat{\ }'n\}$

$\langle proof \rangle$

lemma *division-union-intervals-exists*: **assumes** $\{a..b::real\hat{\ }'n\} \neq \{\}$
obtains p **where** $(insert\ \{a..b\}\ p)$ *division-of* $(\{a..b\} \cup \{c..d\})$ $\langle proof \rangle$

lemma *division-of-unions*: **assumes** *finite f* $\bigwedge p.\ p{\in}f \Longrightarrow p$ *division-of* $(\bigcup p)$
$\bigwedge k1\ k2.\ [\![k1 \in \bigcup f;\ k2 \in \bigcup f;\ k1 \neq k2]\!] \Longrightarrow interior\ k1 \cap interior\ k2 = \{\}$
shows $\bigcup f$ *division-of* $\bigcup\bigcup f$ $\langle proof \rangle$

lemma *elementary-union-interval*: **assumes** p *division-of* $\bigcup p$
obtains q **where** q *division-of* $(\{a..b::real\hat{\ }'n\} \cup \bigcup p)$ $\langle proof \rangle$

lemma *elementary-unions-intervals*:
assumes *finite f* $\bigwedge s.\ s \in f \Longrightarrow \exists\, a\ b.\ s = \{a..b::real\hat{\ }'n\}$
obtains p **where** p *division-of* $(\bigcup f)$ $\langle proof \rangle$

lemma *elementary-union*: **assumes** *ps division-of s pt division-of* $(t::(real\hat{\ }'n)$ *set*)
obtains p **where** p *division-of* $(s \cup t)$
$\langle proof \rangle$

lemma *partial-division-extend*: **fixes** $t::(real\hat{\ }'n)$ *set*
assumes *p division-of s q division-of t* $s \subseteq t$
obtains r **where** $p \subseteq r$ *r division-of t* $\langle proof \rangle$

23.7 Tagged (partial) divisions.

definition *tagged-partial-division-of* (**infixr** *tagged'-partial'-division'-of 40*) **where**
$(s$ *tagged-partial-division-of* $i) \equiv$
finite $s \wedge$
$(\forall x\ k.\ (x,k) \in s \longrightarrow x \in k \wedge k \subseteq i \wedge (\exists\, a\ b.\ k = \{a..b\})) \wedge$
$(\forall x1\ k1\ x2\ k2.\ (x1,k1) \in s \wedge (x2,k2) \in s \wedge ((x1,k1) \neq (x2,k2))$
$\longrightarrow (interior(k1) \cap interior(k2) = \{\}))$

lemma *tagged-partial-division-ofD*[*dest*]: **assumes** *s tagged-partial-division-of i*
shows *finite s* $\bigwedge x\ k.\ (x,k) \in s \Longrightarrow x \in k$ $\bigwedge x\ k.\ (x,k) \in s \Longrightarrow k \subseteq i$
$\bigwedge x\ k.\ (x,k) \in s \Longrightarrow \exists\, a\ b.\ k = \{a..b\}$
$\bigwedge x1\ k1\ x2\ k2.\ (x1,k1) \in s \Longrightarrow (x2,k2) \in s \Longrightarrow (x1,k1) \neq (x2,k2) \Longrightarrow interior(k1) \cap interior(k2) = \{\}$
$\langle proof \rangle$

definition *tagged-division-of* (**infixr** *tagged'-division'-of 40*) **where**
$(s$ *tagged-division-of* $i) \equiv$
$(s$ *tagged-partial-division-of* $i) \wedge (\bigcup\{k.\ \exists\, x.\ (x,k) \in s\} = i)$

lemma *tagged-division-of-finite*[*dest*]: *s tagged-division-of i* $\Longrightarrow$ *finite s*
$\langle proof \rangle$

lemma *tagged-division-of*:

$(s\ \textit{tagged-division-of}\ i) \longleftrightarrow$
$\quad \textit{finite}\ s \wedge$
$\quad (\forall x\ k.\ (x,k) \in s$
$\qquad \longrightarrow x \in k \wedge k \subseteq i \wedge (\exists a\ b.\ k = \{a..b\})) \wedge$
$\quad (\forall x1\ k1\ x2\ k2.\ (x1,k1) \in s \wedge (x2,k2) \in s \wedge \sim((x1,k1) = (x2,k2))$
$\qquad\quad \longrightarrow (\textit{interior}(k1) \cap \textit{interior}(k2) = \{\})) \wedge$
$\quad (\bigcup\{k.\ \exists x.\ (x,k) \in s\} = i)$
$\langle \textit{proof} \rangle$

lemma *tagged-division-ofI*: **assumes**
$\textit{finite}\ s\ \bigwedge x\ k.\ (x,k) \in s \Longrightarrow x \in k\ \bigwedge x\ k.\ (x,k) \in s \Longrightarrow k \subseteq i\ \ \bigwedge x\ k.\ (x,k) \in s \Longrightarrow \exists a\ b.\ k = \{a..b\}$
$\bigwedge x1\ k1\ x2\ k2.\ (x1,k1) \in s \Longrightarrow (x2,k2) \in s \Longrightarrow \sim((x1,k1) = (x2,k2)) \Longrightarrow (\textit{interior}(k1) \cap \textit{interior}(k2) = \{\})$
$(\bigcup\{k.\ \exists x.\ (x,k) \in s\} = i)$
shows $s\ \textit{tagged-division-of}\ i$
$\langle \textit{proof} \rangle$

lemma *tagged-division-ofD*[*dest*]: **assumes** $s\ \textit{tagged-division-of}\ i$
shows $\textit{finite}\ s\ \bigwedge x\ k.\ (x,k) \in s \Longrightarrow x \in k\ \bigwedge x\ k.\ (x,k) \in s \Longrightarrow k \subseteq i\ \ \bigwedge x\ k.\ (x,k) \in s \Longrightarrow \exists a\ b.\ k = \{a..b\}$
$\bigwedge x1\ k1\ x2\ k2.\ (x1,k1) \in s \Longrightarrow (x2,k2) \in s \Longrightarrow \sim((x1,k1) = (x2,k2)) \Longrightarrow (\textit{interior}(k1) \cap \textit{interior}(k2) = \{\})$
$(\bigcup\{k.\ \exists x.\ (x,k) \in s\} = i)$ $\langle \textit{proof} \rangle$

lemma *division-of-tagged-division*: **assumes**$s\ \textit{tagged-division-of}\ i$ **shows** $(\textit{snd}\ `\ s)\ \textit{division-of}\ i$
$\langle \textit{proof} \rangle$

lemma *partial-division-of-tagged-division*: **assumes** $s\ \textit{tagged-partial-division-of}\ i$
shows $(\textit{snd}\ `\ s)\ \textit{division-of}\ \bigcup(\textit{snd}\ `\ s)$
$\langle \textit{proof} \rangle$

lemma *tagged-partial-division-subset*: **assumes** $s\ \textit{tagged-partial-division-of}\ i\ t \subseteq s$
shows $t\ \textit{tagged-partial-division-of}\ i$
$\langle \textit{proof} \rangle$

lemma *setsum-over-tagged-division-lemma*: **fixes** $d::(\textit{real}\hat{\ }'m)\ \textit{set} \Rightarrow 'a::\textit{real-normed-vector}$
assumes $p\ \textit{tagged-division-of}\ i\ \bigwedge u\ v.\ \{u..v\} \neq \{\} \Longrightarrow \textit{content}\ \{u..v\} = 0 \Longrightarrow d\ \{u..v\} = 0$
shows $\textit{setsum}\ (\lambda(x,k).\ d\ k)\ p = \textit{setsum}\ d\ (\textit{snd}\ `\ p)$
$\langle \textit{proof} \rangle$

lemma *tag-in-interval*: $p\ \textit{tagged-division-of}\ i \Longrightarrow (x,k) \in p \Longrightarrow x \in i$ $\langle \textit{proof} \rangle$

lemma *tagged-division-of-empty*: $\{\}\ \textit{tagged-division-of}\ \{\}$
$\langle \textit{proof} \rangle$

lemma *tagged-partial-division-of-trivial*[*simp*]:

p *tagged-partial-division-of* $\{\} \longleftrightarrow p = \{\}$
⟨*proof*⟩

lemma *tagged-division-of-trivial*[*simp*]:
p *tagged-division-of* $\{\} \longleftrightarrow p = \{\}$
⟨*proof*⟩

lemma *tagged-division-of-self*:
$x \in \{a..b\} \Longrightarrow \{(x,\{a..b\})\}$ *tagged-division-of* $\{a..b\}$
⟨*proof*⟩

lemma *tagged-division-union*:
assumes *p1 tagged-division-of s1 p2 tagged-division-of s2 interior s1* $\cap$ *interior s2* $= \{\}$
shows $(p1 \cup p2)$ *tagged-division-of* $(s1 \cup s2)$
⟨*proof*⟩

lemma *tagged-division-unions*:
assumes *finite iset* $\forall\, i{\in}iset.$ $(pfn(i)$ *tagged-division-of* $i)$
$\forall\, i1 \in iset.\ \forall\, i2 \in iset.\ {}^{\sim}(i1 = i2) \longrightarrow (interior(i1) \cap interior(i2) = \{\})$
shows $\bigcup(pfn\ `\ iset)$ *tagged-division-of* $(\bigcup iset)$
⟨*proof*⟩

lemma *tagged-partial-division-of-union-self*:
assumes *p tagged-partial-division-of s* **shows** *p tagged-division-of* $(\bigcup(snd\ `\ p))$
⟨*proof*⟩

lemma *tagged-division-of-union-self*: **assumes** *p tagged-division-of s*
shows *p tagged-division-of* $(\bigcup(snd\ `\ p))$
⟨*proof*⟩

23.8 Fine-ness of a partition w.r.t. a gauge.

definition *fine* (**infixr** *fine 46*) **where**
d *fine* $s \longleftrightarrow (\forall\,(x,k) \in s.\ k \subseteq d(x))$

lemma *fineI*: **assumes** $\bigwedge x\ k.\ (x,k) \in s \Longrightarrow k \subseteq d\ x$
shows d *fine* s ⟨*proof*⟩

lemma *fineD*[*dest*]: **assumes** d *fine* s
shows $\bigwedge x\ k.\ (x,k) \in s \Longrightarrow k \subseteq d\ x$ ⟨*proof*⟩

lemma *fine-inter*: $(\lambda x.\ d1\ x \cap d2\ x)$ *fine* $p \longleftrightarrow d1$ *fine* $p \wedge d2$ *fine* p
⟨*proof*⟩

lemma *fine-inters*:
$(\lambda x.\ \bigcap\ \{f\ d\ x \mid d.\ \ d \in s\})$ *fine* $p \longleftrightarrow (\forall\, d{\in}s.\ (f\ d)$ *fine* $p)$
⟨*proof*⟩

lemma *fine-union*:
$d\ fine\ p1 \Longrightarrow d\ fine\ p2 \Longrightarrow d\ fine\ (p1 \cup p2)$
$\langle proof \rangle$

lemma *fine-unions*:$(\bigwedge p.\ p \in ps \Longrightarrow d\ fine\ p) \Longrightarrow d\ fine\ (\bigcup ps)$
$\langle proof \rangle$

lemma *fine-subset*: $p \subseteq q \Longrightarrow d\ fine\ q \Longrightarrow d\ fine\ p$
$\langle proof \rangle$

23.9 Gauge integral. Define on compact intervals first, then use a limit.

definition *has-integral-compact-interval* (**infixr** *has′-integral′-compact′-interval 46*) **where**
$(f\ has\text{-}integral\text{-}compact\text{-}interval\ y)\ i \equiv$
$(\forall e>0.\ \exists d.\ gauge\ d \wedge$
$(\forall p.\ p\ tagged\text{-}division\text{-}of\ i \wedge d\ fine\ p$
$\longrightarrow norm(setsum\ (\lambda(x,k).\ content\ k *_R f\ x)\ p - y) < e))$

definition *has-integral* (**infixr** *has′-integral 46*) **where**
$((f::(real\hat{\ }'n \Rightarrow 'b::real\text{-}normed\text{-}vector))\ has\text{-}integral\ y)\ i \equiv$
$if\ (\exists a\ b.\ i = \{a..b\})\ then\ (f\ has\text{-}integral\text{-}compact\text{-}interval\ y)\ i$
$else\ (\forall e>0.\ \exists B>0.\ \forall a\ b.\ ball\ 0\ B \subseteq \{a..b\}$
$\longrightarrow (\exists z.\ ((\lambda x.\ if\ x \in i\ then\ f\ x\ else\ 0)\ has\text{-}integral\text{-}compact\text{-}interval\ z)$
$\{a..b\} \wedge$
$norm(z - y) < e))$

lemma *has-integral*:
$(f\ has\text{-}integral\ y)\ (\{a..b\}) \longleftrightarrow$
$(\forall e>0.\ \exists d.\ gauge\ d \wedge (\forall p.\ p\ tagged\text{-}division\text{-}of\ \{a..b\} \wedge d\ fine\ p$
$\longrightarrow norm(setsum\ (\lambda(x,k).\ content(k) *_R f\ x)\ p - y) < e))$
$\langle proof \rangle$

lemma *has-integralD*[*dest*]: **assumes**
$(f\ has\text{-}integral\ y)\ (\{a..b\})\ e>0$
obtains d **where** $gauge\ d\ \bigwedge p.\ p\ tagged\text{-}division\text{-}of\ \{a..b\} \Longrightarrow d\ fine\ p$
$\Longrightarrow norm(setsum\ (\lambda(x,k).\ content(k) *_R f(x))\ p - y) < e$
$\langle proof \rangle$

lemma *has-integral-alt*:
$(f\ has\text{-}integral\ y)\ i \longleftrightarrow$
$(if\ (\exists a\ b.\ i = \{a..b\})\ then\ (f\ has\text{-}integral\ y)\ i$
$else\ (\forall e>0.\ \exists B>0.\ \forall a\ b.\ ball\ 0\ B \subseteq \{a..b\}$
$\longrightarrow (\exists z.\ ((\lambda x.\ if\ x \in i\ then\ f(x)\ else\ 0)$
$has\text{-}integral\ z)\ (\{a..b\}) \wedge$
$norm(z - y) < e)))$
$\langle proof \rangle$

lemma *has-integral-altD*:
assumes $(f\ has\text{-}integral\ y)\ i\ \neg\ (\exists\ a\ b.\ i = \{a..b\})\ e>0$
obtains B **where** $B>0\ \forall\ a\ b.\ ball\ 0\ B \subseteq \{a..b\} \longrightarrow (\exists\ z.\ ((\lambda x.\ if\ x \in i\ then\ f(x)\ else\ 0)\ has\text{-}integral\ z)\ (\{a..b\}) \wedge norm(z - y) < e)$
⟨*proof*⟩

definition *integrable-on* (**infixr** *integrable′-on 46*) **where**
$(f\ integrable\text{-}on\ i) \equiv \exists\ y.\ (f\ has\text{-}integral\ y)\ i$

definition $integral\ i\ f \equiv SOME\ y.\ (f\ has\text{-}integral\ y)\ i$

lemma *integrable-integral*[*dest*]:
$f\ integrable\text{-}on\ i \Longrightarrow (f\ has\text{-}integral\ (integral\ i\ f))\ i$
⟨*proof*⟩

lemma *has-integral-integrable*[*intro*]: $(f\ has\text{-}integral\ i)\ s \Longrightarrow f\ integrable\text{-}on\ s$
⟨*proof*⟩

lemma *has-integral-integral*:$f\ integrable\text{-}on\ s \longleftrightarrow (f\ has\text{-}integral\ (integral\ s\ f))\ s$
⟨*proof*⟩

lemma *has-integral-vec1*: **assumes** $(f\ has\text{-}integral\ k)\ \{a..b\}$
shows $((\lambda x.\ vec1\ (f\ x))\ has\text{-}integral\ (vec1\ k))\ \{a..b\}$
⟨*proof*⟩

lemma *setsum-content-null*:
assumes $content(\{a..b\}) = 0\ p\ tagged\text{-}division\text{-}of\ \{a..b\}$
shows $setsum\ (\lambda(x,k).\ content\ k *_R f\ x)\ p = (0::'a::real\text{-}normed\text{-}vector)$
⟨*proof*⟩

23.10 Some basic combining lemmas.

lemma *tagged-division-unions-exists*:
assumes $finite\ iset\ \forall\ i \in iset.\ \exists\ p.\ p\ tagged\text{-}division\text{-}of\ i \wedge d\ fine\ p$
$\forall\ i1 \in iset.\ \forall\ i2 \in iset.\ {\sim}(i1 = i2) \longrightarrow (interior(i1) \cap interior(i2) = \{\})\ (\bigcup iset = i)$
obtains p **where** $p\ tagged\text{-}division\text{-}of\ i\ d\ fine\ p$
⟨*proof*⟩

23.11 The set we're concerned with must be closed.

lemma *division-of-closed*: $s\ division\text{-}of\ i \Longrightarrow closed\ (i::(real\hat{}'n)\ set)$
⟨*proof*⟩

23.12 General bisection principle for intervals; might be useful elsewhere.

lemma *interval-bisection-step*:
assumes $P\ \{\}\ (\forall\ s\ t.\ P\ s \wedge P\ t \wedge interior(s) \cap interior(t) = \{\} \longrightarrow P(s \cup t))$
${\sim}(P\ \{a..b::real\hat{}'n\})$

obtains $c\ d$ **where** $^\sim(P\{c..d\})$
$\forall\, i.\ a\$i \leq c\$i \wedge c\$i \leq d\$i \wedge d\$i \leq b\$i \wedge 2 * (d\$i - c\$i) \leq b\$i - a\$i$
⟨*proof*⟩

lemma *interval-bisection*:
assumes $P\ \{\}\ (\forall\, s\ t.\ P\ s \wedge P\ t \wedge interior(s) \cap interior(t) = \{\} \longrightarrow P(s \cup t))$
$\neg\ P\ \{a..b::real\hat{\ }'n\}$
obtains x **where** $x \in \{a..b\}\ \forall\, e>0.\ \exists\, c\ d.\ x \in \{c..d\} \wedge \{c..d\} \subseteq ball\ x\ e \wedge \{c..d\} \subseteq \{a..b\} \wedge {}^\sim P(\{c..d\})$
⟨*proof*⟩

