Eudoxus Reals

Ata Keskin

March 17, 2025

Abstract

In this project, we present a peculiar construction of the real numbers, called "Eudoxus reals", using Isabelle/HOL. Similar to the classical method of Dedekind cuts, our approach starts from first principles. However, unlike Dedekind cuts, Eudoxus reals directly derive real numbers from integers, bypassing the intermediate step of constructing rational numbers.

This construction of the real numbers was first discovered by Stephen Schanuel. Schanuel named his construction after the ancient Greek philosopher Eudoxus, who developed a theory of magnitude and proportion to explain the relations between the discrete and the continuous. Our formalization is based on R.D. Arthan's paper detailing the construction [1]. For establishing the existence of multiplicative inverses for positive slopes, we used the idea of finding a suitable representative from Sławomir Kołodyńaski's construction on IsarMathLib which is based on Zermelo–Fraenkel set theory.

Contents

theory *Slope*
imports *HOL.Archimedean-Field*
begin

1 Slopes

1.1 Bounded Functions

definition *bounded* :: $('a \Rightarrow int) \Rightarrow bool$ **where**
bounded f $\longleftrightarrow$ *bdd-above* $((\lambda z. |f\ z|)$ ‘ *UNIV*)

lemma *boundedI*:
assumes $\bigwedge z.\ |f\ z| \leq C$
shows *bounded f*
⟨*proof*⟩

lemma *boundedE*[*elim*]:
assumes *bounded f* $\exists\, C.\ (\forall\, z.\ |f\ z| \leq C) \wedge 0 \leq C \Longrightarrow P$
shows P
⟨*proof*⟩

lemma *boundedE-strict*:
assumes *bounded f* $\exists\, C.\ (\forall\, z.\ |f\ z| < C) \wedge 0 < C \Longrightarrow P$
shows P
⟨*proof*⟩

lemma *bounded-alt-def*: *bounded f* $\longleftrightarrow (\exists\, C.\ \forall\, z.\ |f\ z| \leq C)$ ⟨*proof*⟩

lemma *bounded-iff-finite-range*: *bounded f* $\longleftrightarrow$ *finite* (*range f*)
⟨*proof*⟩

lemma *bounded-constant*:
shows *bounded* $(\lambda$-. $c)$
⟨*proof*⟩

lemma *bounded-add*:
assumes *bounded f bounded g*
shows *bounded* $(\lambda z.\ f\ z + g\ z)$
⟨*proof*⟩

lemma *bounded-mult*:
assumes *bounded f bounded g*
shows *bounded* $(\lambda z.\ f\ z * g\ z)$
⟨*proof*⟩

lemma *bounded-mult-const*:
assumes *bounded f*

shows *bounded* $(\lambda z.\ c * f\ z)$
⟨*proof*⟩

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

lemma *bounded-comp*:
assumes *bounded f*
shows *bounded* $(f\ o\ g)$ **and** *bounded* $(g\ o\ f)$
⟨*proof*⟩

1.2 Properties of Slopes

definition *slope* :: $(int \Rightarrow int) \Rightarrow bool$ **where**
slope f $\longleftrightarrow$ *bounded* $(\lambda(m, n).\ f\ (m + n) - (f\ m + f\ n))$

lemma *bounded-slopeI*:
assumes *bounded f*
shows *slope f*
⟨*proof*⟩

lemma *slopeE*[*elim*]:
assumes *slope f*
obtains C **where** $\bigwedge m\ n.\ |f\ (m + n) - (f\ m + f\ n)| \leq C\ 0 \leq C$ ⟨*proof*⟩

lemma *slope-add*:
assumes *slope f slope g*
shows *slope* $(\lambda z.\ f\ z + g\ z)$
⟨*proof*⟩

lemma *slope-symmetric-bound*:
assumes *slope f*
obtains C **where** $\bigwedge p\ q.\ |p * f\ q - q * f\ p| \leq (|p| + |q| + 2) * C\ 0 \leq C$
⟨*proof*⟩

