이학석사 학위논문

# A Construction of the Reals from an Intuitive and Formal Perspective

직관적 관점과 형식적 관점에서의 실수 건설

2023년 2월

서울대학교 대학원
수 리 과 학 부
남 도 윤

이학석사 학위논문

# A Construction of the Reals from an Intuitive and Formal Perspective

직관적 관점과 형식적 관점에서의 실수 건설

2023년 2월


서울대학교 대학원

수 리 과 학 부

남 도 윤

# A Construction of the Reals from an Intuitive and Formal Perspective

직관적 관점과 형식적 관점에서의 실수 건설

지도교수 Otto van Koert
이 논문을 이학석사 학위논문으로 제출함

2022년 12월

서울대학교 대학원

수 리 과 학 부

남 도 윤

남도윤의 이학석사 학위 논문을 인준함

2023년 2월

위 원 장: 국 웅

부위원장: Otto van Koert

위 원: 서 인 석

# Abstract

Based on intuitive facts about a straight line, we define what a straight line is with the help of R. Dedekind. And we introduce a proof assistant program Coq. After that, adding two operations - addition and multiplication - to the reals, we show that the reals is a Dedekind-complete ordered field by complementing natural language and Coq.

**keywords**: Real numbers, Dedekind-completeness, Proof assistant program, Coq

**student number**: 2018-24398

# Contents

# Chapter 1

# Introduction

A straight line, along with a circle, is one of the longest studied objects by mathematicians. And we can see these objects in nature; for example, a sea horizon line and the Sun. And as civilization developed, mankind gradually became able to make things that resemble straight lines and circles more precisely. These days, we are surrounded by these things.

Coq is a proof assistant program. One can define axioms, definitions, properties, or theorems in Coq, and can write a proof code for a theorem. Coq checks line-by-line correctness of the human-writing proof codes, and if it meets inappropriate line then Coq stops and show what the error is. If Coq proceeds and there is no more things to prove, then Coq shows ‘no more goals’ in the screen. Then people can assert that this proof is correct on the built-in logic of Coq.

Because the set of all rational numbers is not enough to fulfill a straight line, we need a more sufficient condition to be a straight line. This condition is called Dedekind-completeness, and there are many equivalent forms such as least-upper-bound-properties.

We study again how a straight line transforms from a geometric object to an algebraic object. And we use Coq to define definitions and to prove theorems.

# Chapter 2

# Strengths for using Coq

If someone have to choose a method doing lots of calculations (for example, multiplication of two $100$ digits numbers) by hand or by a computer, then almost everyone choose a computer. Because it is faster and more accurate than human. The advantage of using Coq is similar: for proof checking, (if codes are written,) it is extremely faster and more accurate than human.

Thomas Hales' paper [1] summarize well how computer influences to mathematicians historically, and introduce proof assistants and possible weakness of proof assistants. Computers help people in calculation and visualization. (Of course, it also helps with networking.)

A proof assistant is a software program in which people can make a formal proof and check the proof is correct. Proof assistant programs are based on the type theory instead of ZFC set theory. In classical logic, the law of excluded middle ($p \vee \neg p$ for every property $p$) is accepted; however it is not accepted in constructive logic.

For example, try to prove a statement that 'there is no rational number whose squared is two.' Let us denote this property by $p$, which can be written in logic symbol as follows : not ($\exists q \in \mathbb{Q}$, $q^2 = 2$). Then 'not $p$' is the statement ($\exists q \in \mathbb{Q}$, $q^2 = 2$). We basically assumed that $p$ or not $p$ is true. (It is the law of excluded middle.) And by proving that 'not $p$' is false, we conclude that $p$ is true. In classical logic this proof is accepted, however it is not accepted in constructive logic.

A proof assistant may be constructive or classical. In special, Coq is constructive. And there are many useful proof tactics in Coq. They help people can write formal proof more efficiently and easily. To learn Coq, this book [2] is useful. And the official website of Coq provides lots of materials.