23.13 Cousin's lemma.

lemma *fine-division-exists*: **assumes** *gauge g*
obtains p **where** p *tagged-division-of* $\{a..b::real\hat{\ }'n\}$ *g fine p*
⟨*proof*⟩

23.14 Basic theorems about integrals.

lemma *has-integral-unique*: **fixes** $f::real\hat{\ }'n \Rightarrow {}'a::real\text{-}normed\text{-}vector$
assumes $(f\ has\text{-}integral\ k1)\ i\ (f\ has\text{-}integral\ k2)\ i$ **shows** $k1 = k2$
⟨*proof*⟩

lemma *integral-unique*[*intro*]:
$(f\ has\text{-}integral\ y)\ k \Longrightarrow integral\ k\ f = y$
⟨*proof*⟩

lemma *has-integral-is-0*: **fixes** $f::real\hat{\ }'n \Rightarrow {}'a::real\text{-}normed\text{-}vector$
assumes $\forall\, x{\in}s.\ f\ x = 0$ **shows** $(f\ has\text{-}integral\ 0)\ s$
⟨*proof*⟩

lemma *has-integral-0*[*simp*]: $((\lambda x::real\hat{\ }'n.\ 0)\ has\text{-}integral\ 0)\ s$
⟨*proof*⟩

lemma *has-integral-0-eq*[*simp*]: $((\lambda x.\ 0)\ has\text{-}integral\ i)\ s \longleftrightarrow i = 0$
⟨*proof*⟩

lemma *has-integral-linear*: **fixes** $f::real\hat{\ }'n \Rightarrow {}'a::real\text{-}normed\text{-}vector$
assumes $(f\ has\text{-}integral\ y)\ s\ bounded\text{-}linear\ h$ **shows** $((h\ o\ f)\ has\text{-}integral\ ((h\ y)))\ s$
⟨*proof*⟩

lemma *has-integral-cmul*:
shows $(f\ has\text{-}integral\ k)\ s \Longrightarrow ((\lambda x.\ c *_R f\ x)\ has\text{-}integral\ (c *_R k))\ s$
⟨*proof*⟩

lemma *has-integral-neg*:
shows $(f\ has\text{-}integral\ k)\ s \Longrightarrow ((\lambda x.\ -(f\ x))\ has\text{-}integral\ (-k))\ s$
⟨*proof*⟩

lemma *has-integral-add*: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::real\text{-}normed\text{-}vector$
assumes $(f\ has\text{-}integral\ k)\ s$ $(g\ has\text{-}integral\ l)\ s$
shows $((\lambda x.\ f\ x + g\ x)\ has\text{-}integral\ (k + l))\ s$
⟨*proof*⟩

lemma *has-integral-sub*:
shows $(f\ has\text{-}integral\ k)\ s \Longrightarrow (g\ has\text{-}integral\ l)\ s \Longrightarrow ((\lambda x.\ f(x) - g(x))\ has\text{-}integral\ (k - l))\ s$
⟨*proof*⟩

lemma *integral-0*: $integral\ s\ (\lambda x::real\hat{\ }'n.\ 0::real\hat{\ }'m) = 0$
⟨*proof*⟩

lemma *integral-add*:
shows $f\ integrable\text{-}on\ s \Longrightarrow g\ integrable\text{-}on\ s \Longrightarrow$
$integral\ s\ (\lambda x.\ f\ x + g\ x) = integral\ s\ f + integral\ s\ g$
⟨*proof*⟩

lemma *integral-cmul*:
shows $f\ integrable\text{-}on\ s \Longrightarrow integral\ s\ (\lambda x.\ c *_R f\ x) = c *_R integral\ s\ f$
⟨*proof*⟩

lemma *integral-neg*:
shows $f\ integrable\text{-}on\ s \Longrightarrow integral\ s\ (\lambda x.\ - f\ x) = - integral\ s\ f$
⟨*proof*⟩

lemma *integral-sub*:
shows $f\ integrable\text{-}on\ s \Longrightarrow g\ integrable\text{-}on\ s \Longrightarrow integral\ s\ (\lambda x.\ f\ x - g\ x) = integral\ s\ f - integral\ s\ g$
⟨*proof*⟩

lemma *integrable-0*: $(\lambda x.\ 0)\ integrable\text{-}on\ s$
⟨*proof*⟩

lemma *integrable-add*:
shows $f\ integrable\text{-}on\ s \Longrightarrow g\ integrable\text{-}on\ s \Longrightarrow (\lambda x.\ f\ x + g\ x)\ integrable\text{-}on\ s$
⟨*proof*⟩

lemma *integrable-cmul*:
shows $f\ integrable\text{-}on\ s \Longrightarrow (\lambda x.\ c *_R f(x))\ integrable\text{-}on\ s$
⟨*proof*⟩

lemma *integrable-neg*:
shows $f\ integrable\text{-}on\ s \Longrightarrow (\lambda x.\ -f(x))\ integrable\text{-}on\ s$
⟨*proof*⟩

lemma *integrable-sub*:
shows $f\ integrable\text{-}on\ s \Longrightarrow g\ integrable\text{-}on\ s \Longrightarrow (\lambda x.\ f\ x - g\ x)\ integrable\text{-}on\ s$
⟨*proof*⟩

lemma *integrable-linear*:
shows *f integrable-on s* $\Longrightarrow$ *bounded-linear h* $\Longrightarrow$ (*h o f*) *integrable-on s*
⟨*proof*⟩

lemma *integral-linear*:
shows *f integrable-on s* $\Longrightarrow$ *bounded-linear h* $\Longrightarrow$ *integral s* (*h o f*) = *h*(*integral s f*)
⟨*proof*⟩

lemma *integral-component-eq*[*simp*]: **fixes** *f*::*real*ˆ′*n* ⇒ *real*ˆ′*m*
assumes *f integrable-on s* **shows** *integral s* (λx. *f x* $ *k*) = *integral s f* $ *k*
⟨*proof*⟩

lemma *has-integral-setsum*:
assumes *finite t* ∀ *a*∈*t*. ((*f a*) *has-integral* (*i a*)) *s*
shows ((λx. *setsum* (λa. *f a x*) *t*) *has-integral* (*setsum i t*)) *s*
⟨*proof*⟩

lemma *integral-setsum*:
shows *finite t* $\Longrightarrow$ ∀ *a*∈*t*. (*f a*) *integrable-on s* $\Longrightarrow$
integral s (λx. *setsum* (λa. *f a x*) *t*) = *setsum* (λa. *integral s* (*f a*)) *t*
⟨*proof*⟩

lemma *integrable-setsum*:
shows *finite t* $\Longrightarrow$ ∀ *a* ∈ *t*.(*f a*) *integrable-on s* $\Longrightarrow$ (λx. *setsum* (λa. *f a x*) *t*) *integrable-on s*
⟨*proof*⟩

lemma *has-integral-eq*:
assumes ∀ *x*∈*s*. *f x* = *g x* (*f has-integral k*) *s* **shows** (*g has-integral k*) *s*
⟨*proof*⟩

lemma *integrable-eq*:
shows ∀ *x*∈*s*. *f x* = *g x* $\Longrightarrow$ *f integrable-on s* $\Longrightarrow$ *g integrable-on s*
⟨*proof*⟩

lemma *has-integral-eq-eq*:
shows ∀ *x*∈*s*. *f x* = *g x* $\Longrightarrow$ ((*f has-integral k*) *s* $\longleftrightarrow$ (*g has-integral k*) *s*)
⟨*proof*⟩

lemma *has-integral-null*[*dest*]:
assumes *content*({*a*..*b*}) = *0* **shows** (*f has-integral 0*) ({*a*..*b*})
⟨*proof*⟩

lemma *has-integral-null-eq*[*simp*]:
shows *content*({*a*..*b*}) = *0* $\Longrightarrow$ ((*f has-integral i*) ({*a*..*b*}) $\longleftrightarrow$ *i* = *0*)
⟨*proof*⟩

lemma *integral-null*[*dest*]: **shows** $content(\{a..b\}) = 0 \Longrightarrow integral(\{a..b\})\ f = 0$
⟨*proof*⟩

lemma *integrable-on-null*[*dest*]: **shows** $content(\{a..b\}) = 0 \Longrightarrow f$ *integrable-on* $\{a..b\}$
⟨*proof*⟩

lemma *has-integral-empty*[*intro*]: **shows** $(f\ has\text{-}integral\ 0)\ \{\}$
⟨*proof*⟩

lemma *has-integral-empty-eq*[*simp*]: **shows** $(f\ has\text{-}integral\ i)\ \{\} \longleftrightarrow i = 0$
⟨*proof*⟩

lemma *integrable-on-empty*[*intro*]: **shows** f *integrable-on* $\{\}$ ⟨*proof*⟩

lemma *integral-empty*[*simp*]: **shows** *integral* $\{\}\ f = 0$
⟨*proof*⟩

lemma *has-integral-refl*[*intro*]: **shows** $(f\ has\text{-}integral\ 0)\ \{a..a\}\ (f\ has\text{-}integral\ 0)$ $\{a\}$
⟨*proof*⟩

lemma *integrable-on-refl*[*intro*]: **shows** f *integrable-on* $\{a..a\}$ ⟨*proof*⟩

lemma *integral-refl*: **shows** *integral* $\{a..a\}\ f = 0$ ⟨*proof*⟩

23.15 Cauchy-type criterion for integrability.

lemma *integrable-cauchy*: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::\{real\text{-}normed\text{-}vector,complete\text{-}space\}$

shows f *integrable-on* $\{a..b\} \longleftrightarrow$
$(\forall\ e>0.\exists\ d.\ gauge\ d \wedge (\forall\ p1\ p2.\ p1\ tagged\text{-}division\text{-}of\ \{a..b\} \wedge d\ fine\ p1 \wedge$
$p2\ tagged\text{-}division\text{-}of\ \{a..b\} \wedge d\ fine\ p2$
$\longrightarrow norm(setsum\ (\lambda(x,k).\ content\ k *_R f\ x)\ p1\ -$
$setsum\ (\lambda(x,k).\ content\ k *_R f\ x)\ p2) < e))$ (**is** $?l =$
$(\forall\ e>0.\ \exists\ d.\ ?P\ e\ d))$
⟨*proof*⟩

23.16 Additivity of integral on abutting intervals.

lemma *interval-split*:
$\{a..b::real\hat{\ }'n\} \cap \{x.\ x\$k \leq c\} = \{a\ ..\ (\chi\ i.\ if\ i = k\ then\ min\ (b\$k)\ c\ else\ b\$i)\}$
$\{a..b\} \cap \{x.\ x\$k \geq c\} = \{(\chi\ i.\ if\ i = k\ then\ max\ (a\$k)\ c\ else\ a\$i)\ ..\ b\}$
⟨*proof*⟩

lemma *content-split*:
$content\ \{a..b::real\hat{\ }'n\} = content(\{a..b\} \cap \{x.\ x\$k \leq c\}) + content(\{a..b\} \cap \{x.$ $x\$k >= c\})$
⟨*proof*⟩

lemma *division-split-left-inj*:
assumes *d division-of i k1* $\in$ *d k2* $\in$ *d k1* $\neq$ *k2*
k1 $\cap$ $\{x::real^{\wedge}'n.\ x\$k \leq c\} = k2 \cap \{x.\ x\$k \leq c\}$
shows *content*$(k1 \cap \{x.\ x\$k \leq c\}) = 0$
⟨*proof*⟩

lemma *division-split-right-inj*:
assumes *d division-of i k1* $\in$ *d k2* $\in$ *d k1* $\neq$ *k2*
k1 $\cap$ $\{x::real^{\wedge}'n.\ x\$k \geq c\} = k2 \cap \{x.\ x\$k \geq c\}$
shows *content*$(k1 \cap \{x.\ x\$k \geq c\}) = 0$
⟨*proof*⟩

lemma *tagged-division-split-left-inj*:
assumes *d tagged-division-of i* $(x1,k1) \in d$ $(x2,k2) \in d$ $k1 \neq k2$ $k1 \cap \{x.\ x\$k \leq c\} = k2 \cap \{x.\ x\$k \leq c\}$
shows *content*$(k1 \cap \{x.\ x\$k \leq c\}) = 0$
⟨*proof*⟩

lemma *tagged-division-split-right-inj*:
assumes *d tagged-division-of i* $(x1,k1) \in d$ $(x2,k2) \in d$ $k1 \neq k2$ $k1 \cap \{x.\ x\$k \geq c\} = k2 \cap \{x.\ x\$k \geq c\}$
shows *content*$(k1 \cap \{x.\ x\$k \geq c\}) = 0$
⟨*proof*⟩

lemma *division-split*:
assumes *p division-of* $\{a..b::real^{\wedge}'n\}$
shows $\{l \cap \{x.\ x\$k \leq c\} \mid l.\ l \in p \wedge {}^{\sim}(l \cap \{x.\ x\$k \leq c\} = \{\})\}$ *division-of* $(\{a..b\} \cap \{x.\ x\$k \leq c\})$ (**is** *?p1 division-of ?I1*) **and**
$\{l \cap \{x.\ x\$k \geq c\} \mid l.\ l \in p \wedge {}^{\sim}(l \cap \{x.\ x\$k \geq c\} = \{\})\}$ *division-of* $(\{a..b\} \cap \{x.\ x\$k \geq c\})$ (**is** *?p2 division-of ?I2*)
⟨*proof*⟩

lemma *has-integral-split*: **fixes** $f::real^{\wedge}'n \Rightarrow 'a::real\text{-}normed\text{-}vector$
assumes $(f$ *has-integral* $i)$ $(\{a..b\} \cap \{x.\ x\$k \leq c\})$ $(f$ *has-integral* $j)$ $(\{a..b\} \cap \{x.\ x\$k \geq c\})$
shows $(f$ *has-integral* $(i + j))$ $(\{a..b\})$
⟨*proof*⟩

23.17 A sort of converse, integrability on subintervals.

lemma *tagged-division-union-interval*:
assumes *p1 tagged-division-of* $(\{a..b\} \cap \{x::real^{\wedge}'n.\ x\$k \leq (c::real)\})$ *p2 tagged-division-of* $(\{a..b\} \cap \{x.\ x\$k \geq c\})$
shows $(p1 \cup p2)$ *tagged-division-of* $(\{a..b\})$
⟨*proof*⟩

lemma *has-integral-separate-sides*: **fixes** $f::real^{\wedge}'m \Rightarrow 'a::real\text{-}normed\text{-}vector$
assumes $(f$ *has-integral* $i)$ $(\{a..b\})$ $e>0$
obtains *d* **where** *gauge d* $(\forall\, p1\ p2.\ p1$ *tagged-division-of* $(\{a..b\} \cap \{x.\ x\$k \leq$

$c\}) \wedge d$ *fine p1* $\wedge$
p2 tagged-division-of $(\{a..b\} \cap \{x.\ x\$k \geq c\}) \wedge d$ *fine p2*
$\longrightarrow norm((setsum\ (\lambda(x,k).\ content\ k *_R f\ x)\ p1\ +$
$setsum\ (\lambda(x,k).\ content\ k *_R f\ x)\ p2) - i) < e)$
⟨*proof*⟩

lemma *integrable-split*[*intro*]: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::\{real\text{-}normed\text{-}vector, complete\text{-}space\}$
assumes *f integrable-on* $\{a..b\}$
shows *f integrable-on* $(\{a..b\} \cap \{x.\ x\$k \leq c\})$ (**is** *?t1*) **and** *f integrable-on* $(\{a..b\} \cap \{x.\ x\$k \geq c\})$ (**is** *?t2*)
⟨*proof*⟩

23.18 Generalized notion of additivity.

definition *neutral opp* $= (SOME\ x.\ \forall y.\ opp\ x\ y = y \wedge opp\ y\ x = y)$

definition *operative* $:: ('a \Rightarrow 'a \Rightarrow 'a) \Rightarrow ((real\hat{\ }'n)\ set \Rightarrow 'a) \Rightarrow bool$ **where**
operative opp f $\equiv$
$(\forall\ a\ b.\ content\ \{a..b\} = 0 \longrightarrow f\ \{a..b\} = neutral(opp)) \wedge$
$(\forall\ a\ b\ c\ k.\ f(\{a..b\}) =$
$opp\ (f(\{a..b\} \cap \{x.\ x\$k \leq c\}))$
$(f(\{a..b\} \cap \{x.\ x\$k \geq c\})))$

lemma *operativeD*[*dest*]: **assumes** *operative opp f*
shows $\bigwedge a\ b.\ content\ \{a..b\} = 0 \Longrightarrow f\ \{a..b\} = neutral(opp)$
$\bigwedge a\ b\ c\ k.\ f(\{a..b\}) = opp\ (f(\{a..b\} \cap \{x.\ x\$k \leq c\}))\ (f(\{a..b\} \cap \{x.\ x\$k \geq c\}))$
⟨*proof*⟩

lemma *operative-trivial*:
operative opp f $\Longrightarrow content(\{a..b\}) = 0 \Longrightarrow f(\{a..b\}) = neutral\ opp$
⟨*proof*⟩

lemma *property-empty-interval*:
$(\forall\ a\ b.\ content(\{a..b\}) = 0 \longrightarrow P(\{a..b\})) \Longrightarrow P\ \{\}$
⟨*proof*⟩

lemma *operative-empty*: *operative opp f* $\Longrightarrow f\ \{\} = neutral\ opp$
⟨*proof*⟩