lemma *slope-linear-bound*:
assumes *slope f*
obtains $A\ B$ **where** $\forall n.\ |f\ n| \leq A * |n| + B\ 0 \leq A\ 0 \leq B$
⟨*proof*⟩

lemma *slope-comp*:
assumes *slope f slope g*
shows *slope* $(f\ o\ g)$
⟨*proof*⟩

lemma *slope-scale*: *slope* $((*)\ a)$ ⟨*proof*⟩

lemma *slope-zero*: *slope* (λ-. *0*) ⟨*proof*⟩

lemma *slope-one*: *slope id* ⟨*proof*⟩

lemma *slope-uminus*: *slope uminus* ⟨*proof*⟩

lemma *slope-uminus'*:
  assumes *slope f*
  shows *slope* (λ*x*. − *f x*)
  ⟨*proof*⟩

lemma *slope-minus*:
  assumes *slope f slope g*
  shows *slope* (λ*x*. *f x* − *g x*)
  ⟨*proof*⟩

lemma *slope-comp-commute*:
  assumes *slope f slope g*
  shows *bounded* (λ*z*. (*f o g*) *z* − (*g o f*) *z*)
⟨*proof*⟩

lemma *int-set-infiniteI*:
  assumes $\bigwedge C.\ C \geq 0 \Longrightarrow \exists N{\geq}C.\ N \in (A :: int\ set)$
  shows *infinite A*
  ⟨*proof*⟩

lemma *int-set-infiniteD*:
  assumes *infinite* (*A* :: *int set*) $C \geq 0$
  obtains *z* **where** $z \in A$ $C \leq |z|$
⟨*proof*⟩

lemma *bounded-odd*:
  fixes *f* :: *int* ⇒ *int*
  assumes $\bigwedge z.\ z < 0 \Longrightarrow f\ z = -f\ (-z)$ $\bigwedge n.\ n > 0 \Longrightarrow |f\ n| \leq C$
  shows *bounded f*
⟨*proof*⟩

lemma *slope-odd*:
  assumes $\bigwedge z.\ z < 0 \Longrightarrow f\ z = -\ f\ (-\ z)$
    $\bigwedge m\ n.\ [\![m > 0;\ n > 0]\!] \Longrightarrow |f\ (m + n) - (f\ m + f\ n)| \leq C$
  shows *slope f*
⟨*proof*⟩

lemma *slope-bounded-comp-right-abs*:
  assumes *slope f bounded* (*f o abs*)
  shows *bounded f*
⟨*proof*⟩

corollary *slope-finite-range-iff*:

assumes *slope f*
shows *finite* (*range f*) $\longleftrightarrow$ *finite* (*f* ‘ {*0..*}) (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

lemma *slope-positive-lower-bound*:
assumes *slope f infinite* (*f* ‘ {*0..*} ∩ {*0<..*}) $D > 0$
obtains *M* **where** $M > 0$ $\bigwedge m.\ m > 0 \Longrightarrow (m + 1) * D \le f\ (m * M)$
⟨*proof*⟩

1.3 Set Membership of *Inf* and *Sup* on Integers

lemma *int-Inf-mem*:
fixes *S* :: *int set*
assumes $S \neq \{\}$ *bdd-below S*
shows *Inf S* ∈ *S*
⟨*proof*⟩

lemma *int-Sup-mem*:
fixes *S* :: *int set*
assumes $S \neq \{\}$ *bdd-above S*
shows *Sup S* ∈ *S*
⟨*proof*⟩

end

theory *Eudoxus*
imports *Slope*
begin

2 Eudoxus Reals

2.1 Type Definition

Two slopes are said to be equivalent if their difference is bounded.

definition *eudoxus-rel* :: (*int* ⇒ *int*) ⇒ (*int* ⇒ *int*) ⇒ *bool* (**infix** ‹$\sim_e$› *50*) **where**