# Chapter 3

# Characterization of a straight line

## 3.1 Dense linearly ordered sets without endpoints

In this section, let $X$ denote a set.

**Definition 3.1.1.** A *binary relation* $R$ on $X$ is a set whose elements are in $X \times X$. If $(a, b) \in R$, then for convenience, we use the notation $aRb$.

**Example.** Each of the sets $\{(n, n) \mid n \in \mathbb{N}\}$, $\{(n, m) \mid n \in \mathbb{N},\ m \in \mathbb{N},\ n < m\}$, and $\{(n, m) \mid n \in \mathbb{N},\ m \in \mathbb{N},\ n \leq m\}$ are binary relations on $\mathbb{N}$, respectively. We denote each binary relations by $=_{\mathbb{N}}$, $<_{\mathbb{N}}$, and $\leq_{\mathbb{N}}$ in order.

**Definition 3.1.2.** Let $R$ be a binary relation on $X$.

(a) $R$ is *reflexive* if for all $a \in X$, $aRa$.

(b) $R$ is *irreflexive* if for all $a \in X$, not $aRa$.

(c) $R$ is *symmetric* if for all $a, b \in X$, $aRb$ implies $bRa$.

(d) $R$ is *asymmetric* if for all $a, b \in X$, $aRb$ implies not $bRa$.

(e) $R$ is *antisymmetric* if for all $a, b \in X$, $aRb$ and $bRa$ imply $a = b$.

(f) $R$ is *transitive* if for all $a, b, c \in X$, $aRb$ and $bRc$ imply $aRc$.

Reflexivity and irreflexivity are properties related to one element; symmetry, asymmetry and antisymmetry are properties related to two elements; transitivity is a property related to three elements.

We can check that a binary relation $=_{\mathbb{N}}$ is reflexive, symmetric, and transitive; a binary relation $<_{\mathbb{N}}$ is irreflexive, asymmetric, and transitive; a binary relation $\leq_{\mathbb{N}}$ is reflexive, antisymmetric, and transitive. By generalizing these, we define an equivalence relation, a strict order, and a partial order.

**Definition 3.1.3.** Let $R$ be a binary relation on $X$.

(a) $R$ is called an *equivalence relation* on $X$ if it is reflexive, symmetric, and transitive.

(b) $R$ is called a *strict order* on $X$ if it is irreflexive, asymmetric, and transitive.

(c) $R$ is called a *partial order* on $X$ if it is reflexive, antisymmetric, and transitive.

**Remark.** We can easily show that 'irreflexivity and transitivity implies asymmetry', and 'asymmetry implies irreflexivity'. Hence, to show that a binary relation $R$ is a strict order, it is enough to show that $R$ is irreflexive and transitive, or $R$ is asymmetric and transitive.

For natural numbers $a$ and $b$, we are accustomed the fact that $a < b$ if and only if $a \leq b$ and $a \neq b$, and that $a \leq b$ if and only if $a < b$ or $a = b$. This relation between partial order and strict order can be generalized.

The following two theorems are well known theorems. (see [3])

**Theorem 3.1.1.** If $T$ is a partial order on $X$, then we define a binary relation $S_T$ on $X$ as follows : $(a, b) \in S_T \iff (a, b) \in T$ and $a \neq b$. Then this binary relation $S_T$ is a strict order on $X$.

Similarly, if $U$ is a strict order on $X$, then we define a binary relation $P_U$ on $X$ as follows : $(a, b) \in P_U \iff (a, b) \in U$ or $a = b$. Then this binary relation $P_U$ is a partial order on $X$.

**Theorem 3.1.2.** If $T$ is a partial order on $X$, then $S_T$ is a strict order on $X$, and $P_{S_T}$ is a partial order on $X$. These two partial orders $T$ and $P_{S_T}$ are the same.

Similarly, from a strict order $U$ on $X$, we can make a partial order $P_U$ on $X$, and then we can make a strict order $S_{P_U}$ on $X$. Then $U = S_{P_U}$.