23.19 Using additivity of lifted function to encode definedness.

lemma *forall-option*: $(\forall x.\ P\ x) \longleftrightarrow P\ None \wedge (\forall x.\ P(Some\ x))$
⟨*proof*⟩

lemma *exists-option*:
$(\exists x.\ P\ x) \longleftrightarrow P\ None \vee (\exists x.\ P(Some\ x))$
⟨*proof*⟩

fun *lifted* **where**
$lifted\ (opp::'a \Rightarrow 'a \Rightarrow 'b)\ (Some\ x)\ (Some\ y) = Some(opp\ x\ y)\ |$
$lifted\ opp\ None\ \text{-} = (None::'b\ option)\ |$
$lifted\ opp\ \text{-}\ None = None$

lemma *lifted-simp-1*[*simp*]: $lifted\ opp\ v\ None = None$
⟨*proof*⟩

definition $monoidal\ opp \equiv\ (\forall x\ y.\ opp\ x\ y = opp\ y\ x) \wedge$
$(\forall x\ y\ z.\ opp\ x\ (opp\ y\ z) = opp\ (opp\ x\ y)\ z) \wedge$
$(\forall x.\ opp\ (neutral\ opp)\ x = x)$

lemma *monoidalI*: **assumes** $\bigwedge x\ y.\ opp\ x\ y = opp\ y\ x$
$\bigwedge x\ y\ z.\ opp\ x\ (opp\ y\ z) = opp\ (opp\ x\ y)\ z$
$\bigwedge x.\ opp\ (neutral\ opp)\ x = x$ **shows** $monoidal\ opp$
⟨*proof*⟩

lemma *monoidal-ac*: **assumes** $monoidal\ opp$
shows $opp\ (neutral\ opp)\ a = a\ opp\ a\ (neutral\ opp) = a\ opp\ a\ b = opp\ b\ a$
$opp\ (opp\ a\ b)\ c = opp\ a\ (opp\ b\ c)\ \ opp\ a\ (opp\ b\ c) = opp\ b\ (opp\ a\ c)$
⟨*proof*⟩

lemma *monoidal-simps*[*simp*]: **assumes** $monoidal\ opp$
shows $opp\ (neutral\ opp)\ a = a\ opp\ a\ (neutral\ opp) = a$
⟨*proof*⟩

lemma *neutral-lifted*[*cong*]: **assumes** $monoidal\ opp$
shows $neutral\ (lifted\ opp) = Some(neutral\ opp)$
⟨*proof*⟩

lemma *monoidal-lifted*[*intro*]: **assumes** $monoidal\ opp$ **shows** $monoidal(lifted\ opp)$
⟨*proof*⟩

definition $support\ opp\ f\ s = \{x.\ x \in s \wedge f\ x \neq neutral\ opp\}$
definition $fold'\ opp\ e\ s \equiv (if\ finite\ s\ then\ fold\ opp\ e\ s\ else\ e)$
definition $iterate\ opp\ s\ f \equiv fold'\ (\lambda x\ a.\ opp\ (f\ x)\ a)\ (neutral\ opp)\ (support\ opp\ f\ s)$

lemma *support-subset*[*intro*]:$support\ opp\ f\ s \subseteq s$ ⟨*proof*⟩
lemma *support-empty*[*simp*]:$support\ opp\ f\ \{\} = \{\}$ ⟨*proof*⟩

lemma *fun-left-comm-monoidal*[*intro*]: **assumes** $monoidal\ opp$ **shows** $fun\text{-}left\text{-}comm\ opp$
⟨*proof*⟩

lemma *support-clauses*:
$\bigwedge f\ g\ s.\ support\ opp\ f\ \{\} = \{\}$
$\bigwedge f\ g\ s.\ support\ opp\ f\ (insert\ x\ s) = (if\ f(x) = neutral\ opp\ then\ support\ opp\ f\ s$

else insert x (*support opp* f s))
$\bigwedge f\ g\ s.$ *support opp* f $(s - \{x\}) =$ (*support opp* f s) $- \{x\}$
$\bigwedge f\ g\ s.$ *support opp* f $(s \cup t) =$ (*support opp* f s) $\cup$ (*support opp* f t)
$\bigwedge f\ g\ s.$ *support opp* f $(s \cap t) =$ (*support opp* f s) $\cap$ (*support opp* f t)
$\bigwedge f\ g\ s.$ *support opp* f $(s - t) =$ (*support opp* f s) $-$ (*support opp* f t)
$\bigwedge f\ g\ s.$ *support opp* g (f ' s) $= f$ ' (*support opp* (g *o* f) s)
⟨*proof*⟩

lemma *finite-support*[*intro*]:*finite* $s \Longrightarrow$ *finite* (*support opp* f s)
⟨*proof*⟩

lemma *iterate-empty*[*simp*]:*iterate opp* {} f = *neutral opp*
⟨*proof*⟩

lemma *iterate-insert*[*simp*]: **assumes** *monoidal opp finite* s
shows *iterate opp* (*insert* x s) f = (*if* $x \in s$ *then iterate opp* s f *else opp* (f x) (*iterate opp* s f))
⟨*proof*⟩

lemma *iterate-some*:
assumes *monoidal opp finite* s
shows *iterate* (*lifted opp*) s ($\lambda x.$ *Some*(f x)) = *Some* (*iterate opp* s f) ⟨*proof*⟩

23.20 Two key instances of additivity.

lemma *neutral-add*[*simp*]:
neutral op + = (*0*::-::*comm-monoid-add*) ⟨*proof*⟩

lemma *operative-content*[*intro*]: *operative* (*op* +) *content*
⟨*proof*⟩

lemma *neutral-monoid*: *neutral* ((*op* +)::('*a*::*comm-monoid-add*) ⇒ '*a* ⇒ '*a*) = *0*
⟨*proof*⟩

lemma *monoidal-monoid*[*intro*]:
shows *monoidal* ((*op* +)::('*a*::*comm-monoid-add*) ⇒ '*a* ⇒ '*a*)
⟨*proof*⟩

lemma *operative-integral*: **fixes** f::*real*^'*n* ⇒ '*a*::*banach*
shows *operative* (*lifted*(*op* +)) ($\lambda i.$ *if* f *integrable-on* i *then Some*(*integral* i f) *else None*)
⟨*proof*⟩

23.21 Points of division of a partition.

definition *division-points* (k::(*real*^'*n*) *set*) d =
{(j,x). (*interval-lowerbound* k)\$$j < x \wedge x <$ (*interval-upperbound* k)\$$j$ ∧
(∃ $i \in d$. (*interval-lowerbound* i)\$$j = x \vee$ (*interval-upperbound* i)\$$j = x$)}

lemma *division-points-finite*: **assumes** d *division-of* i

shows *finite* (*division-points* i d)
⟨*proof*⟩

lemma *division-points-subset*:
assumes d *division-of* $\{a..b\}$ $\forall i.\ a\$i < b\$i$ $a\$k < c$ $c < b\$k$
shows *division-points* $(\{a..b\} \cap \{x.\ x\$k \leq c\})$ $\{l \cap \{x.\ x\$k \leq c\} \mid l\ .\ l \in d \wedge$ $\sim(l \cap \{x.\ x\$k \leq c\} = \{\})\}$
$\subseteq$ *division-points* $(\{a..b\})$ d (**is** *?t1*) **and**
division-points $(\{a..b\} \cap \{x.\ x\$k \geq c\})$ $\{l \cap \{x.\ x\$k \geq c\} \mid l\ .\ l \in d \wedge \sim(l$ $\cap \{x.\ x\$k \geq c\} = \{\})\}$
$\subseteq$ *division-points* $(\{a..b\})$ d (**is** *?t2*)
⟨*proof*⟩

lemma *division-points-psubset*:
assumes d *division-of* $\{a..b\}$ $\forall i.\ a\$i < b\$i$ $a\$k < c$ $c < b\$k$
$l \in d$ *interval-lowerbound* $l\$k = c \vee$ *interval-upperbound* $l\$k = c$
shows *division-points* $(\{a..b\} \cap \{x.\ x\$k \leq c\})$ $\{l \cap \{x.\ x\$k \leq c\} \mid l.\ l{\in}d \wedge l \cap$ $\{x.\ x\$k \leq c\} \neq \{\}\} \subset$ *division-points* $(\{a..b\})$ d (**is** *?D1* $\subset$ *?D*)
division-points $(\{a..b\} \cap \{x.\ x\$k \geq c\})$ $\{l \cap \{x.\ x\$k \geq c\} \mid l.\ l{\in}d \wedge l \cap$ $\{x.\ x\$k \geq c\} \neq \{\}\} \subset$ *division-points* $(\{a..b\})$ d (**is** *?D2* $\subset$ *?D*)
⟨*proof*⟩

23.22 Preservation by divisions and tagged divisions.

lemma *support-support*[*simp*]:*support opp f* (*support opp f s*) = *support opp f s*
⟨*proof*⟩

lemma *iterate-support*[*simp*]: *iterate opp* (*support opp f s*) *f* = *iterate opp s f*
⟨*proof*⟩

lemma *iterate-expand-cases*:
iterate opp s f = (*if finite*(*support opp f s*) *then iterate opp* (*support opp f s*) *f else neutral opp*)
⟨*proof*⟩

lemma *iterate-image*: **assumes** *monoidal opp* *inj-on f s*
shows *iterate opp* (f ‘ s) g = *iterate opp s* $(g \circ f)$
⟨*proof*⟩

lemma *iterate-nonzero-image-lemma*:
assumes *monoidal opp finite s* $g(a)$ = *neutral opp*
$\forall x{\in}s.\ \forall y{\in}s.\ f\,x = f\,y \wedge x \neq y \longrightarrow g(f\,x)$ = *neutral opp*
shows *iterate opp* $\{f\,x \mid x.\ x \in s \wedge f\,x \neq a\}$ g = *iterate opp s* $(g \circ f)$
⟨*proof*⟩

lemma *iterate-eq-neutral*:
assumes *monoidal opp* $\forall x \in s.\ (f(x)$ = *neutral opp*)

shows $(iterate\ opp\ s\ f = neutral\ opp)$
⟨*proof*⟩

lemma *iterate-op*: **assumes** *monoidal opp finite s*
shows $iterate\ opp\ s\ (\lambda x.\ opp\ (f\ x)\ (g\ x)) = opp\ (iterate\ opp\ s\ f)\ (iterate\ opp\ s\ g)$ ⟨*proof*⟩

lemma *iterate-eq*: **assumes** *monoidal opp* $\bigwedge x.\ x \in s \Longrightarrow f\ x = g\ x$
shows $iterate\ opp\ s\ f = iterate\ opp\ s\ g$
⟨*proof*⟩

lemma *nonempty-witness*: **assumes** $s \neq \{\}$ **obtains** x **where** $x \in s$ ⟨*proof*⟩

lemma *operative-division*: **fixes** f::$(real\hat{\ }'n)\ set \Rightarrow 'a$
assumes *monoidal opp operative opp f d division-of* $\{a..b\}$
shows $iterate\ opp\ d\ f = f\ \{a..b\}$
⟨*proof*⟩

lemma *iterate-image-nonzero*: **assumes** *monoidal opp*
finite s $\forall x{\in}s.\ \forall y{\in}s.\ {\sim}(x = y) \wedge f\ x = f\ y \longrightarrow g(f\ x) = neutral\ opp$
shows $iterate\ opp\ (f\ \text{`}\ s)\ g = iterate\ opp\ s\ (g \circ f)$ ⟨*proof*⟩

lemma *operative-tagged-division*: **assumes** *monoidal opp operative opp f d tagged-division-of* $\{a..b\}$
shows $iterate(opp)\ d\ (\lambda(x,l).\ f\ l) = f\ \{a..b\}$
⟨*proof*⟩

23.23 Additivity of content.

lemma *setsum-iterate*:**assumes** *finite s* **shows** $setsum\ f\ s = iterate\ op + s\ f$
⟨*proof*⟩

lemma *additive-content-division*: **assumes** *d division-of* $\{a..b\}$
shows $setsum\ content\ d = content(\{a..b\})$
⟨*proof*⟩

lemma *additive-content-tagged-division*:
assumes *d tagged-division-of* $\{a..b\}$
shows $setsum\ (\lambda(x,l).\ content\ l)\ d = content(\{a..b\})$
⟨*proof*⟩

23.24 Finally, the integral of a constant

lemma *has-integral-const*[*intro*]:
$((\lambda x.\ c)\ has\text{-}integral\ (content(\{a..b::real\hat{\ }'n\}) *_R c))\ (\{a..b\})$
⟨*proof*⟩

23.25 Bounds on the norm of Riemann sums and the integral itself.

lemma *dsum-bound*: **assumes** p *division-of* $\{a..b\}$ $norm(c) \le e$
shows $norm(setsum\ (\lambda l.\ content\ l *_R c)\ p) \le e * content(\{a..b\})$ (**is** $?l \le ?r$)
⟨*proof*⟩

lemma *rsum-bound*: **assumes** p *tagged-division-of* $\{a..b\}$ $\forall x\in\{a..b\}.\ norm(f\ x) \le e$
shows $norm(setsum\ (\lambda(x,k).\ content\ k *_R f\ x)\ p) \le e * content(\{a..b\})$
⟨*proof*⟩

lemma *rsum-diff-bound*:
assumes p *tagged-division-of* $\{a..b\}$ $\forall x\in\{a..b\}.\ norm(f\ x - g\ x) \le e$
shows $norm(setsum\ (\lambda(x,k).\ content\ k *_R f\ x)\ p - setsum\ (\lambda(x,k).\ content\ k *_R g\ x)\ p) \le e * content(\{a..b\})$
⟨*proof*⟩

lemma *has-integral-bound*: **fixes** $f::real\hat{}'n \Rightarrow 'a::real\text{-}normed\text{-}vector$
assumes $0 \le B$ (f *has-integral* i) ($\{a..b\}$) $\forall x\in\{a..b\}.\ norm(f\ x) \le B$
shows $norm\ i \le B * content\ \{a..b\}$
⟨*proof*⟩

23.26 Similar theorems about relationship among components.

lemma *rsum-component-le*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}'m$
assumes p *tagged-division-of* $\{a..b\}$ $\forall x\in\{a..b\}.\ (f\ x)\$i \le (g\ x)\$i$
shows $(setsum\ (\lambda(x,k).\ content\ k *_R f\ x)\ p)\$i \le (setsum\ (\lambda(x,k).\ content\ k *_R g\ x)\ p)\$i$
⟨*proof*⟩

lemma *has-integral-component-le*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}'m$
assumes (f *has-integral* i) s (g *has-integral* j) s $\forall x\in s.\ (f\ x)\$k \le (g\ x)\$k$
shows $i\$k \le j\$k$
⟨*proof*⟩

lemma *integral-component-le*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}'m$
assumes f *integrable-on* s g *integrable-on* s $\forall x\in s.\ (f\ x)\$k \le (g\ x)\$k$
shows $(integral\ s\ f)\$k \le (integral\ s\ g)\$k$
⟨*proof*⟩

lemma *has-integral-dest-vec1-le*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}1$
assumes (f *has-integral* i) s (g *has-integral* j) s $\forall x\in s.\ f\ x \le g\ x$
shows $dest\text{-}vec1\ i \le dest\text{-}vec1\ j$ ⟨*proof*⟩

lemma *integral-dest-vec1-le*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}1$
assumes f *integrable-on* s g *integrable-on* s $\forall x\in s.\ f\ x \le g\ x$
shows $dest\text{-}vec1(integral\ s\ f) \le dest\text{-}vec1(integral\ s\ g)$
⟨*proof*⟩

lemma *has-integral-component-nonneg*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}'m$
assumes $(f\ has\text{-}integral\ i)\ s\ \forall x\in s.\ 0 \leq (f\ x)\$k$ **shows** $0 \leq i\$k$
⟨*proof*⟩

lemma *integral-component-nonneg*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}'m$
assumes $f\ integrable\text{-}on\ s\ \forall x\in s.\ 0 \leq (f\ x)\$k$ **shows** $0 \leq (integral\ s\ f)\$k$
⟨*proof*⟩

lemma *has-integral-dest-vec1-nonneg*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}1$
assumes $(f\ has\text{-}integral\ i)\ s\ \forall x\in s.\ 0 \leq f\ x$ **shows** $0 \leq i$
⟨*proof*⟩

lemma *integral-dest-vec1-nonneg*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}1$
assumes $f\ integrable\text{-}on\ s\ \forall x\in s.\ 0 \leq f\ x$ **shows** $0 \leq integral\ s\ f$
⟨*proof*⟩

lemma *has-integral-component-neg*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}'m$
assumes $(f\ has\text{-}integral\ i)\ s\ \forall x\in s.\ (f\ x)\$k \leq 0$ **shows** $i\$k \leq 0$
⟨*proof*⟩

lemma *has-integral-dest-vec1-neg*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}1$
assumes $(f\ has\text{-}integral\ i)\ s\ \forall x\in s.\ f\ x \leq 0$ **shows** $i \leq 0$
⟨*proof*⟩

lemma *has-integral-component-lbound*:
assumes $(f\ has\text{-}integral\ i)\ \{a..b\}\ \ \forall x\in\{a..b\}.\ B \leq f(x)\$k$ **shows** $B * content$ $\{a..b\} \leq i\$k$
⟨*proof*⟩

lemma *has-integral-component-ubound*:
assumes $(f\ has\text{-}integral\ i)\ \{a..b\}\ \forall x\in\{a..b\}.\ f\ x\$k \leq B$
shows $i\$k \leq B * content(\{a..b\})$
⟨*proof*⟩

lemma *integral-component-lbound*:
assumes $f\ integrable\text{-}on\ \{a..b\}\ \forall x\in\{a..b\}.\ B \leq f(x)\$k$
shows $B * content(\{a..b\}) \leq (integral(\{a..b\})\ f)\$k$
⟨*proof*⟩

lemma *integral-component-ubound*:
assumes $f\ integrable\text{-}on\ \{a..b\}\ \forall x\in\{a..b\}.\ f(x)\$k \leq B$
shows $(integral(\{a..b\})\ f)\$k \leq B * content(\{a..b\})$
⟨*proof*⟩