$f \sim_e g \equiv$ *slope f* ∧ *slope g* ∧ *bounded* (λ*n. f n* − *g n*)

lemma *eudoxus-rel-equivp*:
part-equivp eudoxus-rel
⟨*proof*⟩

We define the reals as the set of all equivalence classes of the relation ($\sim_e$).

quotient-type *real* = (*int* ⇒ *int*) / *partial*: *eudoxus-rel*
⟨*proof*⟩

lemma *real-quot-type*: *quot-type* ($\sim_e$) *Abs-real Rep-real*

⟨*proof*⟩

lemma *slope-refl*: *slope f* = (*f* $\sim_e$ *f*)
⟨*proof*⟩

declare *slope-refl*[*THEN iffD2*, *simp*]

lemmas *slope-reflI* = *slope-refl*[*THEN iffD1*]

lemma *slope-induct*[*consumes 0*, *case-names slope*]:
assumes $\bigwedge f$. *slope f* ⟹ *P* (*abs-real f*)
shows *P x*
⟨*proof*⟩

lemma *abs-real-eq-iff*: *f* $\sim_e$ *g* ⟷ *slope f* ∧ *slope g* ∧ *abs-real f* = *abs-real g*
⟨*proof*⟩

lemma *abs-real-eqI*[*intro*]: *f* $\sim_e$ *g* ⟹ *abs-real f* = *abs-real g* ⟨*proof*⟩

lemmas *eudoxus-rel-sym*[*sym*] = *Quotient-symp*[*OF Quotient-real*, *THEN sympD*]
lemmas *eudoxus-rel-trans*[*trans*] = *Quotient-transp*[*OF Quotient-real*, *THEN transpD*]

lemmas *rep-real-abs-real-refl* = *Quotient-rep-abs*[*OF Quotient-real*, *OF slope-refl*[*THEN iffD1*], *intro*!]
lemmas *rep-real-iff* = *Quotient-rel-rep*[*OF Quotient-real*, *iff*]

declare *Quotient-abs-rep*[*OF Quotient-real*, *simp*]

lemma *slope-rep-real*: *slope* (*rep-real x*) ⟨*proof*⟩

lemma *eudoxus-relI*:
assumes *slope f slope g* $\bigwedge n$. $n \geq N$ ⟹ $|f\ n - g\ n| \leq C$
shows *f* $\sim_e$ *g*
⟨*proof*⟩

2.2 Addition and Subtraction

We define addition, subtraction and the additive identity as follows.

instantiation *real* :: {*zero*, *plus*, *minus*, *uminus*}
begin

quotient-definition
0 :: *real* **is** *abs-real* (λ-. *0*) ⟨*proof*⟩

declare *slope-zero*[*intro*!, *simp*]

lemma *zero-iff-bounded*: *f* $\sim_e$ (λ-. *0*) ⟷ *bounded f* ⟨*proof*⟩
lemma *zero-iff-bounded'*: *x* = *0* ⟷ *bounded* (*rep-real x*) ⟨*proof*⟩

lemma *zero-def*: *0* = *abs-real* (λ-. *0*) ⟨*proof*⟩

definition *eudoxus-plus* :: (*int* ⇒ *int*) ⇒ (*int* ⇒ *int*) ⇒ (*int* ⇒ *int*) (**infixl** ‹$+_e$› *60*) **where**
  (*f* :: *int* ⇒ *int*) $+_e$ *g* = (λ*z*. *f z* + *g z*)

declare *slope-add*[*intro*, *simp*]

quotient-definition
  (+) :: (*real* ⇒ *real* ⇒ *real*) **is** ($+_e$)
⟨*proof*⟩

lemmas *eudoxus-plus-cong* = *apply-rsp'*[*OF plus-real.rsp*, *THEN rel-funD*, *intro*]

lemma *abs-real-plus*[*simp*]:
  assumes *slope f slope g*
  shows *abs-real f* + *abs-real g* = *abs-real* (*f* $+_e$ *g*)
  ⟨*proof*⟩