Thus we can interchange a partial order and a strict order. For example, when it is easy to deal with strict order, then we use a strict order. And after that, if dealing with partial order is easy, then we use the corresponding partial order.

**Notation.** For notational convenience, we shall use $\leq_X$ for a partial order on $X$, and use $<_X$ for a strict order on $X$. When we use both $\leq_X$ and $<_X$ notation in the same paragraph, then the two orders are assumed to be corresponding orders.

**Definition 3.1.4.** Let $\leq_X$ be a partial order on $X$. If $a \leq_X b$ or $b \leq_X a$ for some $a, b \in X$, then we say that $a$ and $b$ are *comparable* in the order $\leq_X$. If every two elements of $X$ are comparable in the order $\leq_X$, then we say that a pair $(X, \leq_X)$ is a *linearly ordered set*.

**Remark.** By Theorem 3.1.1, it is easy to check that '$a \leq_X b$ or $b \leq_X a$' and '$a <_X b$ or $a = b$ or $b <_X a$' are equivalent. Thus the latter statement can be used as a definition of comparable elements. And we can easily show that if $a <_X b$ or $a = b$ or $b <_X a$, then only one of the three statements is true.

There is no natural number between arbitrary two consecutive natural numbers. For example, there is no natural number between 3 and 4. However, for any two distinct rational numbers, there is another rational number between them. For example between $\frac{1}{5}$ and $\frac{4}{7}$, the rational number $\frac{1}{3}$ exists.

**Definition 3.1.5.** Let $(X, <_X)$ be a linearly ordered set. It is *dense*, if for every two distinct elements of $X$, there is another element of $X$ between them, *i.e.*, $\forall a, b \in X\ (a <_X b \implies \exists c \in X,\ a <_X c <_X b)$.

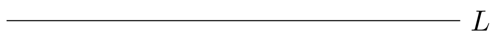
**Remark.** Let $(X, <_X)$ be a dense linearly ordered set and $Y$ be a subset of $X$. If for every $a, b$ in $X$ with $a < b$ there exists corresponding $c$ in $Y$ such that $a <_X c <_X b$, then we say that $Y$ is dense in $X$.

**Definition 3.1.6.** Let $(X, <_X)$ be a linearly ordered set. If for every element $x$ of $X$, there exist two elements $y$, $z$ of $X$ such that $y <_X x$ and $x <_X z$, then $X$ is said to be *without endpoints*.

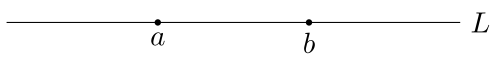
It is well-known that $\mathbb{Q}$ is a dense linearly ordered set without endpoints.

## 3.2 Dedekind-complete

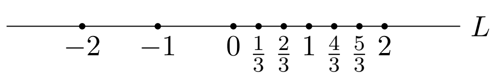
Let $L$ be a (horizontal) straight line without endpoints, or abusively, the set of points of this straight line. The following argument for $L$ depends on intuitive observation.




For two distinct points $a$ and $b$ of $L$, we define $a <_L b$ if $a$ is on the left of $b$. Then $(L, <_L)$ is a dense linearly ordered set without endpoints.

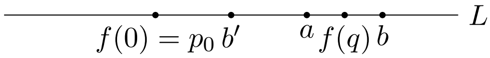



And as it is well known, we can make a correspondence from each point of $\mathbb{Q}$ to some point of $L$.




Let $f : \mathbb{Q} \to L$ be such correspondence. We define $f$ in this way: we assign some point $p_0$ in $L$ to 0, and some other point $p_1$ (on the right of $p_0$) to 1. And then we assign the point $p_2$ to 2 which satisfies that $\overrightarrow{p_0 p_1} = \overrightarrow{p_1 p_2}$, *i.e.*, have the same distance and direction. In this way, we can define $f(x)$ for all $x \in \mathbb{Z}$, and also we can expand $f$ to $\mathbb{Q}$. Then $f$ is an order-preserving map, i.e., $\forall q_1, q_2 \in \mathbb{Q}, q_1 < q_2 \implies f(q_1) <_L f(q_2)$.