23.27 Uniform limit of integrable functions is integrable.

lemma *real-arch-invD*:
$0 < (e::real) \Longrightarrow (\exists n::nat.\ n \neq 0 \wedge 0 < inverse\ (real\ n) \wedge inverse\ (real\ n) <$

e)
⟨*proof*⟩

lemma *integrable-uniform-limit*: **fixes** f::*real*ˆ′*n* ⇒ ′*a*::*banach*
assumes $\forall\, e>0.\ \exists\, g.\ (\forall\, x\in\{a..b\}.\ norm(f\,x - g\,x) \le e) \wedge g$ *integrable-on* $\{a..b\}$
shows f *integrable-on* $\{a..b\}$
⟨*proof*⟩

23.28 Negligible sets.

definition *indicator* $s \equiv (\lambda x.\ if\ x \in s\ then\ 1\ else\ (0::real))$

lemma *dest-vec1-indicator*:
indicator $s\ x = (if\ x \in s\ then\ 1\ else\ 0)$ ⟨*proof*⟩

lemma *indicator-pos-le*[*intro*]: $0 \le (indicator\ s\ x)$ ⟨*proof*⟩

lemma *indicator-le-1*[*intro*]: $(indicator\ s\ x) \le 1$ ⟨*proof*⟩

lemma *dest-vec1-indicator-abs-le-1*: $abs(indicator\ s\ x) \le 1$
⟨*proof*⟩

definition *negligible* (s::(*real*ˆ′*n*) *set*) ≡ (∀ *a b*. ((*indicator s*) *has-integral 0*) $\{a..b\}$)

lemma *indicator-simps*[*simp*]:$x\in s \Longrightarrow indicator\ s\ x = 1\ x\notin s \Longrightarrow indicator\ s\ x = 0$
⟨*proof*⟩

23.29 Negligibility of hyperplane.

lemma *vsum-nonzero-image-lemma*:
assumes *finite* s $g(a) = 0$
$\forall\, x\in s.\ \forall\, y\in s.\ f\,x = f\,y \wedge x \ne y \longrightarrow g(f\,x) = 0$
shows *setsum* g $\{f\,x\ |x.\ x \in s \wedge f\,x \ne a\}$ = *setsum* $(g\ o\ f)\ s$
⟨*proof*⟩

lemma *interval-doublesplit*: **shows** $\{a..b\} \cap \{x\ .\ abs(x\$k - c) \le (e::real)\}$ =
$\{(\chi\ i.\ if\ i = k\ then\ max\ (a\$k)\ (c - e)\ else\ a\$i)\ ..\ (\chi\ i.\ if\ i = k\ then\ min\ (b\$k)\ (c + e)\ else\ b\$i)\}$
⟨*proof*⟩

lemma *division-doublesplit*: **assumes** p *division-of* $\{a..b$::*real*ˆ′*n*$\}$
shows $\{l \cap \{x.\ abs(x\$k - c) \le e\}\ |l.\ l \in p \wedge l \cap \{x.\ abs(x\$k - c) \le e\} \ne \{\}\}$ *division-of* $(\{a..b\} \cap \{x.\ abs(x\$k - c) \le e\})$
⟨*proof*⟩

lemma *content-doublesplit*: **assumes** $0 < e$
obtains d **where** $0 < d$ $content(\{a..b\} \cap \{x.\ abs(x\$k - c) \le d\}) < e$
⟨*proof*⟩

lemma *negligible-standard-hyperplane*[*intro*]: *negligible* $\{x.\ x\$k = (c::real)\}$
⟨*proof*⟩

23.30 A technical lemma about "refinement" of division.

lemma *tagged-division-finer*: **fixes** $p::((real^{\wedge\prime}n) \times ((real^{\wedge\prime}n)\ set))\ set$
assumes *p tagged-division-of* $\{a..b\}$ *gauge d*
obtains *q* **where** *q tagged-division-of* $\{a..b\}$ *d fine q* $\forall (x,k) \in p.\ k \subseteq d(x) \longrightarrow (x,k) \in q$
⟨*proof*⟩

23.31 Hence the main theorem about negligible sets.

lemma *finite-product-dependent*: **assumes** *finite s* $\bigwedge x.\ x{\in}s \Longrightarrow$ *finite* $(t\ x)$
shows *finite* $\{(i, j) \mid i\ j.\ i \in s \wedge j \in t\ i\}$ ⟨*proof*⟩

lemma *sum-sum-product*: **assumes** *finite s* $\forall i{\in}s.$ *finite* $(t\ i)$
shows *setsum* $(\lambda i.\ setsum\ (x\ i)\ (t\ i)::real)\ s = setsum\ (\lambda(i,j).\ x\ i\ j)\ \{(i,j) \mid i\ j.\ i \in s \wedge j \in t\ i\}$ ⟨*proof*⟩

lemma *has-integral-negligible*: **fixes** $f::real^{\wedge\prime}n \Rightarrow {}'a::real\text{-}normed\text{-}vector$
assumes *negligible s* $\forall x{\in}(t - s).\ f\ x = 0$
shows $(f$ *has-integral* $0)\ t$
⟨*proof*⟩

lemma *has-integral-spike*: **fixes** $f::real^{\wedge\prime}n \Rightarrow {}'a::real\text{-}normed\text{-}vector$
assumes *negligible s* $(\forall x{\in}(t - s).\ g\ x = f\ x)$ $(f$ *has-integral* $y)\ t$
shows $(g$ *has-integral* $y)\ t$
⟨*proof*⟩

lemma *has-integral-spike-eq*:
assumes *negligible s* $\forall x{\in}(t - s).\ g\ x = f\ x$
shows $((f$ *has-integral* $y)\ t \longleftrightarrow (g$ *has-integral* $y)\ t)$
⟨*proof*⟩

lemma *integrable-spike*: **assumes** *negligible s* $\forall x{\in}(t - s).\ g\ x = f\ x$ *f integrable-on t*
shows *g integrable-on t*
⟨*proof*⟩

lemma *integral-spike*: **assumes** *negligible s* $\forall x{\in}(t - s).\ g\ x = f\ x$
shows *integral t f* = *integral t g*
⟨*proof*⟩

23.32 Some other trivialities about negligible sets.

lemma *negligible-subset*[*intro*]: **assumes** *negligible s* $t \subseteq s$ **shows** *negligible t*
⟨*proof*⟩

lemma *negligible-diff*[*intro?*]: **assumes** *negligible s* **shows** *negligible*$(s - t)$ ⟨*proof*⟩

lemma *negligible-inter*: **assumes** *negligible s* ∨ *negligible t* **shows** *negligible*($s \cap t$) ⟨*proof*⟩

lemma *negligible-union*: **assumes** *negligible s negligible t* **shows** *negligible* ($s \cup t$) ⟨*proof*⟩

lemma *negligible-union-eq*[*simp*]: *negligible* ($s \cup t$) ⟷ (*negligible s* ∧ *negligible t*)
⟨*proof*⟩

lemma *negligible-sing*[*intro*]: *negligible* {*a*::*real*^'*n*}
⟨*proof*⟩

lemma *negligible-insert*[*simp*]: *negligible*(*insert a s*) ⟷ *negligible s*
⟨*proof*⟩

lemma *negligible-empty*[*intro*]: *negligible* {} ⟨*proof*⟩

lemma *negligible-finite*[*intro*]: **assumes** *finite s* **shows** *negligible s*
⟨*proof*⟩

lemma *negligible-unions*[*intro*]: **assumes** *finite s* ∀ *t*∈*s*. *negligible t* **shows** *negligible*($\bigcup s$)
⟨*proof*⟩

lemma *negligible*: *negligible s* ⟷ (∀ *t*::(*real*^'*n*) *set*. (*indicator s has-integral 0*) *t*)
⟨*proof*⟩

23.33 Finite case of the spike theorem is quite commonly needed.

lemma *has-integral-spike-finite*: **assumes** *finite s* ∀ *x*∈*t*−*s*. *g x* = *f x*
(*f has-integral y*) *t* **shows** (*g has-integral y*) *t*
⟨*proof*⟩

lemma *has-integral-spike-finite-eq*: **assumes** *finite s* ∀ *x*∈*t*−*s*. *g x* = *f x*
shows ((*f has-integral y*) *t* ⟷ (*g has-integral y*) *t*)
⟨*proof*⟩

lemma *integrable-spike-finite*:
assumes *finite s* ∀ *x*∈*t*−*s*. *g x* = *f x f integrable-on t* **shows** *g integrable-on t*
⟨*proof*⟩

23.34 In particular, the boundary of an interval is negligible.

lemma *negligible-frontier-interval*: *negligible*({*a*..*b*} − {*a*<..<*b*})
⟨*proof*⟩

lemma *has-integral-spike-interior*:
assumes $\forall x \in \{a<..<b\}.\ g\ x = f\ x$ (f *has-integral* y) ($\{a..b\}$) **shows** (g *has-integral* y) ($\{a..b\}$)
⟨*proof*⟩

lemma *has-integral-spike-interior-eq*:
assumes $\forall x \in \{a<..<b\}.\ g\ x = f\ x$ **shows** ((f *has-integral* y) ($\{a..b\}$) $\longleftrightarrow$ (g *has-integral* y) ($\{a..b\}$))
⟨*proof*⟩

lemma *integrable-spike-interior*: **assumes** $\forall x \in \{a<..<b\}.\ g\ x = f\ x$ f *integrable-on* $\{a..b\}$ **shows** g *integrable-on* $\{a..b\}$
⟨*proof*⟩

23.35 Integrability of continuous functions.

lemma *neutral-and*[*simp*]: *neutral op* $\wedge$ = *True*
⟨*proof*⟩

lemma *monoidal-and*[*intro*]: *monoidal op* $\wedge$ ⟨*proof*⟩

lemma *iterate-and*[*simp*]: **assumes** *finite s* **shows** (*iterate op* $\wedge$) $s\ p \longleftrightarrow (\forall x \in s.\ p\ x)$ ⟨*proof*⟩

lemma *operative-division-and*: **assumes** *operative op* $\wedge$ P d *division-of* $\{a..b\}$
shows $(\forall i \in d.\ P\ i) \longleftrightarrow P\ \{a..b\}$
⟨*proof*⟩

lemma *operative-approximable*: **assumes** $0 \le e$ **fixes** f::*real*^'*n* $\Rightarrow$ '*a*::*banach*
shows *operative op* $\wedge$ $(\lambda i.\ \exists g.\ (\forall x \in i.\ norm\ (f\ x - g\ (x{::}real\hat{}'n)) \le e) \wedge g$ *integrable-on* i) ⟨*proof*⟩

lemma *approximable-on-division*: **fixes** f::*real*^'*n* $\Rightarrow$ '*a*::*banach*
assumes $0 \le e$ d *division-of* $\{a..b\}$ $\forall i \in d.\ \exists g.\ (\forall x \in i.\ norm\ (f\ x - g\ x) \le e) \wedge g$ *integrable-on* i
obtains g **where** $\forall x \in \{a..b\}.\ norm\ (f\ x - g\ x) \le e$ g *integrable-on* $\{a..b\}$
⟨*proof*⟩

lemma *integrable-continuous*: **fixes** f::*real*^'*n* $\Rightarrow$ '*a*::*banach*
assumes *continuous-on* $\{a..b\}$ f **shows** f *integrable-on* $\{a..b\}$
⟨*proof*⟩

23.36 Specialization of additivity to one dimension.

lemma *operative-1-lt*: **assumes** *monoidal opp*
shows *operative opp* $f \longleftrightarrow ((\forall a\ b.\ b \le a \longrightarrow f\ \{a..b{::}real\hat{}1\} = neutral\ opp) \wedge$
$(\forall a\ b\ c.\ a < c \wedge c < b \longrightarrow opp\ (f\{a..c\})(f\{c..b\}) = f\ \{a..b\}))$
⟨*proof*⟩

lemma *operative-1-le*: **assumes** *monoidal opp*
shows $operative\ opp\ f \longleftrightarrow ((\forall\, a\ b.\ b \le a \longrightarrow f\ \{a..b::real\hat{\ }1\} = neutral\ opp) \wedge$
$(\forall\, a\ b\ c.\ a \le c \wedge c \le b \longrightarrow opp\ (f\{a..c\})(f\{c..b\}) = f\ \{a..b\}))$
⟨*proof*⟩

23.37 Special case of additivity we need for the FCT.

lemma *interval-bound-sing*[*simp*]: $interval\text{-}upperbound\ \{a\} = a\ \ interval\text{-}lowerbound\ \{a\} = a$
⟨*proof*⟩

lemma *additive-tagged-division-1*: **fixes** $f::real\hat{\ }1 \Rightarrow {}'a::real\text{-}normed\text{-}vector$
assumes $dest\text{-}vec1\ a \le dest\text{-}vec1\ b\ \ p\ tagged\text{-}division\text{-}of\ \{a..b\}$
shows $setsum\ (\lambda(x,k).\ f(interval\text{-}upperbound\ k) - f(interval\text{-}lowerbound\ k))\ p = f\ b - f\ a$
⟨*proof*⟩

23.38 A useful lemma allowing us to factor out the content size.

lemma *has-integral-factor-content*:
$(f\ has\text{-}integral\ i)\ \{a..b\} \longleftrightarrow (\forall\, e>0.\ \exists\, d.\ gauge\ d \wedge (\forall\, p.\ p\ tagged\text{-}division\text{-}of\ \{a..b\} \wedge d\ fine\ p$
$\longrightarrow norm\ (setsum\ (\lambda(x,k).\ content\ k *_R f\ x)\ p - i) \le e * content\ \{a..b\}))$
⟨*proof*⟩

23.39 Fundamental theorem of calculus.

lemma *fundamental-theorem-of-calculus*: **fixes** $f::real\hat{\ }1 \Rightarrow {}'a::banach$
assumes $a \le b\ \ \forall\, x\in\{a..b\}.\ ((f\ o\ vec1)\ has\text{-}vector\text{-}derivative\ f'(vec1\ x))\ (at\ x\ within\ \{a..b\})$
shows $(f'\ has\text{-}integral\ (f(vec1\ b) - f(vec1\ a)))\ (\{vec1\ a..vec1\ b\})$
⟨*proof*⟩

23.40 Attempt a systematic general set of "offset" results for components.

lemma *gauge-modify*:
assumes $(\forall\, s.\ open\ s \longrightarrow open\ \{x.\ f(x) \in s\})\ gauge\ d$
shows $gauge\ (\lambda x\ y.\ d\ (f\ x)\ (f\ y))$
⟨*proof*⟩

23.41 Only need trivial subintervals if the interval itself is trivial.

lemma *division-of-nontrivial*: **fixes** $s::(real\hat{\ }'n)\ set\ set$
assumes $s\ division\text{-}of\ \{a..b\}\ \ content(\{a..b\}) \ne 0$
shows $\{k.\ k \in s \wedge content\ k \ne 0\}\ division\text{-}of\ \{a..b\}$ ⟨*proof*⟩

23.42 Integrabibility on subintervals.

lemma *operative-integrable*: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::banach$ **shows**
operative op $\wedge$ ($\lambda i.$ *f integrable-on i*)
⟨*proof*⟩

lemma *integrable-subinterval*: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::banach$
assumes *f integrable-on* $\{a..b\}$ $\{c..d\} \subseteq \{a..b\}$ **shows** *f integrable-on* $\{c..d\}$
⟨*proof*⟩

23.43 Combining adjacent intervals in 1 dimension.

lemma *has-integral-combine*: **assumes** $(a::real\hat{\ }1) \leq c$ $c \leq b$
(*f has-integral i*) $\{a..c\}$ (*f has-integral* $(j::'a::banach)$) $\{c..b\}$
shows (*f has-integral* $(i + j)$) $\{a..b\}$
⟨*proof*⟩

lemma *integral-combine*: **fixes** $f::real\hat{\ }1 \Rightarrow 'a::banach$
assumes $a \leq c$ $c \leq b$ *f integrable-on* $(\{a..b\})$
shows *integral* $\{a..c\}$ f + *integral* $\{c..b\}$ f = *integral*$(\{a..b\})$ f
⟨*proof*⟩

lemma *integrable-combine*: **fixes** $f::real\hat{\ }1 \Rightarrow 'a::banach$
assumes $a \leq c$ $c \leq b$ *f integrable-on* $\{a..c\}$ *f integrable-on* $\{c..b\}$
shows *f integrable-on* $\{a..b\}$ ⟨*proof*⟩

23.44 Reduce integrability to "local" integrability.

lemma *integrable-on-little-subintervals*: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::banach$
assumes $\forall x\in\{a..b\}.\ \exists d>0.\ \forall u\ v.\ x \in \{u..v\} \wedge \{u..v\} \subseteq ball\ x\ d \wedge \{u..v\} \subseteq \{a..b\} \longrightarrow$ *f integrable-on* $\{u..v\}$
shows *f integrable-on* $\{a..b\}$
⟨*proof*⟩