definition *eudoxus-uminus* :: (*int* ⇒ *int*) ⇒ (*int* ⇒ *int*) (‹$-_e$›) **where**
  $-_e$ (*f* :: *int* ⇒ *int*) = (λ*x*. − *f x*)

declare *slope-uminus'*[*intro*, *simp*]

quotient-definition
  (*uminus*) :: (*real* ⇒ *real*) **is** $-_e$
⟨*proof*⟩

lemmas *eudoxus-uminus-cong* = *apply-rsp'*[*OF uminus-real.rsp*, *simplified*, *intro*]

lemma *abs-real-uminus*[*simp*]:
  assumes *slope f*
  shows − *abs-real f* = *abs-real* ($-_e$ *f*)
  ⟨*proof*⟩

definition *x* − (*y*::*real*) = *x* + − *y*

declare *slope-minus*[*intro*, *simp*]

lemma *abs-real-minus*[*simp*]:
  assumes *slope g slope f*
  shows *abs-real g* − *abs-real f* = *abs-real* (*g* $+_e$ ($-_e$ *f*))
  ⟨*proof*⟩

instance ⟨*proof*⟩
end

The Eudoxus reals equipped with addition and negation specified as above constitute an Abelian group.

instance *real* :: *ab-group-add*
⟨*proof*⟩

2.3 Multiplication

We define multiplication as the composition of two slopes.

instantiation *real* :: {*one*, *times*}
begin

quotient-definition
1 :: *real* **is** *abs-real id* ⟨*proof*⟩

declare *slope-one*[*intro*!, *simp*]

lemma *one-def*: *1* = *abs-real id* ⟨*proof*⟩

definition *eudoxus-times* :: (*int* ⇒ *int*) ⇒ (*int* ⇒ *int*) ⇒ *int* ⇒ *int* (**infixl** ‹$*_e$› *60*) **where**
$f *_e g = f \circ g$

declare *slope-comp*[*intro*, *simp*]
declare *slope-scale*[*intro*, *simp*]

quotient-definition
(∗) :: *real* ⇒ *real* ⇒ *real* **is** ($*_e$)
⟨*proof*⟩

lemmas *eudoxus-times-cong* = *apply-rsp′*[*OF times-real.rsp*, *THEN rel-funD*, *intro*]
lemmas *eudoxus-rel-comp* = *eudoxus-times-cong*[*unfolded eudoxus-times-def*]

lemma *eudoxus-times-commute*:
assumes *slope f slope g*
shows $(f *_e g) \sim_e (g *_e f)$
⟨*proof*⟩

lemma *abs-real-times*[*simp*]:
assumes *slope f slope g*
shows *abs-real f* ∗ *abs-real g* = *abs-real* $(f *_e g)$
⟨*proof*⟩

instance ⟨*proof*⟩
end

lemma *neg-one-def*: − *1* = *abs-real* ($-_e$ *id*) ⟨*proof*⟩
lemma *slope-neg-one*[*intro*, *simp*]: *slope* ($-_e$ *id*) ⟨*proof*⟩

With the definitions provided above, the Eudoxus reals are a commutative ring with unity.

instance *real* :: *comm-ring-1*
⟨*proof*⟩

lemma *real-of-nat*:
of-nat n = *abs-real* ((∗) (*of-nat n*))
⟨*proof*⟩

lemma *real-of-int*:
of-int z = *abs-real* ((∗) *z*)
⟨*proof*⟩

The Eudoxus reals are a ring of characteristic *0*.

instance *real* :: *ring-char-0*
⟨*proof*⟩

2.4 Ordering

We call a slope positive, if it tends to infinity. Similarly, we call a slope negative if it tends to negative infinity.