And for every two distinct points $a, b$ of $L$, there exists a rational number $q$ such that $f(q)$ is between $a$ and $b$.




For example, let $a$, $b$ be points of $L$ such that $0 <_L a <_L b$. Let $p_0$ denote $f(0)$. If we move $a$ to $p_0$ and $b$ to $b'$ such that $\overrightarrow{p_0 b'} = \overrightarrow{ab}$. Then for natural number $n$, as $n$ increases, the corresponding point $f(1/n)$ is close to $p_0$. Thus there exists $n \in \mathbb{N}$ such that $f(1/n) <_L b'$. It is a kind of Archimedean property. Thus the distance between two points $a$ and $b$ are greater than $f(1/n)$. Roughly speaking, then the distance between two points $n \cdot a$ and $n \cdot b$ are greater than $f(1)$, where $n \cdot a$ means that $a + \cdots + a$ for $n$ times, or the endpoint of $p_0 + n \times \overrightarrow{p_0 a}$. Thus there exists $m \in \mathbb{N}$ such that $n \cdot a <_L f(m) <_L n \cdot b$, equivalently $a <_L f(m/n) <_L b$. This explains the necessity of the condition that $f(\mathbb{Q})$ is dense in $L$.

**Notation.** Assume that $(X, <_X)$ is a dense linearly ordered set without endpoints and $x$ is an element of $X$. For convenience, we shall use the following notations.

$$\begin{aligned} (-\infty, y)_X &:= \{x \in X \mid x <_X y\}, & (-\infty, y]_X &:= \{x \in X \mid x \leq_X y\}, \\ (y, \infty)_X &:= \{x \in X \mid y <_X x\}, & [y, \infty)_X &:= \{x \in X \mid y \leq_X x\}. \end{aligned}$$

We know that $\mathbb{Q}$ cannot fulfill $L$. For example $\sqrt{2}$ is constructed from unit distance 1 with a ruler and a compass; $\sqrt{2}$ is a distance of a diagonal of a unit square. However we know that $\sqrt{2}$ is not a rational number.

Dedekind [4] first consider what properties a straight line and $\mathbb{Q}$ commonly have. A linear order is one of them. And he think that if we choose a point $p$ in $L$, then this point $p$ divides $L$ into two pieces; $(-\infty, p)_L$ and $[p, \infty)_L$, or $(-\infty, p]_L$ and $(p, \infty)_L$. In each partitions, each element of the first part is less than (or on the left of) each element of the second part. And this property also holds in $\mathbb{Q}$.

**Definition 3.2.1.** Let $(X, <_X)$ be a dense linearly ordered set (without endpoints). Let $\{A, B\}$ be a partition of $X$, *i.e.*, $A \cup B = X$, $A \cap B = \emptyset$ ,$A \neq \emptyset$, and $B \neq \emptyset$. A pair

$(A, B)$ is called a *comparable partition* of $X$ if $a <_X b$ for every $a \in A$ and for every $b \in B$.

In other words, a pair $(A, B)$ is a comparable partition of $X$ if $\{A, B\}$ is a partition of $X$ and every element of $A$ is less than every element of $B$.

**Lemma 3.2.1.** Assume that $(A, B)$ is a comparable partition of $X$. If $a \in A$ and $a' <_X a$ then $a' \in A$, and if $b \in B$ and $b <_X b'$ then $b' \in B$. And $A$ is bounded above, and $B$ is bounded below.

*Proof.* Assume that $a \in A$ and $a' <_X a$. Since $\{A, B\}$ is a partition of $X$, if $a' \notin A$ then $a' \in B$. Because $(A, B)$ is a comparable partition of $X$ and $a \in A$ and $a' \in B$, it follows that $a <_X a'$. By asymmetry of $<_X$, we meet a contradiction. Thus if $a \in A$ and $a' <_X a$, then $a' \in A$. And for arbitrary fixed element $b$ of $B$, we see that $a <_X b$ for every $a \in A$. Hence $A$ is bounded above. The rest part is proved by the same way. $\square$

If $(A, B)$ is a comparable partition of a dense linearly ordered set $X$, then there are four possibilities:

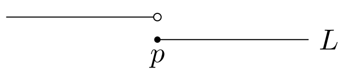
(a) $A$ does not have the greatest element, and $B$ has the least element.