23.45 Second FCT or existence of antiderivative.

lemma *integrable-const*[*intro*]:($\lambda x.\ c$) *integrable-on* $\{a..b\}$
⟨*proof*⟩

lemma *integral-has-vector-derivative*: **fixes** $f::real \Rightarrow 'a::banach$
assumes *continuous-on* $\{a..b\}$ f $x \in \{a..b\}$
shows (($\lambda u.$ *integral* $\{vec\ a..vec\ u\}$ (*f o dest-vec1*)) *has-vector-derivative* $f(x)$) (*at x within* $\{a..b\}$)
⟨*proof*⟩

lemma *integral-has-vector-derivative'*: **fixes** $f::real\hat{\ }1 \Rightarrow 'a::banach$
assumes *continuous-on* $\{a..b\}$ f $x \in \{a..b\}$
shows (($\lambda u.$ (*integral* $\{a..vec\ u\}$ f)) *has-vector-derivative* $f\ x$) (*at* $(x\$1)$ *within* $\{a\$1..b\$1\}$)
⟨*proof*⟩

lemma *antiderivative-continuous*: **assumes** *continuous-on* $\{a..b::real\}$ *f*
obtains *g* **where** $\forall x \in \{a..b\}$. (*g has-vector-derivative* $(f(x)::-::banach)$) (*at x within* $\{a..b\}$)
⟨*proof*⟩

23.46 Combined fundamental theorem of calculus.

lemma *antiderivative-integral-continuous*: **fixes** $f::real \Rightarrow 'a::banach$ **assumes** *continuous-on* $\{a..b\}$ *f*
obtains *g* **where** $\forall u \in \{a..b\}. \forall v \in \{a..b\}. u \leq v \longrightarrow$ ((*f o dest-vec1*) *has-integral* $(g\ v - g\ u)$) $\{vec\ u..vec\ v\}$
⟨*proof*⟩

23.47 General "twiddling" for interval-to-interval function image.

lemma *has-integral-twiddle*:
assumes $0 < r$ $\forall x.\ h(g\ x) = x$ $\forall x.\ g(h\ x) = x$ $\forall x.$ *continuous* (*at x*) *g*
$\forall u\ v. \exists w\ z.\ g$ ‘ $\{u..v\} = \{w..z\}$
$\forall u\ v. \exists w\ z.\ h$ ‘ $\{u..v\} = \{w..z\}$
$\forall u\ v.$ *content*(*g* ‘ $\{u..v\}$) $= r *$ *content* $\{u..v\}$
(*f has-integral i*) $\{a..b\}$
shows $((\lambda x.\ f(g\ x))$ *has-integral* $(1 / r) *_R i)$ (*h* ‘ $\{a..b\}$)
⟨*proof*⟩

23.48 Special case of a basic affine transformation.

lemma *interval-image-affinity-interval*: **shows** $\exists u\ v.\ (\lambda x.\ m *_R (x::real\hat{\ }'n) + c)$ ‘ $\{a..b\} = \{u..v\}$
⟨*proof*⟩

lemmas *Cart-simps* = *Cart-nth.add Cart-nth.minus Cart-nth.zero Cart-nth.diff Cart-nth.scaleR real-scaleR-def Cart-lambda-beta*
Cart-eq vector-le-def vector-less-def

lemma *setprod-cong2*: **assumes** $\bigwedge x.\ x \in A \Longrightarrow f\ x = g\ x$ **shows** *setprod f A* = *setprod g A*
⟨*proof*⟩

lemma *content-image-affinity-interval*:
content$((\lambda x::real\hat{\ }'n.\ m *_R x + c)$ ‘ $\{a..b\}$) = (*abs m*) ^ *CARD*('*n*) $*$ *content* $\{a..b\}$ (**is** *?l* = *?r*)
⟨*proof*⟩

lemma *has-integral-affinity*: **assumes** (*f has-integral i*) $\{a..b::real\hat{\ }'n\}$ $m \neq 0$
shows $((\lambda x.\ f(m *_R x + c))$ *has-integral* $((1 / (abs(m)\ \hat{}\ CARD('n::finite))) *_R i))$ $((\lambda x.\ (1 / m) *_R x + -((1 / m) *_R c))$ ‘ $\{a..b\})$
⟨*proof*⟩

lemma *integrable-affinity*: **assumes** *f integrable-on* $\{a..b\}$ $m \neq 0$
shows $(\lambda x.\ f(m *_R x + c))$ *integrable-on* $((\lambda x.\ (1\ /\ m) *_R x + -((1/m) *_R c))\ `\ \{a..b\})$
⟨*proof*⟩

23.49 Special case of stretching coordinate axes separately.

lemma *image-stretch-interval*:
$(\lambda x.\ \chi\ k.\ m\ k * x\$k)\ `\ \{a..b::real\hat{}'n\} =$
$(if\ \{a..b\} = \{\}\ then\ \{\}\ else\ \{(\chi\ k.\ min\ (m(k) * a\$k)\ (m(k) * b\$k))\ ..\ (\chi\ k.\ max\ (m(k) * a\$k)\ (m(k) * b\$k))\})$ (**is** $?l = ?r$)
⟨*proof*⟩

lemma *interval-image-stretch-interval*: $\exists\, u\ v.\ (\lambda x.\ \chi\ k.\ m\ k * x\$k)\ `\ \{a..b::real\hat{}'n\} = \{u..v\}$
⟨*proof*⟩

lemma *content-image-stretch-interval*:
$content((\lambda x::real\hat{}'n.\ \chi\ k.\ m\ k * x\$k)\ `\ \{a..b\}) = abs(setprod\ m\ UNIV) * content(\{a..b\})$
⟨*proof*⟩

lemma *has-integral-stretch*: **assumes** $(f\ has\text{-}integral\ i)\ \{a..b\}$ $\forall\, k.\ {}^{\sim}(m\ k = 0)$
shows $((\lambda x.\ f(\chi\ k.\ m\ k * x\$k))\ has\text{-}integral$
$((1/(abs(setprod\ m\ UNIV))) *_R i))\ ((\lambda x.\ \chi\ k.\ 1/(m\ k) * x\$k)\ `\ \{a..b\})$
⟨*proof*⟩

lemma *integrable-stretch*:
assumes *f integrable-on* $\{a..b\}$ $\forall\, k.\ {}^{\sim}(m\ k = 0)$
shows $(\lambda x.\ f(\chi\ k.\ m\ k * x\$k))$ *integrable-on* $((\lambda x.\ \chi\ k.\ 1/(m\ k) * x\$k)\ `\ \{a..b\})$
⟨*proof*⟩

23.50 even more special cases.

lemma *uminus-interval-vector*[*simp*]:*uminus* $`\ \{a..b\} = \{-b\ ..\ -a::real\hat{}'n\}$
⟨*proof*⟩

lemma *has-integral-reflect-lemma*[*intro*]: **assumes** $(f\ has\text{-}integral\ i)\ \{a..b\}$
shows $((\lambda x.\ f(-x))\ has\text{-}integral\ i)\ \{-b\ ..\ -a\}$
⟨*proof*⟩

lemma *has-integral-reflect*[*simp*]: $((\lambda x.\ f(-x))\ has\text{-}integral\ i)\ \{-b..-a\} \longleftrightarrow (f\ has\text{-}integral\ i)\ (\{a..b\})$
⟨*proof*⟩

lemma *integrable-reflect*[*simp*]: $(\lambda x.\ f(-x))$ *integrable-on* $\{-b..-a\} \longleftrightarrow f$ *integrable-on* $\{a..b\}$
⟨*proof*⟩

lemma *integral-reflect*[*simp*]: $integral\ \{-b..-a\}\ (\lambda x.\ f(-x)) = integral\ (\{a..b\})\ f$

⟨*proof*⟩

23.51 Stronger form of FCT; quite a tedious proof.

declare *norm-triangle-ineq4* [*intro*]

lemma *bgauge-existence-lemma*: $(\forall x \in s.\ \exists d::real.\ 0 < d \wedge q\ d\ x) \longleftrightarrow (\forall x.\ \exists d>0.\ x \in s \longrightarrow q\ d\ x)$ ⟨*proof*⟩

lemma *additive-tagged-division-1'*: **fixes** $f::real \Rightarrow 'a::real\text{-}normed\text{-}vector$
assumes $a \leq b$ p *tagged-division-of* $\{vec1\ a..vec1\ b\}$
shows *setsum* $(\lambda(x,k).\ f\ (dest\text{-}vec1\ (interval\text{-}upperbound\ k)) - f(dest\text{-}vec1\ (interval\text{-}lowerbound\ k)))\ p = f\ b - f\ a$
⟨*proof*⟩

lemma *split-minus*[*simp*]:$(\lambda(x,\ k).\ f\ x\ k)\ x - (\lambda(x,\ k).\ g\ x\ k)\ x = (\lambda(x,\ k).\ f\ x\ k - g\ x\ k)\ x$
⟨*proof*⟩

lemma *norm-triangle-le-sub*: $norm\ x + norm\ y \leq e \Longrightarrow norm\ (x - y) \leq e$
⟨*proof*⟩

lemma *fundamental-theorem-of-calculus-interior*:
assumes $a \leq b$ *continuous-on* $\{a..b\}\ f\ \forall x \in \{a<..<b\}.\ (f\ has\text{-}vector\text{-}derivative\ f'(x))\ (at\ x)$
shows $((f'\ o\ dest\text{-}vec1)\ has\text{-}integral\ (f\ b - f\ a))\ \{vec\ a..vec\ b\}$
⟨*proof*⟩

23.52 Stronger form with finite number of exceptional points.

lemma *fundamental-theorem-of-calculus-interior-strong*: **fixes** $f::real \Rightarrow 'a::banach$
assumes *finite* s $a \leq b$ *continuous-on* $\{a..b\}\ f$
$\forall x \in \{a<..<b\} - s.\ (f\ has\text{-}vector\text{-}derivative\ f'(x))\ (at\ x)$
shows $((f'\ o\ dest\text{-}vec1)\ has\text{-}integral\ (f\ b - f\ a))\ \{vec\ a..vec\ b\}$ ⟨*proof*⟩

lemma *fundamental-theorem-of-calculus-strong*: **fixes** $f::real \Rightarrow 'a::banach$
assumes *finite* s $a \leq b$ *continuous-on* $\{a..b\}\ f$
$\forall x \in \{a..b\} - s.\ (f\ has\text{-}vector\text{-}derivative\ f'(x))\ (at\ x)$
shows $((f'\ o\ dest\text{-}vec1)\ has\text{-}integral\ (f(b) - f(a)))\ \{vec1\ a..vec1\ b\}$
⟨*proof*⟩

lemma *indefinite-integral-continuous-left*: **fixes** $f::real\hat{~}1 \Rightarrow 'a::banach$
assumes f *integrable-on* $\{a..b\}$ $a < c$ $c \leq b$ $0 < e$
obtains d **where** $0 < d$ $\forall t.\ c\$1 - d < t\$1 \wedge t \leq c \longrightarrow norm(integral\ \{a..c\}\ f - integral\ \{a..t\}\ f) < e$
⟨*proof*⟩

lemma *indefinite-integral-continuous-right*: **fixes** $f::real\hat{~}1 \Rightarrow 'a::banach$
assumes f *integrable-on* $\{a..b\}$ $a \leq c$ $c < b$ $0 < e$

obtains *d* **where** $0 < d$ $\forall t.\ c \le t \wedge t\$1 < c\$1 + d \longrightarrow norm(integral\{a..c\}\ f - integral\{a..t\}\ f) < e$
⟨*proof*⟩

declare *dest-vec1-eq*[*simp del*] *not-less*[*simp*] *not-le*[*simp*]

lemma *indefinite-integral-continuous*: **fixes** $f::real\hat{}1 \Rightarrow 'a::banach$
assumes *f integrable-on* $\{a..b\}$ **shows** *continuous-on* $\{a..b\}$ $(\lambda x.\ integral\ \{a..x\}\ f)$
⟨*proof*⟩

23.53 This doesn't directly involve integration, but that gives an easy proof.

lemma *has-derivative-zero-unique-strong-interval*: **fixes** $f::real \Rightarrow 'a::banach$
assumes *finite k continuous-on* $\{a..b\}$ f $f\ a = y$
$\forall x\in(\{a..b\} - k).\ (f\ has\text{-}derivative\ (\lambda h.\ 0))\ (at\ x\ within\ \{a..b\})$ $x \in \{a..b\}$
shows $f\ x = y$
⟨*proof*⟩

23.54 Generalize a bit to any convex set.

lemmas *scaleR-simps* = *scaleR-zero-left scaleR-minus-left scaleR-left-diff-distrib scaleR-zero-right scaleR-minus-right scaleR-right-diff-distrib scaleR-eq-0-iff scaleR-cancel-left scaleR-cancel-right scaleR.add-right scaleR.add-left real-vector-class.scaleR-one*

lemma *has-derivative-zero-unique-strong-convex*: **fixes** $f::real\hat{}'n \Rightarrow 'a::banach$
assumes *convex s finite k continuous-on s f* $c \in s$ $f\ c = y$
$\forall x\in(s - k).\ (f\ has\text{-}derivative\ (\lambda h.\ 0))\ (at\ x\ within\ s)$ $x \in s$
shows $f\ x = y$
⟨*proof*⟩

23.55 Also to any open connected set with finite set of exceptions. Could generalize to locally convex set with limpt-free set of exceptions.

lemma *has-derivative-zero-unique-strong-connected*: **fixes** $f::real\hat{}'n \Rightarrow 'a::banach$
assumes *connected s open s finite k continuous-on s f* $c \in s$ $f\ c = y$
$\forall x\in(s - k).\ (f\ has\text{-}derivative\ (\lambda h.\ 0))\ (at\ x\ within\ s)$ $x\in s$
shows $f\ x = y$
⟨*proof*⟩

23.56 Integrating characteristic function of an interval.

lemma *has-integral-restrict-open-subinterval*: **fixes** $f::real\hat{}'n \Rightarrow 'a::banach$
assumes $(f\ has\text{-}integral\ i)\ \{c..d\}$ $\{c..d\} \subseteq \{a..b\}$
shows $((\lambda x.\ if\ x \in \{c<..<d\}\ then\ f\ x\ else\ 0)\ has\text{-}integral\ i)\ \{a..b\}$
⟨*proof*⟩

lemma *has-integral-restrict-closed-subinterval*: **fixes** $f::real\hat{}'n \Rightarrow 'a::banach$
assumes $(f\ has\text{-}integral\ i)\ (\{c..d\})\ \{c..d\} \subseteq \{a..b\}$
shows $((\lambda x.\ if\ x \in \{c..d\}\ then\ f\ x\ else\ 0)\ has\text{-}integral\ i)\ \{a..b\}$
⟨*proof*⟩

lemma *has-integral-restrict-closed-subintervals-eq*: **fixes** $f::real\hat{}'n \Rightarrow 'a::banach$
assumes $\{c..d\} \subseteq \{a..b\}$
shows $((\lambda x.\ if\ x \in \{c..d\}\ then\ f\ x\ else\ 0)\ has\text{-}integral\ i)\ \{a..b\} \longleftrightarrow (f\ has\text{-}integral\ i)\ \{c..d\}$ (**is** $?l = ?r$)
⟨*proof*⟩

23.57 Hence we can apply the limit process uniformly to all integrals.

lemma *has-integral'*: **fixes** $f::real\hat{}'n \Rightarrow 'a::banach$ **shows**
$(f\ has\text{-}integral\ i)\ s \longleftrightarrow (\forall\, e>0.\ \exists\, B>0.\ \forall\, a\ b.\ ball\ 0\ B \subseteq \{a..b\}$
$\longrightarrow (\exists\, z.\ ((\lambda x.\ if\ x \in s\ then\ f(x)\ else\ 0)\ has\text{-}integral\ z)\ \{a..b\} \wedge norm(z - i) < e))$ (**is** $?l \longleftrightarrow (\forall\, e>0.\ ?r\ e)$)
⟨*proof*⟩

lemma *has-integral-trans*[*simp*]: **fixes** $f::real\hat{}'n \Rightarrow real$ **shows**
$((\lambda x.\ vec1\ (f\ x))\ has\text{-}integral\ vec1\ i)\ s \longleftrightarrow (f\ has\text{-}integral\ i)\ s$
⟨*proof*⟩

lemma *integral-trans*[*simp*]: **assumes** $(f::real\hat{}'n \Rightarrow real)\ integrable\text{-}on\ s$
shows $integral\ s\ (\lambda x.\ vec1\ (f\ x)) = vec1\ (integral\ s\ f)$
⟨*proof*⟩

lemma *integrable-on-trans*[*simp*]: **fixes** $f::real\hat{}'n \Rightarrow real$ **shows**
$(\lambda x.\ vec1\ (f\ x))\ integrable\text{-}on\ s \longleftrightarrow (f\ integrable\text{-}on\ s)$
⟨*proof*⟩

lemma *has-integral-le*: **fixes** $f::real\hat{}'n \Rightarrow real$
assumes $(f\ has\text{-}integral\ i)\ s\ (g\ has\text{-}integral\ j)\ s\ \ \forall\, x{\in}s.\ (f\ x) \le (g\ x)$
shows $i \le j$ ⟨*proof*⟩

lemma *integral-le*: **fixes** $f::real\hat{}'n \Rightarrow real$
assumes $f\ integrable\text{-}on\ s\ g\ integrable\text{-}on\ s\ \forall\, x{\in}s.\ f\ x \le g\ x$
shows $integral\ s\ f \le integral\ s\ g$
⟨*proof*⟩

lemma *has-integral-nonneg*: **fixes** $f::real\hat{}'n \Rightarrow real$
assumes $(f\ has\text{-}integral\ i)\ s\ \forall\, x{\in}s.\ 0 \le f\ x$ **shows** $0 \le i$
⟨*proof*⟩

lemma *integral-nonneg*: **fixes** $f::real\hat{}'n \Rightarrow real$
assumes $f\ integrable\text{-}on\ s\ \forall\, x{\in}s.\ 0 \le f\ x$ **shows** $0 \le integral\ s\ f$
⟨*proof*⟩