instantiation *real* :: {*ord*, *abs*, *sgn*}
begin

definition *pos* :: (*int* ⇒ *int*) ⇒ *bool* **where**
pos f = (∀ *C* ≥ *0*. ∃ *N*. ∀ *n* ≥ *N*. *f n* ≥ *C*)

definition *neg* :: (*int* ⇒ *int*) ⇒ *bool* **where**
neg f = (∀ *C* ≥ *0*. ∃ *N*. ∀ *n* ≥ *N*. *f n* ≤ −*C*)

lemma *pos-neg-exclusive*: ¬ (*pos f* ∧ *neg f*) ⟨*proof*⟩

lemma *pos-iff-neg-uminus*: *pos f* = *neg* ($-_e$ *f*) ⟨*proof*⟩

lemma *neg-iff-pos-uminus*: *neg f* = *pos* ($-_e$ *f*) ⟨*proof*⟩

lemma *pos-iff*:
assumes *slope f*
shows *pos f* = *infinite* (*f* ‘ {*0*..} ∩ {*0*<..}) (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *neg-iff*:
assumes *slope f*
shows *neg f* = *infinite* (*f* ‘ {*0*..} ∩ {..<*0*}) (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *pos-cong*:
assumes *f* $\sim_e$ *g*
shows *pos f* = *pos g*
⟨*proof*⟩

lemma *neg-cong*:
  assumes $f \sim_e g$
  shows *neg f* = *neg g*
⟨*proof*⟩

lemma *pos-iff-nonneg-nonzero*:
  assumes *slope f*
  shows *pos f* ⟷ (¬ *neg f*) ∧ (¬ *bounded f*) (**is** *?lhs* ⟷ *?rhs*)
⟨*proof*⟩

lemma *neg-iff-nonpos-nonzero*:
  assumes *slope f*
  shows *neg f* ⟷ (¬ *pos f*) ∧ (¬ *bounded f*)
  ⟨*proof*⟩

We define the sign of a slope to be *id* if it is positive, $-_e$ *id* if it is negative and λ-. *0* otherwise.

definition *eudoxus-sgn* :: (*int* ⇒ *int*) ⇒ (*int* ⇒ *int*) **where**
  eudoxus-sgn f = (*if pos f then id else if neg f then* $-_e$ *id else* (λ-. *0*))

lemma *eudoxus-sgn-iff*:
  assumes *slope f*
  shows *eudoxus-sgn f* = (λ-. *0*) ⟷ *bounded f*
    eudoxus-sgn f = *id* ⟷ *pos f*
    eudoxus-sgn f = ($-_e$ *id*) ⟷ *neg f*
  ⟨*proof*⟩

quotient-definition
  (*sgn* :: *real* ⇒ *real*) **is** *eudoxus-sgn*
  ⟨*proof*⟩

lemmas *eudoxus-sgn-cong* = *apply-rsp*′[*OF sgn-real.rsp*, *intro*]

lemma *eudoxus-sgn-cong*′[*cong*]:
  assumes $f \sim_e g$
  shows *eudoxus-sgn f* = *eudoxus-sgn g*
  ⟨*proof*⟩

lemma *sgn-range*: *sgn* (*x* :: *real*) ∈ {−*1*, *0*, *1*} ⟨*proof*⟩

lemma *sgn-abs-real-zero-iff*:
  assumes *slope f*
  shows *sgn* (*abs-real f*) = *0* ⟷ (*eudoxus-sgn f* = (λ-. *0*)) (**is** *?lhs* ⟷ *?rhs*)
  ⟨*proof*⟩

lemma *sgn-zero-iff*[*simp*]: *sgn* (*x* :: *real*) = *0* ⟷ *x* = *0*
  ⟨*proof*⟩

lemma *sgn-zero*[*simp*]: $sgn\ (0 :: real) = 0$ ⟨*proof*⟩

lemma *sgn-abs-real-one-iff*:
assumes *slope f*
shows $sgn\ (abs\text{-}real\ f) = 1 \longleftrightarrow pos\ f$
⟨*proof*⟩

lemmas *sgn-pos* = *sgn-abs-real-one-iff*[*THEN iffD2*, *simp*]

lemma *sgn-one*[*simp*]: $sgn\ (1 :: real) = 1$ ⟨*proof*⟩

lemma *sgn-abs-real-neg-one-iff*:
assumes *slope f*
shows $sgn\ (abs\text{-}real\ f) = -\ 1 \longleftrightarrow neg\ f$
⟨*proof*⟩