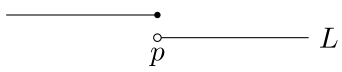
(b) $A$ has the greatest element, and $B$ does not have the least element.

(c) $A$ does not have the greatest element, and $B$ does not have the least element.

(d) $A$ has the greatest element, and $B$ has the least element.

If $A$ has the greatest element $\alpha$ and $B$ has the least element $\beta$, then it follows that $\alpha <_X \beta$. Since $X$ is dense, there exists $c \in X$ such that $\alpha <_X c <_X \beta$. If $c \in A$, then $\alpha$ is not the greatest element of $A$; if $c \in B$, then $\beta$ is not the least element of $B$, which leads a contradiction in each case. Hence the case (d) does not happen.






We can say that each point $p$ of $L$ make two comparable partitions of $L$: $(-\infty, p)_L$ and $[p, \infty)_L$, or $(-\infty, p]_L$ and $(p, \infty)_L$. (These two comparable partitions corresponds to $p$, thus we can identify them if we want.) It is Dedekind's idea for completeness that every comparable partition of $L$ is made by some point $p$ of $L$ [4], or equivalently, for every comparable partition $(A, B)$ of $L$, $A$ has the greatest element $p$ or $B$ has the least element $p$, where $p$ is in $L$.

**Definition 3.2.2.** Let $X$ be a dense linearly ordered set (without endpoints). The set $X$ is *Dedekind-complete* if for each comparable partition $(A, B)$ of $X$, the set $A$ has the greatest element or $B$ has the least element in $X$.

Note that $\mathbb{Q}$ is not Dedekind-complete. For example, let $A$ and $B$ be two subsets of $\mathbb{Q}$ defined by

$$A = \{q \in \mathbb{Q} : q \leq 0\} \cup \{q \in \mathbb{Q} : 0 < q \text{ and } q^2 < 2\},$$
$$B = \{q \in \mathbb{Q} : 0 < q \text{ and } 2 < q^2\}.$$

Then $(A, B)$ is a comparable partition of $\mathbb{Q}$. However, we can easily show that $A$ does not have the greatest element and $B$ does not have the least element. Thus $\mathbb{Q}$ is not Dedekind-complete.

In summary, we characterize a straight line $L$ as follows :

(a) $L$ is a dense linearly ordered set without endpoints.

(b) There is an order-preserving map $f : \mathbb{Q} \to L$ such that $f(\mathbb{Q})$ is dense in $L$.

(c) $L$ is Dedekind-complete.

Nowadays, it is well known that there are several equivalent conditions for completeness of the reals. One of them is the least-upper-bound-property. We show that Dedekind-completeness is equivalent to the least-upper-bound-property.

**Theorem 3.2.2.** Suppose that $(S, <_S)$ is a dense linearly ordered set without end-points. The set $S$ is Dedekind-complete if and only if $S$ has the least upper bound property.

*Proof.* Assume that $S$ is Dedekind-complete, and that $A$ is a nonempty subset of $S$ bounded above. Define subsets $X, Y$ of $S$ as follows:

$$\begin{aligned} X &= \{x \in S \mid x \text{ is not an upper bound of } A\} \\ &= \{x \in S \mid x <_S a \text{ for some } a \in A\}, \\ Y &= \{y \in S \mid y \text{ is an upper bound of } A\} \\ &= \{y \in S \mid a \leq_S y \text{ for all } a \in A\}. \end{aligned}$$

Then $(X, Y)$ is a comparable partition of $S$. Since $S$ is Dedekind complete, $X$ has the greatest element or $Y$ has the least element. If $X$ has the greatest element $g$, then since $g \in X$, $g <_S a$ for some $a \in A$. Because $S$ is dense, there is $z \in S$ such that $g <_S z$ and $z <_S a$. Since $z <_S a$, we see that $z \in X$. Then for $z$, the element $g$ is not the greatest element in $X$. Therefore $Y$ has the least element. It is exactly the least upper bound of $A$. Thus $A$ has the least-upper-bound-property.