23.58 Hence a general restriction property.

lemma *has-integral-restrict*[*simp*]: **assumes** $s \subseteq t$ **shows**
$((\lambda x.\ \mathit{if}\ x \in s\ \mathit{then}\ f\ x\ \mathit{else}\ (0::'a::banach))\ \mathit{has\text{-}integral}\ i)\ t \longleftrightarrow (f\ \mathit{has\text{-}integral}\ i)\ s$
⟨*proof*⟩

lemma *has-integral-restrict-univ*: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::banach$ **shows**
$((\lambda x.\ \mathit{if}\ x \in s\ \mathit{then}\ f\ x\ \mathit{else}\ 0)\ \mathit{has\text{-}integral}\ i)\ UNIV \longleftrightarrow (f\ \mathit{has\text{-}integral}\ i)\ s$ ⟨*proof*⟩

lemma *has-integral-on-superset*: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::banach$
assumes $\forall x.\ {}^\sim(x \in s) \longrightarrow f\ x = 0$ $s \subseteq t$ $(f\ \mathit{has\text{-}integral}\ i)\ s$
shows $(f\ \mathit{has\text{-}integral}\ i)\ t$
⟨*proof*⟩

lemma *integrable-on-superset*: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::banach$
assumes $\forall x.\ {}^\sim(x \in s) \longrightarrow f\ x = 0$ $s \subseteq t$ $f\ \mathit{integrable\text{-}on}\ s$
shows $f\ \mathit{integrable\text{-}on}\ t$
⟨*proof*⟩

lemma *integral-restrict-univ*[*intro*]: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::banach$
shows $f\ \mathit{integrable\text{-}on}\ s \Longrightarrow \mathit{integral}\ UNIV\ (\lambda x.\ \mathit{if}\ x \in s\ \mathit{then}\ f\ x\ \mathit{else}\ 0) = \mathit{integral}\ s\ f$
⟨*proof*⟩

lemma *integrable-restrict-univ*: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::banach$ **shows**
$(\lambda x.\ \mathit{if}\ x \in s\ \mathit{then}\ f\ x\ \mathit{else}\ 0)\ \mathit{integrable\text{-}on}\ UNIV \longleftrightarrow f\ \mathit{integrable\text{-}on}\ s$
⟨*proof*⟩

lemma *negligible-on-intervals*: $\mathit{negligible}\ s \longleftrightarrow (\forall\ a\ b.\ \mathit{negligible}(s \cap \{a..b\}))$ (**is** $?l = ?r$)
⟨*proof*⟩

lemma *has-integral-spike-set-eq*: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::banach$
assumes $\mathit{negligible}((s - t) \cup (t - s))$ **shows** $((f\ \mathit{has\text{-}integral}\ y)\ s \longleftrightarrow (f\ \mathit{has\text{-}integral}\ y)\ t)$
⟨*proof*⟩

lemma *has-integral-spike-set*[*dest*]: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::banach$
assumes $\mathit{negligible}((s - t) \cup (t - s))$ $(f\ \mathit{has\text{-}integral}\ y)\ s$
shows $(f\ \mathit{has\text{-}integral}\ y)\ t$
⟨*proof*⟩

lemma *integrable-spike-set*[*dest*]: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::banach$
assumes $\mathit{negligible}((s - t) \cup (t - s))$ $f\ \mathit{integrable\text{-}on}\ s$
shows $f\ \mathit{integrable\text{-}on}\ t$ ⟨*proof*⟩

lemma *integrable-spike-set-eq*: **fixes** $f::real\hat{\ }'n \Rightarrow 'a::banach$
assumes $\mathit{negligible}((s - t) \cup (t - s))$
shows $(f\ \mathit{integrable\text{-}on}\ s \longleftrightarrow f\ \mathit{integrable\text{-}on}\ t)$

$\langle proof \rangle$

23.59 More lemmas that are useful later.

lemma *has-integral-subset-component-le*: **fixes** f::*real*ˆ′*n* ⇒ *real*ˆ′*m*
assumes $s \subseteq t$ (f *has-integral* i) s (f *has-integral* j) t $\forall\, x{\in}t.\ 0 \leq f(x)\$k$
shows $i\$k \leq j\$k$
$\langle proof \rangle$

lemma *has-integral-subset-le*: **fixes** f::*real*ˆ′*n* ⇒ *real*
assumes $s \subseteq t$ (f *has-integral* i) s (f *has-integral* j) t $\forall\, x{\in}t.\ 0 \leq f(x)$
shows $i \leq j$ $\langle proof \rangle$

lemma *integral-subset-component-le*: **fixes** f::*real*ˆ′*n* ⇒ *real*ˆ′*m*
assumes $s \subseteq t$ f *integrable-on* s f *integrable-on* t $\forall\, x \in t.\ 0 \leq f(x)\$k$
shows (*integral* s f)$\$k \leq$ (*integral* t f)$\$k$
$\langle proof \rangle$

lemma *integral-subset-le*: **fixes** f::*real*ˆ′*n* ⇒ *real*
assumes $s \subseteq t$ f *integrable-on* s f *integrable-on* t $\forall\, x \in t.\ 0 \leq f(x)$
shows (*integral* s f) $\leq$ (*integral* t f)
$\langle proof \rangle$

lemma *has-integral-alt*′: **fixes** f::*real*ˆ′*n* ⇒ ′*a*::*banach*
shows (f *has-integral* i) s $\longleftrightarrow$ ($\forall\, a\ b.$ ($\lambda x.$ *if* $x \in s$ *then* $f\ x$ *else* 0) *integrable-on* $\{a..b\}$) $\wedge$
($\forall\, e{>}0.\ \exists\, B{>}0.\ \forall\, a\ b.$ *ball* $0\ B \subseteq \{a..b\} \longrightarrow$ *norm*(*integral* $\{a..b\}$ ($\lambda x.$ *if* $x \in s$ *then* $f\ x$ *else* 0) $-$ i) $<$ e) (**is** *?l* = *?r*)
$\langle proof \rangle$

23.60 Continuity of the integral (for a 1-dimensional interval).

lemma *integrable-alt*: **fixes** f::*real*ˆ′*n* ⇒ ′*a*::*banach* **shows**
f *integrable-on* s $\longleftrightarrow$
($\forall\, a\ b.$ ($\lambda x.$ *if* $x \in s$ *then* $f\ x$ *else* 0) *integrable-on* $\{a..b\}$) $\wedge$
($\forall\, e{>}0.\ \exists\, B{>}0.\ \forall\, a\ b\ c\ d.$ *ball* $0\ B \subseteq \{a..b\} \wedge$ *ball* $0\ B \subseteq \{c..d\}$
$\longrightarrow$ *norm*(*integral* $\{a..b\}$ ($\lambda x.$ *if* $x \in s$ *then* $f\ x$ *else* 0) $-$
integral $\{c..d\}$ ($\lambda x.$ *if* $x \in s$ *then* $f\ x$ *else* 0)) $<$ e) (**is** *?l* = *?r*)
$\langle proof \rangle$

lemma *integrable-altD*: **fixes** f::*real*ˆ′*n* ⇒ ′*a*::*banach*
assumes f *integrable-on* s
shows $\bigwedge a\ b.$ ($\lambda x.$ *if* $x \in s$ *then* $f\ x$ *else* 0) *integrable-on* $\{a..b\}$
$\bigwedge e.\ e{>}0 \Longrightarrow \exists\, B{>}0.\ \forall\, a\ b\ c\ d.$ *ball* $0\ B \subseteq \{a..b\} \wedge$ *ball* $0\ B \subseteq \{c..d\}$
$\longrightarrow$ *norm*(*integral* $\{a..b\}$ ($\lambda x.$ *if* $x \in s$ *then* $f\ x$ *else* 0) $-$ *integral* $\{c..d\}$ ($\lambda x.$ *if* $x \in s$ *then* $f\ x$ *else* 0)) $<$ e
$\langle proof \rangle$

lemma *integrable-on-subinterval*: **fixes** $f::real\hat{}'n \Rightarrow 'a::banach$
assumes f *integrable-on* s $\{a..b\} \subseteq s$ **shows** f *integrable-on* $\{a..b\}$
$\langle proof\rangle$

23.61 A straddling criterion for integrability.

lemma *integrable-straddle-interval*: **fixes** $f::real\hat{}'n \Rightarrow real$
assumes $\forall e>0.\ \exists g\ \ h\ i\ j.\ (g\ has\text{-}integral\ i)\ (\{a..b\}) \wedge (h\ has\text{-}integral\ j)\ (\{a..b\})$
$\wedge$
$norm(i - j) < e \wedge (\forall x\in\{a..b\}.\ (g\ x) \leq (f\ x) \wedge (f\ x) \leq (h\ x))$
shows f *integrable-on* $\{a..b\}$
$\langle proof\rangle$

lemma *integrable-straddle*: **fixes** $f::real\hat{}'n \Rightarrow real$
assumes $\forall e>0.\ \exists g\ h\ i\ j.\ (g\ has\text{-}integral\ i)\ s \wedge (h\ has\text{-}integral\ j)\ s \wedge$
$norm(i - j) < e \wedge (\forall x\in s.\ (g\ x) \leq (f\ x) \wedge (f\ x) \leq (h\ x))$
shows f *integrable-on* s
$\langle proof\rangle$

23.62 Adding integrals over several sets.

lemma *has-integral-union*: **fixes** $f::real\hat{}'n \Rightarrow 'a::banach$
assumes $(f\ has\text{-}integral\ i)\ s\ (f\ has\text{-}integral\ j)\ t\ negligible(s \cap t)$
shows $(f\ has\text{-}integral\ (i + j))\ (s \cup t)$
$\langle proof\rangle$

lemma *has-integral-unions*: **fixes** $f::real\hat{}'n \Rightarrow 'a::banach$
assumes $finite\ t\ \forall s\in t.\ (f\ has\text{-}integral\ (i\ s))\ s\ \ \forall s\in t.\ \forall s'\in t.\ {}^\sim(s = s') \longrightarrow$
$negligible(s \cap s')$
shows $(f\ has\text{-}integral\ (setsum\ i\ t))\ (\bigcup t)$
$\langle proof\rangle$

23.63 In particular adding integrals over a division, maybe not of an interval.

lemma *has-integral-combine-division*: **fixes** $f::real\hat{}'n \Rightarrow 'a::banach$
assumes $d\ division\text{-}of\ s\ \forall k\in d.\ (f\ has\text{-}integral\ (i\ k))\ k$
shows $(f\ has\text{-}integral\ (setsum\ i\ d))\ s$
$\langle proof\rangle$

lemma *integral-combine-division-bottomup*: **fixes** $f::real\hat{}'n \Rightarrow 'a::banach$
assumes $d\ division\text{-}of\ s\ \forall k\in d.\ f\ integrable\text{-}on\ k$
shows $integral\ s\ f = setsum\ (\lambda i.\ integral\ i\ f)\ d$
$\langle proof\rangle$

lemma *has-integral-combine-division-topdown*: **fixes** $f::real\hat{}'n \Rightarrow 'a::banach$
assumes $f\ integrable\text{-}on\ s\ d\ division\text{-}of\ k\ k \subseteq s$
shows $(f\ has\text{-}integral\ (setsum\ (\lambda i.\ integral\ i\ f)\ d))\ k$
$\langle proof\rangle$

lemma *integral-combine-division-topdown*: **fixes** $f::real^\wedge{}'n \Rightarrow {}'a::banach$
assumes *f integrable-on s d division-of s*
shows *integral s f* = *setsum* (λi. *integral i f*) *d*
⟨*proof*⟩

lemma *integrable-combine-division*: **fixes** $f::real^\wedge{}'n \Rightarrow {}'a::banach$
assumes *d division-of s* $\forall i \in d$. *f integrable-on i*
shows *f integrable-on s*
⟨*proof*⟩

lemma *integrable-on-subdivision*: **fixes** $f::real^\wedge{}'n \Rightarrow {}'a::banach$
assumes *d division-of i f integrable-on s* $i \subseteq s$
shows *f integrable-on i*
⟨*proof*⟩

23.64 Also tagged divisions.

lemma *has-integral-combine-tagged-division*: **fixes** $f::real^\wedge{}'n \Rightarrow {}'a::banach$
assumes *p tagged-division-of s* $\forall (x,k) \in p$. (*f has-integral* (*i k*)) *k*
shows (*f has-integral* (*setsum* ($\lambda(x,k)$. *i k*) *p*)) *s*
⟨*proof*⟩

lemma *integral-combine-tagged-division-bottomup*: **fixes** $f::real^\wedge{}'n \Rightarrow {}'a::banach$
assumes *p tagged-division-of* $\{a..b\}$ $\forall (x,k) \in p$. *f integrable-on k*
shows *integral* $\{a..b\}$ *f* = *setsum* ($\lambda(x,k)$. *integral k f*) *p*
⟨*proof*⟩

lemma *has-integral-combine-tagged-division-topdown*: **fixes** $f::real^\wedge{}'n \Rightarrow {}'a::banach$
assumes *f integrable-on* $\{a..b\}$ *p tagged-division-of* $\{a..b\}$
shows (*f has-integral* (*setsum* ($\lambda(x,k)$. *integral k f*) *p*)) $\{a..b\}$
⟨*proof*⟩

lemma *integral-combine-tagged-division-topdown*: **fixes** $f::real^\wedge{}'n \Rightarrow {}'a::banach$
assumes *f integrable-on* $\{a..b\}$ *p tagged-division-of* $\{a..b\}$
shows *integral* $\{a..b\}$ *f* = *setsum* ($\lambda(x,k)$. *integral k f*) *p*
⟨*proof*⟩

23.65 Henstock's lemma.

lemma *henstock-lemma-part1*: **fixes** $f::real^\wedge{}'n \Rightarrow {}'a::banach$
assumes *f integrable-on* $\{a..b\}$ $0 < e$ *gauge d*
($\forall p$. *p tagged-division-of* $\{a..b\} \wedge$ *d fine p* $\longrightarrow$ *norm* (*setsum* ($\lambda(x,k)$. *content k* $*_R$ *f x*) *p* − *integral*($\{a..b\}$) *f*) < *e*)
and *p*:*p tagged-partial-division-of* $\{a..b\}$ *d fine p*
shows *norm*(*setsum* ($\lambda(x,k)$. *content k* $*_R$ *f x* − *integral k f*) *p*) $\leq e$ (**is** *?x* $\leq e$)
⟨*proof*⟩

lemma *henstock-lemma-part2*: **fixes** $f::real^\wedge{}'m \Rightarrow real^\wedge{}'n$
assumes *f integrable-on* $\{a..b\}$ $0 < e$ *gauge d*

$\forall p.\ p$ *tagged-division-of* $\{a..b\} \wedge d$ *fine* $p \longrightarrow$ *norm* (*setsum* $(\lambda(x,k).$ *content* k $*_R f\ x)\ p$ −
integral$(\{a..b\})\ f) < e$ p *tagged-partial-division-of* $\{a..b\}$ d *fine* p
shows *setsum* $(\lambda(x,k).$ *norm*(*content* $k *_R f\ x$ − *integral* $k\ f$)) $p \leq 2 *$ *real* $(CARD('n)) * e$
⟨*proof*⟩

lemma *henstock-lemma*: **fixes** f::*real*ˆ′*m* ⇒ *real*ˆ′*n*
assumes f *integrable-on* $\{a..b\}$ $e{>}0$
obtains d **where** *gauge* d
$\forall p.\ p$ *tagged-partial-division-of* $\{a..b\} \wedge d$ *fine* p
$\longrightarrow$ *setsum* $(\lambda(x,k).$ *norm*(*content* $k *_R f\ x$ − *integral* $k\ f$)) $p < e$
⟨*proof*⟩

23.66 monotone convergence (bounded interval first).

lemma *monotone-convergence-interval*: **fixes** f::*nat* ⇒ *real*ˆ′*n* ⇒ *real*ˆ*1*
assumes $\forall k.\ (f\ k)$ *integrable-on* $\{a..b\}$
$\forall k.\ \forall x{\in}\{a..b\}.$ *dest-vec1*$(f\ k\ x) \leq$ *dest-vec1*$(f\ (Suc\ k)\ x)$
$\forall x{\in}\{a..b\}.\ ((\lambda k.\ f\ k\ x)$ −−−> $g\ x)$ *sequentially*
bounded {*integral* $\{a..b\}\ (f\ k) \mid k\ .\ k \in UNIV$}
shows g *integrable-on* $\{a..b\} \wedge ((\lambda k.$ *integral* $(\{a..b\})\ (f\ k))$ −−−> *integral* $(\{a..b\})\ g)$ *sequentially*
⟨*proof*⟩

lemma *monotone-convergence-increasing*: **fixes** f::*nat* ⇒ *real*ˆ′*n* ⇒ *real*ˆ*1*
assumes $\forall k.\ (f\ k)$ *integrable-on* s $\forall k.\ \forall x{\in}s.\ (f\ k\ x)\$1 \leq (f\ (Suc\ k)\ x)\$1$
$\forall x{\in}s.\ ((\lambda k.\ f\ k\ x)$ −−−> $g\ x)$ *sequentially* *bounded* {*integral* $s\ (f\ k) \mid k.\ True$}
shows g *integrable-on* $s \wedge ((\lambda k.$ *integral* $s\ (f\ k))$ −−−> *integral* $s\ g)$ *sequentially*
⟨*proof*⟩

lemma *monotone-convergence-decreasing*: **fixes** f::*nat* ⇒ *real*ˆ′*n* ⇒ *real*ˆ*1*
assumes $\forall k.\ (f\ k)$ *integrable-on* s $\forall k.\ \forall x{\in}s.\ (f\ (Suc\ k)\ x)\$1 \leq (f\ k\ x)\$1$
$\forall x{\in}s.\ ((\lambda k.\ f\ k\ x)$ −−−> $g\ x)$ *sequentially* *bounded* {*integral* $s\ (f\ k) \mid k.\ True$}
shows g *integrable-on* $s \wedge ((\lambda k.$ *integral* $s\ (f\ k))$ −−−> *integral* $s\ g)$ *sequentially*
⟨*proof*⟩

lemma *monotone-convergence-increasing-real*: **fixes** f::*nat* ⇒ *real*ˆ′*n* ⇒ *real*
assumes $\forall k.\ (f\ k)$ *integrable-on* s $\forall k.\ \forall x{\in}s.\ (f\ (Suc\ k)\ x) \geq (f\ k\ x)$
$\forall x{\in}s.\ ((\lambda k.\ f\ k\ x)$ −−−> $g\ x)$ *sequentially* *bounded* {*integral* $s\ (f\ k) \mid k.\ True$}
shows g *integrable-on* $s \wedge ((\lambda k.$ *integral* $s\ (f\ k))$ −−−> *integral* $s\ g)$ *sequentially*
⟨*proof*⟩

lemma *monotone-convergence-decreasing-real*: **fixes** f::*nat* ⇒ *real*ˆ′*n* ⇒ *real*
assumes $\forall k.\ (f\ k)$ *integrable-on* s $\forall k.\ \forall x{\in}s.\ (f\ (Suc\ k)\ x) \leq (f\ k\ x)$
$\forall x{\in}s.\ ((\lambda k.\ f\ k\ x)$ −−−> $g\ x)$ *sequentially* *bounded* {*integral* $s\ (f\ k) \mid k.\ True$}
shows g *integrable-on* $s \wedge ((\lambda k.$ *integral* $s\ (f\ k))$ −−−> *integral* $s\ g)$ *sequentially*
⟨*proof*⟩