lemmas *sgn-neg* = *sgn-abs-real-neg-one-iff*[*THEN iffD2*, *simp*]

lemma *sgn-neg-one*[*simp*]: $sgn\ (-\ 1 :: real) = -\ 1$ ⟨*proof*⟩

lemma *sgn-plus*:
assumes $sgn\ x = (1 :: real)\ sgn\ y = 1$
shows $sgn\ (x + y) = 1$
⟨*proof*⟩

lemma *sgn-times*: $sgn\ ((x :: real) * y) = sgn\ x * sgn\ y$
⟨*proof*⟩

lemma *sgn-uminus*: $sgn\ (-\ (x :: real)) = -\ sgn\ x$ ⟨*proof*⟩

lemma *sgn-plus'*:
assumes $sgn\ x = (-1 :: real)\ sgn\ y = -1$
shows $sgn\ (x + y) = -1$
⟨*proof*⟩

lemma *pos-dual-def*:
assumes *slope f*
shows $pos\ f = (\forall\ C \geq 0.\ \exists\ N.\ \forall\ n \leq N.\ f\ n \leq -C)$
⟨*proof*⟩

lemma *neg-dual-def*:
assumes *slope f*
shows $neg\ f = (\forall\ C \geq 0.\ \exists\ N.\ \forall\ n \leq N.\ f\ n \geq C)$
⟨*proof*⟩

lemma *pos-representative*:
assumes *slope f pos f*
obtains *g* **where** $f \sim_e g\ \bigwedge n.\ n \geq N \Longrightarrow g\ n \geq C$
⟨*proof*⟩

lemma *pos-representative'*:
  assumes *slope f pos f*
  obtains g **where** $f \sim_e g \bigwedge n.\ g\ n \geq C \Longrightarrow n \geq N$
⟨*proof*⟩

lemma *neg-representative*:
  assumes *slope f neg f*
  obtains g **where** $f \sim_e g \bigwedge n.\ n \geq N \Longrightarrow g\ n \leq -\ C$
⟨*proof*⟩

lemma *neg-representative'*:
  assumes *slope f neg f*
  obtains g **where** $f \sim_e g \bigwedge n.\ g\ n \leq -\ C \Longrightarrow n \geq N$
⟨*proof*⟩

We call a real x less than another real y, if their difference is positive.

definition
  $x < (y{::}real) \equiv sgn\ (y - x) = 1$

definition
  $x \leq (y{::}real) \equiv x < y \vee x = y$

definition
  abs-real: $|x :: real| = (if\ 0 \leq x\ then\ x\ else\ -\ x)$

instance ⟨*proof*⟩
end

instance *real* :: *linorder*
⟨*proof*⟩

lemma *real-leI*:
  assumes $sgn\ (y - x) \in \{0 :: real,\ 1\}$
  shows $x \leq y$
  ⟨*proof*⟩

lemma *real-lessI*:
  assumes $sgn\ (y - x) = (1 :: real)$
  shows $x < y$
  ⟨*proof*⟩

lemma *abs-real-leI*:
  assumes *slope f slope g* $\bigwedge z.\ z \geq N \Longrightarrow f\ z \geq g\ z$
  shows *abs-real f* $\geq$ *abs-real g*
⟨*proof*⟩

lemma *abs-real-lessI*:

assumes *slope f slope g* $\bigwedge z.\ z \geq N \Longrightarrow f\ z \geq g\ z\ \bigwedge C.\ C \geq 0 \Longrightarrow \exists z.\ f\ z \geq g\ z + C$
shows *abs-real f* > *abs-real g*
⟨*proof*⟩

lemma *abs-real-lessD*:
assumes *slope f slope g abs-real f* > *abs-real g*
obtains *z* **where** $z \geq N\ f\ z > g\ z$
⟨*proof*⟩

2.5 Multiplicative Inverse

We now define the multiplicative inverse. We start by constructing a candidate for positive slopes first and then extend it to the entire domain using the choice function *Eps*.

instantiation *real* :: {*inverse*}
begin

definition *eudoxus-pos-inverse* :: (*int* ⇒ *int*) ⇒ (*int* ⇒ *int*) **where**
eudoxus-pos-inverse f z = *sgn z* ∗ *Inf* ({*0*..} ∩ {*n*. *f n* ≥ |*z*|})

lemma *eudoxus-pos-inverse*:
assumes *slope f pos f*
obtains *g* **where** $f \sim_e g$ *slope* (*eudoxus-pos-inverse g*) *eudoxus-pos-inverse g* $*_e$ $f \sim_e$ *id*
⟨*proof*⟩

definition *eudoxus-inverse* :: (*int* ⇒ *int*) ⇒ (*int* ⇒ *int*) **where**
eudoxus-inverse f = (*if* ¬ *bounded f then SOME g. slope g* ∧ (*g* $*_e$ *f*) $\sim_e$ *id else* (λ-. *0*))

lemma
assumes *slope f*
shows *slope-eudoxus-inverse*: *slope* (*eudoxus-inverse f*) (**is** *?slope*) **and**
eudoxus-inverse-id: ¬ *bounded f* ⟹ *eudoxus-inverse f* $*_e$ *f* $\sim_e$ *id* (**is** ¬ *bounded f* ⟹ *?id*)
⟨*proof*⟩

quotient-definition
(*inverse* :: *real* ⇒ *real*) **is** *eudoxus-inverse*
⟨*proof*⟩

definition
x div (*y*::*real*) = *inverse y* ∗ *x*

instance ⟨*proof*⟩
end

lemmas *eudoxus-inverse-cong* = *apply-rsp′*[*OF inverse-real.rsp*, *intro*]

lemma *eudoxus-inverse-abs*[*simp*]:
  assumes *slope f* $\neg$ *bounded f*
  shows *inverse* (*abs-real f*) $*$ *abs-real f* = *1*
  ⟨*proof*⟩

The Eudoxus reals are a field, with inverses defined as above.

instance *real* :: *field*
⟨*proof*⟩

instantiation *real* :: *distrib-lattice*
begin

definition
  (*inf* :: *real* ⇒ *real* ⇒ *real*) = *min*

definition
  (*sup* :: *real* ⇒ *real* ⇒ *real*) = *max*

instance ⟨*proof*⟩

end

The ordering on the Eudoxus reals is linear.

instance *real* :: *linordered-field*
⟨*proof*⟩

The Eudoxus reals fulfill the Archimedean property.

instance *real* :: *archimedean-field*
⟨*proof*⟩

2.6 Completeness

To show that the Eudoxus reals are complete, we first introduce the floor function.

instantiation *real* :: *floor-ceiling*
begin

definition
  (*floor* :: (*real* ⇒ *int*)) = (λx. (*SOME z*. *of-int z* $\leq x \wedge x <$ *of-int z* + *1*))

instance
⟨*proof*⟩
end

lemma *eudoxus-dense-rational*:
  fixes *x y* :: *real*
  assumes $x < y$

obtains $m\ n$ **where** $x < (\mathit{of\text{-}int}\ m\ /\ \mathit{of\text{-}int}\ n)$ $(\mathit{of\text{-}int}\ m\ /\ \mathit{of\text{-}int}\ n) < y$ $n > 0$
$\langle proof \rangle$

The Eudoxus reals are a complete field.

lemma *eudoxus-complete*:
assumes $S \neq \{\}$ *bdd-above* S
obtains $u :: \mathit{real}$ **where** $\bigwedge s.\ s \in S \Longrightarrow s \leq u$ $\bigwedge y.\ (\bigwedge s.\ s \in S \Longrightarrow s \leq y) \Longrightarrow u \leq y$
$\langle proof \rangle$

end

References

[1] R. D. Arthan. The Eudoxus real numbers. arXiv:math/0405454, 2004.