23.67 absolute integrability (this is the same as Lebesgue integrability).

definition *absolutely-integrable-on* (**infixr** *absolutely'-integrable'-on 46*) **where**
f absolutely-integrable-on s $\longleftrightarrow$ *f integrable-on s* $\wedge$ ($\lambda x.$ (*norm*(*f x*))) *integrable-on s*

lemma *absolutely-integrable-onI*[*intro?*]:
f integrable-on s $\Longrightarrow$ ($\lambda x.$ (*norm*(*f x*))) *integrable-on s* $\Longrightarrow$ *f absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-onD*[*dest*]: **assumes** *f absolutely-integrable-on s*
shows *f integrable-on s* ($\lambda x.$ (*norm*(*f x*))) *integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-on-trans*[*simp*]: **fixes** *f*::*real^'n* $\Rightarrow$ *real* **shows**
(*vec1 o f*) *absolutely-integrable-on s* $\longleftrightarrow$ *f absolutely-integrable-on s*
⟨*proof*⟩

lemma *integral-norm-bound-integral*: **fixes** *f*::*real^'n* $\Rightarrow$ *'a*::*banach*
assumes *f integrable-on s g integrable-on s* $\forall x\in s.$ *norm*(*f x*) $\leq$ *g x*
shows *norm*(*integral s f*) $\leq$ (*integral s g*)
⟨*proof*⟩

lemma *integral-norm-bound-integral-component*: **fixes** *f*::*real^'n* $\Rightarrow$ *'a*::*banach*
assumes *f integrable-on s g integrable-on s* $\forall x\in s.$ *norm*(*f x*) $\leq$ (*g x*)$*k*
shows *norm*(*integral s f*) $\leq$ (*integral s g*)$*k*
⟨*proof*⟩

lemma *has-integral-norm-bound-integral-component*: **fixes** *f*::*real^'n* $\Rightarrow$ *'a*::*banach*
assumes (*f has-integral i*) *s* (*g has-integral j*) *s* $\forall x\in s.$ *norm*(*f x*) $\leq$ (*g x*)$*k*
shows *norm*(*i*) $\leq$ *j*$*k* ⟨*proof*⟩

lemma *absolutely-integrable-le*: **fixes** *f*::*real^'n* $\Rightarrow$ *'a*::*banach*
assumes *f absolutely-integrable-on s*
shows *norm*(*integral s f*) $\leq$ *integral s* ($\lambda x.$ *norm*(*f x*))
⟨*proof*⟩

lemma *absolutely-integrable-0*[*intro*]: ($\lambda x.$ *0*) *absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-cmul*[*intro*]:
f absolutely-integrable-on s $\Longrightarrow$ ($\lambda x.$ $c *_R f\ x$) *absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-neg*[*intro*]:
f absolutely-integrable-on s $\Longrightarrow$ ($\lambda x.$ $-f(x)$) *absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-norm*[*intro*]:
f *absolutely-integrable-on* $s \Longrightarrow (\lambda x.\ norm(f\ x))$ *absolutely-integrable-on* s
⟨*proof*⟩

lemma *absolutely-integrable-abs*[*intro*]:
f *absolutely-integrable-on* $s \Longrightarrow (\lambda x.\ abs(f\ x{::}real))$ *absolutely-integrable-on* s
⟨*proof*⟩

lemma *absolutely-integrable-on-subinterval*: **fixes** $f{::}real\hat{\ }'n \Rightarrow 'a{::}banach$ **shows**
f *absolutely-integrable-on* $s \Longrightarrow \{a..b\} \subseteq s \Longrightarrow f$ *absolutely-integrable-on* $\{a..b\}$
⟨*proof*⟩

lemma *absolutely-integrable-bounded-variation*: **fixes** $f{::}real\hat{\ }'n \Rightarrow 'a{::}banach$
assumes f *absolutely-integrable-on* *UNIV*
obtains B **where** $\forall\, d.\ d$ *division-of* $(\bigcup d) \longrightarrow setsum\ (\lambda k.\ norm(integral\ k\ f))\ d \leq B$
⟨*proof*⟩

lemma *helplemma*:
assumes $setsum\ (\lambda x.\ norm(f\ x - g\ x))\ s < e$ *finite* s
shows $abs(setsum\ (\lambda x.\ norm(f\ x))\ s - setsum\ (\lambda x.\ norm(g\ x))\ s) < e$
⟨*proof*⟩

lemma *bounded-variation-absolutely-integrable-interval*:
fixes $f{::}real\hat{\ }'n \Rightarrow real\hat{\ }'m$ **assumes** f *integrable-on* $\{a..b\}$
$\forall\, d.\ d$ *division-of* $\{a..b\} \longrightarrow setsum\ (\lambda k.\ norm(integral\ k\ f))\ d \leq B$
shows f *absolutely-integrable-on* $\{a..b\}$
⟨*proof*⟩

lemma *bounded-variation-absolutely-integrable*: **fixes** $f{::}real\hat{\ }'n \Rightarrow real\hat{\ }'m$
assumes f *integrable-on* *UNIV* $\forall\, d.\ d$ *division-of* $(\bigcup d) \longrightarrow setsum\ (\lambda k.\ norm(integral\ k\ f))\ d \leq B$
shows f *absolutely-integrable-on* *UNIV*
⟨*proof*⟩

lemma *absolutely-integrable-restrict-univ*:
$(\lambda x.\ if\ x \in s\ then\ f\ x\ else\ (0{::}'a{::}banach))$ *absolutely-integrable-on* *UNIV* $\longleftrightarrow f$ *absolutely-integrable-on* s
⟨*proof*⟩

lemma *absolutely-integrable-add*[*intro*]: **fixes** $f\ g{::}real\hat{\ }'n \Rightarrow real\hat{\ }'m$
assumes f *absolutely-integrable-on* s g *absolutely-integrable-on* s
shows $(\lambda x.\ f(x) + g(x))$ *absolutely-integrable-on* s
⟨*proof*⟩

lemma *absolutely-integrable-sub*[*intro*]: **fixes** $f\ g{::}real\hat{\ }'n \Rightarrow real\hat{\ }'m$
assumes f *absolutely-integrable-on* s g *absolutely-integrable-on* s
shows $(\lambda x.\ f(x) - g(x))$ *absolutely-integrable-on* s
⟨*proof*⟩

lemma *absolutely-integrable-linear*: **fixes** $f::real\hat{}'m \Rightarrow real\hat{}'n$ **and** $h::real\hat{}'n \Rightarrow real\hat{}'p$
assumes *f absolutely-integrable-on s bounded-linear h*
shows (*h o f*) *absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-setsum*: **fixes** $f::'a \Rightarrow real\hat{}'n \Rightarrow real\hat{}'m$
assumes *finite t* $\bigwedge a.\ a \in t \Longrightarrow (f\ a)$ *absolutely-integrable-on s*
shows ($\lambda x.$ *setsum* ($\lambda a.$ *f a x*) *t*) *absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-vector-abs*:
assumes *f absolutely-integrable-on s*
shows ($\lambda x.$ (χ *i. abs*(*f x*\$*i*))$::real\hat{}'n$) *absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-max*: **fixes** $f::real\hat{}'m \Rightarrow real\hat{}'n$
assumes *f absolutely-integrable-on s g absolutely-integrable-on s*
shows ($\lambda x.$ (χ *i. max* (*f*(*x*)\$*i*) (*g*(*x*)\$*i*))$::real\hat{}'n$) *absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-max-real*: **fixes** $f::real\hat{}'m \Rightarrow real$
assumes *f absolutely-integrable-on s g absolutely-integrable-on s*
shows ($\lambda x.$ *max* (*f x*) (*g x*)) *absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-min*: **fixes** $f::real\hat{}'m \Rightarrow real\hat{}'n$
assumes *f absolutely-integrable-on s g absolutely-integrable-on s*
shows ($\lambda x.$ (χ *i. min* (*f*(*x*)\$*i*) (*g*(*x*)\$*i*))$::real\hat{}'n$) *absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-min-real*: **fixes** $f::real\hat{}'m \Rightarrow real$
assumes *f absolutely-integrable-on s g absolutely-integrable-on s*
shows ($\lambda x.$ *min* (*f x*) (*g x*)) *absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-abs-eq*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}'m$
shows *f absolutely-integrable-on s* $\longleftrightarrow$ *f integrable-on s* $\wedge$
($\lambda x.$ (χ *i. abs*(*f x*\$*i*))$::real\hat{}'m$) *integrable-on s* (**is** *?l* = *?r*)
⟨*proof*⟩

lemma *nonnegative-absolutely-integrable*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}'m$
assumes $\forall x \in s.\ \forall i.\ 0 \leq f(x)\$i$ *f integrable-on s*
shows *f absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-integrable-bound*: **fixes** $f::real\hat{}'n \Rightarrow real\hat{}'m$
assumes $\forall x \in s.\ norm(f\ x) \leq g\ x$ *f integrable-on s g integrable-on s*

shows *f absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-integrable-bound-real*: **fixes** f::*real*^'*n* ⇒ *real*
assumes $\forall x{\in}s.$ *norm*$(f\ x) \leq g\ x$ *f integrable-on s g integrable-on s*
shows *f absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-absolutely-integrable-bound*:
fixes f::*real*^'*n* ⇒ *real*^'*m* **and** g::*real*^'*n* ⇒ *real*^'*p*
assumes $\forall x{\in}s.$ *norm*$(f\ x) \leq$ *norm*$(g\ x)$ *f integrable-on s g absolutely-integrable-on s*
shows *f absolutely-integrable-on s*
⟨*proof*⟩

lemma *absolutely-integrable-inf-real*:
assumes *finite k* $k \neq \{\}$
$\forall i{\in}k.$ $(\lambda x.\ (fs\ x\ i)$::*real*) *absolutely-integrable-on s*
shows $(\lambda x.\ (Inf\ ((fs\ x)\ `\ k)))$ *absolutely-integrable-on s* ⟨*proof*⟩

lemma *absolutely-integrable-sup-real*:
assumes *finite k* $k \neq \{\}$
$\forall i{\in}k.$ $(\lambda x.\ (fs\ x\ i)$::*real*) *absolutely-integrable-on s*
shows $(\lambda x.\ (Sup\ ((fs\ x)\ `\ k)))$ *absolutely-integrable-on s* ⟨*proof*⟩

23.68 Dominated convergence.

lemma *dominated-convergence*: **fixes** f::*nat* ⇒ *real*^'*n* ⇒ *real*
assumes $\bigwedge k.$ $(f\ k)$ *integrable-on s h integrable-on s*
$\bigwedge k.\ \forall x \in s.$ *norm*$(f\ k\ x) \leq (h\ x)$
$\forall x \in s.\ ((\lambda k.\ f\ k\ x) \ {-}{-}{-}{>}\ g\ x)$ *sequentially*
shows *g integrable-on s* $((\lambda k.$ *integral s* $(f\ k))\ {-}{-}{-}{>}$ *integral s g*) *sequentially*
⟨*proof*⟩

declare [[*smt-certificates*=]]
declare [[*smt-fixed*=*false*]]

end

24 Real-Integration: Integration on real intervals

theory *Real-Integration*
imports *Integration*
begin

We follow John Harrison in formalizing the Gauge integral.

definition *Integral* :: *real set* ⇒ (*real* ⇒ *real*) ⇒ *real* ⇒ *bool* **where**
Integral s f k = (*f o dest-vec1 has-integral k*) (*vec1* ‘ *s*)

lemmas *integral-unfold* = *Integral-def split-conv o-def vec1-interval*

lemma *Integral-unique*:
[| *Integral*{*a..b*} *f k1*; *Integral*{*a..b*} *f k2* |] ==> *k1* = *k2*
⟨*proof*⟩

lemma *Integral-zero* [*simp*]: *Integral*{*a..a*} *f 0*
⟨*proof*⟩

lemma *Integral-eq-diff-bounds*: **assumes** $a \leq b$ **shows** *Integral*{*a..b*} (%*x*. *1*) (*b* − *a*)
⟨*proof*⟩

lemma *Integral-mult-const*: **assumes** $a \leq b$ **shows** *Integral*{*a..b*} (%*x*. *c*) (*c*∗(*b* − *a*))
⟨*proof*⟩

lemma *Integral-mult*: **assumes** *Integral*{*a..b*} *f k* **shows** *Integral*{*a..b*} (%*x*. *c* ∗ *f x*) (*c* ∗ *k*)
⟨*proof*⟩

lemma *Integral-add*:
assumes *Integral* {*a..b*} *f x1 Integral* {*b..c*} *f x2* $a \leq b$ **and** $b \leq c$
shows *Integral* {*a..c*} *f* (*x1* + *x2*)
⟨*proof*⟩

lemma *FTC1*: **assumes** $a \leq b$ $\forall x.\ a \leq x$ & $x \leq b$ −−> *DERIV f x* :> $f'(x)$
shows *Integral*{*a..b*} f' $(f(b) - f(a))$
⟨*proof*⟩

lemma *Integral-subst*: [| *Integral*{*a..b*} *f k1*; *k2*=*k1* |] ==> *Integral*{*a..b*} *f k2*
⟨*proof*⟩

24.1 Additivity Theorem of Gauge Integral

Bartle/Sherbert: Theorem 10.1.5 p. 278

lemma *Integral-add-fun*: [| *Integral*{*a..b*} *f k1*; *Integral*{*a..b*} *g k2* |] ==> *Integral*{*a..b*} (%*x*. *f x* + *g x*) (*k1* + *k2*)
⟨*proof*⟩

lemma *norm-vec1* ′[*simp*]:*norm* (*vec1 x*) = *norm x*
⟨*proof*⟩

lemma *Integral-le*: **assumes** $a \leq b$ $\forall x.\ a \leq x$ & $x \leq b$ −−> $f(x) \leq g(x)$ *Integral*{*a..b*} *f k1 Integral*{*a..b*} *g k2* **shows** $k1 \leq k2$
⟨*proof*⟩

lemma *monotonic-anti-derivative*:

fixes $f\ g$ :: $real$ => $real$ **shows**
$[|\ a \leq b;\ \forall c.\ a \leq c\ \&\ c \leq b\ {-}{-}{>}\ f'\ c \leq g'\ c;$
$\forall x.\ DERIV\ f\ x :> f'\ x;\ \forall x.\ DERIV\ g\ x :> g'\ x\ |]$
$==> f\ b - f\ a \leq g\ b - g\ a$
$\langle proof \rangle$

end

25 Path-Connected: Continuous paths and path-connected sets

theory *Path-Connected*
imports *Convex-Euclidean-Space*
begin

25.1 Paths.

definition
$path :: (real \Rightarrow 'a::topological\text{-}space) \Rightarrow bool$
where $path\ g \longleftrightarrow continuous\text{-}on\ \{0\ ..\ 1\}\ g$

definition
$pathstart :: (real \Rightarrow 'a::topological\text{-}space) \Rightarrow 'a$
where $pathstart\ g = g\ 0$

definition
$pathfinish :: (real \Rightarrow 'a::topological\text{-}space) \Rightarrow 'a$
where $pathfinish\ g = g\ 1$

definition
$path\text{-}image :: (real \Rightarrow 'a::topological\text{-}space) \Rightarrow 'a\ set$
where $path\text{-}image\ g = g\ `\ \{0\ ..\ 1\}$

definition
$reversepath :: (real \Rightarrow 'a::topological\text{-}space) \Rightarrow (real \Rightarrow 'a)$
where $reversepath\ g = (\lambda x.\ g(1 - x))$

definition
$joinpaths :: (real \Rightarrow 'a::topological\text{-}space) \Rightarrow (real \Rightarrow 'a) \Rightarrow (real \Rightarrow 'a)$
(**infixr** +++ *75*)
where $g1 +{+}+ g2 = (\lambda x.\ if\ x \leq 1/2\ then\ g1\ (2 * x)\ else\ g2\ (2 * x - 1))$

definition
$simple\text{-}path :: (real \Rightarrow 'a::topological\text{-}space) \Rightarrow bool$
where $simple\text{-}path\ g \longleftrightarrow$
$(\forall x{\in}\{0..1\}.\ \forall y{\in}\{0..1\}.\ g\ x = g\ y \longrightarrow x = y \lor x = 0 \land y = 1 \lor x = 1 \land y = 0)$

definition
injective-path :: $(real \Rightarrow {}'a::topological\text{-}space) \Rightarrow bool$
where *injective-path* $g \longleftrightarrow (\forall\, x{\in}\{0..1\}.\ \forall\, y{\in}\{0..1\}.\ g\ x = g\ y \longrightarrow x = y)$

25.2 Some lemmas about these concepts.

lemma *injective-imp-simple-path*:
injective-path $g \Longrightarrow$ *simple-path* g
⟨*proof*⟩

lemma *path-image-nonempty*: *path-image* $g \neq \{\}$
⟨*proof*⟩

lemma *pathstart-in-path-image*[*intro*]: (*pathstart* g) $\in$ *path-image* g
⟨*proof*⟩

lemma *pathfinish-in-path-image*[*intro*]: (*pathfinish* g) $\in$ *path-image* g
⟨*proof*⟩

lemma *connected-path-image*[*intro*]: *path* $g \Longrightarrow$ *connected*(*path-image* g)
⟨*proof*⟩

lemma *compact-path-image*[*intro*]: *path* $g \Longrightarrow$ *compact*(*path-image* g)
⟨*proof*⟩

lemma *reversepath-reversepath*[*simp*]: *reversepath*(*reversepath* g) = g
⟨*proof*⟩

lemma *pathstart-reversepath*[*simp*]: *pathstart*(*reversepath* g) = *pathfinish* g
⟨*proof*⟩

lemma *pathfinish-reversepath*[*simp*]: *pathfinish*(*reversepath* g) = *pathstart* g
⟨*proof*⟩

lemma *pathstart-join*[*simp*]: *pathstart*($g1$ +++ $g2$) = *pathstart* $g1$
⟨*proof*⟩

lemma *pathfinish-join*[*simp*]:*pathfinish*($g1$ +++ $g2$) = *pathfinish* $g2$
⟨*proof*⟩

lemma *path-image-reversepath*[*simp*]: *path-image*(*reversepath* g) = *path-image* g
⟨*proof*⟩

lemma *path-reversepath*[*simp*]: *path*(*reversepath* g) $\longleftrightarrow$ *path* g ⟨*proof*⟩

lemmas *reversepath-simps* = *path-reversepath path-image-reversepath pathstart-reversepath pathfinish-reversepath*

lemma *path-join*[*simp*]: **assumes** *pathfinish g1* = *pathstart g2* **shows** *path* (*g1* +++ *g2*) $\longleftrightarrow$ *path g1* $\wedge$ *path g2*
⟨*proof*⟩

lemma *path-image-join-subset*: *path-image*(*g1* +++ *g2*) $\subseteq$ (*path-image g1* $\cup$ *path-image g2*) ⟨*proof*⟩

lemma *subset-path-image-join*:
assumes *path-image g1* $\subseteq$ *s path-image g2* $\subseteq$ *s* **shows** *path-image*(*g1* +++ *g2*) $\subseteq$ *s*
⟨*proof*⟩

lemma *path-image-join*:
assumes *path g1 path g2 pathfinish g1* = *pathstart g2*
shows *path-image*(*g1* +++ *g2*) = (*path-image g1*) $\cup$ (*path-image g2*)
⟨*proof*⟩

lemma *not-in-path-image-join*:
assumes *x* $\notin$ *path-image g1 x* $\notin$ *path-image g2* **shows** *x* $\notin$ *path-image*(*g1* +++ *g2*)
⟨*proof*⟩

lemma *simple-path-reversepath*: **assumes** *simple-path g* **shows** *simple-path* (*reversepath g*)
⟨*proof*⟩

lemma *simple-path-join-loop*:
assumes *injective-path g1 injective-path g2 pathfinish g2* = *pathstart g1*
(*path-image g1* $\cap$ *path-image g2*) $\subseteq$ {*pathstart g1*,*pathstart g2*}
shows *simple-path*(*g1* +++ *g2*)
⟨*proof*⟩

lemma *injective-path-join*:
assumes *injective-path g1 injective-path g2 pathfinish g1* = *pathstart g2*
(*path-image g1* $\cap$ *path-image g2*) $\subseteq$ {*pathstart g2*}
shows *injective-path*(*g1* +++ *g2*)
⟨*proof*⟩

lemmas *join-paths-simps* = *path-join path-image-join pathstart-join pathfinish-join*

25.3 Reparametrizing a closed curve to start at some chosen point.

definition *shiftpath a* (*f*::*real* $\Rightarrow$ *'a*::*topological-space*) =
(λx. *if* (*a* + *x*) $\leq$ *1 then f*(*a* + *x*) *else f*(*a* + *x* − *1*))

lemma *pathstart-shiftpath*: *a* $\leq$ *1* $\Longrightarrow$ *pathstart*(*shiftpath a g*) = *g a*
⟨*proof*⟩

lemma *pathfinish-shiftpath*: **assumes** $0 \leq a$ *pathfinish* $g =$ *pathstart* g
shows *pathfinish*(*shiftpath* a g) $= g\ a$
⟨*proof*⟩

lemma *endpoints-shiftpath*:
assumes *pathfinish* $g =$ *pathstart* g $a \in \{0\ ..\ 1\}$
shows *pathfinish*(*shiftpath* a g) $= g\ a$ *pathstart*(*shiftpath* a g) $= g\ a$
⟨*proof*⟩

lemma *closed-shiftpath*:
assumes *pathfinish* $g =$ *pathstart* g $a \in \{0..1\}$
shows *pathfinish*(*shiftpath* a g) $=$ *pathstart*(*shiftpath* a g)
⟨*proof*⟩

lemma *path-shiftpath*:
assumes *path* g *pathfinish* $g =$ *pathstart* g $a \in \{0..1\}$
shows *path*(*shiftpath* a g) ⟨*proof*⟩

lemma *shiftpath-shiftpath*: **assumes** *pathfinish* $g =$ *pathstart* g $a \in \{0..1\}$ $x \in \{0..1\}$
shows *shiftpath* $(1 - a)$ (*shiftpath* a g) $x = g\ x$
⟨*proof*⟩

lemma *path-image-shiftpath*:
assumes $a \in \{0..1\}$ *pathfinish* $g =$ *pathstart* g
shows *path-image*(*shiftpath* a g) $=$ *path-image* g ⟨*proof*⟩

25.4 Special case of straight-line paths.

definition
linepath :: $'a$::*real-normed-vector* $\Rightarrow$ $'a \Rightarrow$ *real* $\Rightarrow$ $'a$ **where**
linepath $a\ b = (\lambda x.\ (1 - x) *_R a + x *_R b)$

lemma *pathstart-linepath*[*simp*]: *pathstart*(*linepath* a b) $= a$
⟨*proof*⟩

lemma *pathfinish-linepath*[*simp*]: *pathfinish*(*linepath* a b) $= b$
⟨*proof*⟩

lemma *continuous-linepath-at*[*intro*]: *continuous* (*at* x) (*linepath* a b)
⟨*proof*⟩

lemma *continuous-on-linepath*[*intro*]: *continuous-on* s (*linepath* a b)
⟨*proof*⟩

lemma *path-linepath*[*intro*]: *path*(*linepath* a b)
⟨*proof*⟩

lemma *path-image-linepath*[*simp*]: *path-image*(*linepath* a b) $=$ (*closed-segment* a

$b)$
$\langle proof \rangle$

lemma *reversepath-linepath*[*simp*]: *reversepath*(*linepath a b*) = *linepath b a*
$\langle proof \rangle$

lemma *injective-path-linepath*:
assumes $a \neq b$ **shows** *injective-path*(*linepath a b*)
$\langle proof \rangle$

lemma *simple-path-linepath*[*intro*]: $a \neq b \Longrightarrow$ *simple-path*(*linepath a b*) $\langle proof \rangle$

25.5 Bounding a point away from a path.

lemma *not-on-path-ball*:
fixes g :: *real* $\Rightarrow$ *'a::heine-borel*
assumes *path g* $z \notin$ *path-image g*
shows $\exists\, e>0.$ *ball z e* $\cap$ (*path-image g*) = {} $\langle proof \rangle$

lemma *not-on-path-cball*:
fixes g :: *real* $\Rightarrow$ *'a::heine-borel*
assumes *path g* $z \notin$ *path-image g*
shows $\exists\, e>0.$ *cball z e* $\cap$ (*path-image g*) = {} $\langle proof \rangle$

25.6 Path component, considered as a "joinability" relation (from Tom Hales).

definition *path-component s x y* $\longleftrightarrow$ ($\exists\, g.$ *path g* $\wedge$ *path-image g* $\subseteq s \wedge$ *pathstart g* $= x \wedge$ *pathfinish g* $= y$)

lemmas *path-defs* = *path-def pathstart-def pathfinish-def path-image-def path-component-def*

lemma *path-component-mem*: **assumes** *path-component s x y* **shows** $x \in s$ $y \in s$
$\langle proof \rangle$

lemma *path-component-refl*: **assumes** $x \in s$ **shows** *path-component s x x*
$\langle proof \rangle$

lemma *path-component-refl-eq*: *path-component s x x* $\longleftrightarrow x \in s$
$\langle proof \rangle$

lemma *path-component-sym*: *path-component s x y* $\Longrightarrow$ *path-component s y x*
$\langle proof \rangle$

lemma *path-component-trans*: **assumes** *path-component s x y path-component s y z* **shows** *path-component s x z*
$\langle proof \rangle$

lemma *path-component-of-subset*: $s \subseteq t \Longrightarrow$ *path-component* $s\ x\ y \Longrightarrow$ *path-component* $t\ x\ y$
  ⟨*proof*⟩

25.7 Can also consider it as a set, as the name suggests.

lemma *path-component-set*: *path-component* $s\ x = \{\ y.\ (\exists\, g.\ path\ g \wedge path\text{-}image\ g \subseteq s \wedge pathstart\ g = x \wedge pathfinish\ g = y\ )\}$
  ⟨*proof*⟩

lemma *mem-path-component-set*:$x \in$ *path-component* $s\ y \longleftrightarrow$ *path-component* $s\ y\ x$ ⟨*proof*⟩

lemma *path-component-subset*: $(path\text{-}component\ s\ x) \subseteq s$
  ⟨*proof*⟩

lemma *path-component-eq-empty*: *path-component* $s\ x = \{\} \longleftrightarrow x \notin s$
  ⟨*proof*⟩

25.8 Path connectedness of a space.

definition *path-connected* $s \longleftrightarrow (\forall\, x{\in}s.\ \forall\, y{\in}s.\ \exists\, g.\ path\ g \wedge (path\text{-}image\ g) \subseteq s \wedge pathstart\ g = x \wedge pathfinish\ g = y)$

lemma *path-connected-component*: *path-connected* $s \longleftrightarrow (\forall\, x{\in}s.\ \forall\, y{\in}s.\ path\text{-}component\ s\ x\ y)$
  ⟨*proof*⟩

lemma *path-connected-component-set*: *path-connected* $s \longleftrightarrow (\forall\, x{\in}s.\ path\text{-}component\ s\ x = s)$
  ⟨*proof*⟩

25.9 Some useful lemmas about path-connectedness.

lemma *convex-imp-path-connected*:
  fixes s :: *'a::real-normed-vector set*
  assumes *convex s* **shows** *path-connected s*
  ⟨*proof*⟩

lemma *path-connected-imp-connected*: **assumes** *path-connected s* **shows** *connected s*
  ⟨*proof*⟩

lemma *open-path-component*:
  fixes s :: *'a::real-normed-vector set*
  assumes *open s* **shows** *open*(*path-component s x*)
  ⟨*proof*⟩

lemma *open-non-path-component*:
  fixes s :: *'a::real-normed-vector set*

assumes *open s* **shows** *open*(s − *path-component s x*)
⟨*proof*⟩

lemma *connected-open-path-connected*:
fixes s :: $'a$::*real-normed-vector set*
assumes *open s connected s* **shows** *path-connected s*
⟨*proof*⟩

lemma *path-connected-continuous-image*:
assumes *continuous-on s f path-connected s* **shows** *path-connected* (f ‘ s)
⟨*proof*⟩

lemma *homeomorphic-path-connectedness*:
s homeomorphic t $\Longrightarrow$ (*path-connected s* $\longleftrightarrow$ *path-connected t*)
⟨*proof*⟩

lemma *path-connected-empty*: *path-connected* {}
⟨*proof*⟩

lemma *path-connected-singleton*: *path-connected* {a}
⟨*proof*⟩

lemma *path-connected-Un*: **assumes** *path-connected s path-connected t* $s \cap t \neq \{\}$
shows *path-connected* ($s \cup t$) ⟨*proof*⟩

25.10 sphere is path-connected.

lemma *path-connected-punctured-universe*:
assumes $2 \leq$ *CARD*($'n$::*finite*) **shows** *path-connected*((*UNIV*::(*real*^$'n$) *set*) − {a}) ⟨*proof*⟩

lemma *path-connected-sphere*: **assumes** $2 \leq$ *CARD*($'n$::*finite*) **shows** *path-connected* {x::*real*^$'n$. *norm*($x - a$) = r} ⟨*proof*⟩

lemma *connected-sphere*: $2 \leq$ *CARD*($'n$) $\Longrightarrow$ *connected* {x::*real*^$'n$. *norm*($x - a$) = r}
⟨*proof*⟩

end

26 Fashoda: Fashoda meet theorem.

theory *Fashoda*
imports *Brouwer-Fixpoint Vec1 Path-Connected*
begin

26.1 Fashoda meet theorem.

lemma *infnorm-2*: *infnorm* $(x::real\hat{~}2) = max\ (abs(x\$1))\ (abs(x\$2))$
⟨*proof*⟩

lemma *infnorm-eq-1-2*: *infnorm* $(x::real\hat{~}2) = 1 \longleftrightarrow$
$(abs(x\$1) \leq 1 \wedge abs(x\$2) \leq 1 \wedge (x\$1 = -1 \vee x\$1 = 1 \vee x\$2 = -1 \vee x\$2 = 1))$
⟨*proof*⟩

lemma *infnorm-eq-1-imp*: **assumes** *infnorm* $(x::real\hat{~}2) = 1$ **shows** $abs(x\$1) \leq 1$ $abs(x\$2) \leq 1$
⟨*proof*⟩

lemma *fashoda-unit*: **fixes** $f\ g::real \Rightarrow real\hat{~}2$
assumes $f\ `\ \{-\ 1..1\} \subseteq \{-\ 1..1\}$ $g\ `\ \{-\ 1..1\} \subseteq \{-\ 1..1\}$
continuous-on $\{-\ 1..1\}$ f *continuous-on* $\{-\ 1..1\}$ g
$f\ (-\ 1)\$1 = -\ 1$ $f\ 1\$1 = 1$ $g\ (-\ 1)\ \$2 = -1$ $g\ 1\ \$2 = 1$
shows $\exists\, s \in \{-\ 1..1\}.\ \exists\, t \in \{-\ 1..1\}.\ f\ s = g\ t$ ⟨*proof*⟩

lemma *fashoda-unit-path*: **fixes** $f\ ::real \Rightarrow real\hat{~}2$ **and** $g\ ::real \Rightarrow real\hat{~}2$
assumes *path f path g path-image f* $\subseteq \{-\ 1..1\}$ *path-image g* $\subseteq \{-\ 1..1\}$
$(pathstart\ f)\$1 = -1$ $(pathfinish\ f)\$1 = 1$ $(pathstart\ g)\$2 = -1$ $(pathfinish\ g)\$2 = 1$
obtains z **where** $z \in$ *path-image f* $z \in$ *path-image g* ⟨*proof*⟩

lemma *fashoda*: **fixes** $b::real\hat{~}2$
assumes *path f path g path-image f* $\subseteq \{a..b\}$ *path-image g* $\subseteq \{a..b\}$
$(pathstart\ f)\$1 = a\$1$ $(pathfinish\ f)\$1 = b\$1$
$(pathstart\ g)\$2 = a\$2$ $(pathfinish\ g)\$2 = b\$2$
obtains z **where** $z \in$ *path-image f* $z \in$ *path-image g* ⟨*proof*⟩

26.2 Some slightly ad hoc lemmas I use below

lemma *segment-vertical*: **fixes** $a::real\hat{~}2$ **assumes** $a\$1 = b\$1$
shows $x \in$ *closed-segment a b* $\longleftrightarrow (x\$1 = a\$1 \wedge x\$1 = b\$1 \wedge$
$(a\$2 \leq x\$2 \wedge x\$2 \leq b\$2 \vee b\$2 \leq x\$2 \wedge x\$2 \leq a\$2))$ (**is** - = *?R*)
⟨*proof*⟩

lemma *segment-horizontal*: **fixes** $a::real\hat{~}2$ **assumes** $a\$2 = b\$2$
shows $x \in$ *closed-segment a b* $\longleftrightarrow (x\$2 = a\$2 \wedge x\$2 = b\$2 \wedge$
$(a\$1 \leq x\$1 \wedge x\$1 \leq b\$1 \vee b\$1 \leq x\$1 \wedge x\$1 \leq a\$1))$ (**is** - = *?R*)
⟨*proof*⟩

26.3 useful Fashoda corollary pointed out to me by Tom Hales.

lemma *fashoda-interlace*: **fixes** $a::real\hat{~}2$
assumes *path f path g*
path-image f $\subseteq \{a..b\}$ *path-image g* $\subseteq \{a..b\}$

$(pathstart\ f)\$2 = a\$2$ $(pathfinish\ f)\$2 = a\$2$
$(pathstart\ g)\$2 = a\$2$ $(pathfinish\ g)\$2 = a\$2$
$(pathstart\ f)\$1 < (pathstart\ g)\$1$ $(pathstart\ g)\$1 < (pathfinish\ f)\$1$
$(pathfinish\ f)\$1 < (pathfinish\ g)\$1$
obtains z **where** $z \in path\text{-}image\ f$ $z \in path\text{-}image\ g$
$\langle proof \rangle$

end

theory *Multivariate-Analysis*
imports *Determinants Integration Real-Integration Fashoda*
begin

end