Assume that $S$ has the least-upper-bound-property. Let $(A, B)$ be a comparable partition of $S$. Since $A$ is bounded above (by every element of $B$), the set $A$ has the least upper bound in $S$, say it $\alpha$. Because every element of $B$ is an upper bound of $A$ and $\alpha$ is the least upper bound of $A$, we know that $\alpha \leq_S b$ for every $b \in B$. Thus if $\alpha \in B$, then $\alpha$ is the least element of $B$. If $\alpha \in A$, then since $\alpha$ is (the least) upper bound of $A$, we obtain that $\alpha$ is the greatest element of $A$. Thus $S$ is Dedekind-complete. $\square$

# Chapter 4

# Construction of the reals 1

## 4.1 Existence of the reals

In the previous section, we characterize a straight line. The corresponding algebraic structure to a straight line is called the reals. In this section, we construct the reals.

Let $R$ denote the set of all comparable partitions $(A, B)$ of $\mathbb{Q}$ such that $A$ does not have the greatest element. Roughly speaking, $(A, B)$ corresponds to a point in a straight line between $A$ and $B$, or a point not less than every points of $A$ and not greater than every points of $B$. We define equality and inequality in $R$. Two elements of $R$ equals in $R$ if two elements are identical. And for two elements $(A_1, A_2)$, $(B_1, B_2)$ in $R$, we define a binary relation $(A_1, A_2) <_R (B_1, B_2)$ if there is an element $q \in \mathbb{Q}$ such that $q \in A_2 \cap B_1$. We shall show that this $R$ is the reals. And we define $\iota : \mathbb{Q} \to R$ by $\iota(q) = ((-\infty, q)_{\mathbb{Q}}, [q, \infty)_{\mathbb{Q}})$. It is natural injection from $\mathbb{Q}$ into $R$.

**Theorem 4.1.1.** *$R$ is a dense linearly ordered set without endpoints. And $\iota : \mathbb{Q} \to R$ is an order-preserving map such that $\iota(\mathbb{Q})$ is dense in $R$.*

*Proof.* If $q_1 < q_2$ for $q_1$, $q_2$ in $\mathbb{Q}$, then $q_1 \in [q_1, \infty)_{\mathbb{Q}} \cap (-\infty, q_2)_{\mathbb{Q}}$. Thus $\iota(q_1) <_R \iota(q_2)$. If $(A_1, A_2) <_R (B_1, B_2)$, then there is $x \in \mathbb{Q}$ such that $x \in A_2 \cap B_1$. Since $B_1$ does not have the greatest element, there exists $y \in \mathbb{Q}$ such that $y \in B_1$ and $x < y$.

Let $z$ denote $(x+y)/2$, *i.e.*, $x < z < y$. Hence $x \in A_2 \cap (-\infty, z)_{\mathbb{Q}}$, which means that $(A_1, A_2) <_R \iota(z)$. Similarly $y \in [z, \infty)_{\mathbb{Q}} \cap B_1$, which means that $\iota(z) <_R (B_1, B_2)$. Thus $\iota$ is an order-preserving map such that $\iota(\mathbb{Q})$ is dense in $R$.

Assume that $(A_1, A_2) <_R (A_1, A_2)$ for some element in $R$. Then there exists $q \in A_2 \cap A_1$. Since $(A_1, A_2)$ is a comparable partition, we know that $A_1 \cap A_2$ is empty, which leads a contradiction. Thus $<_R$ is irreflexive. Assume that $(A_1, A_2) <_R (B_1, B_2)$ and $(B_1, B_2) <_R (C_1, C_2)$. Then there exists $p \in A_2 \cap B_1$ and $q \in B_2 \cap C_1$. Since $p \in B_1$, $q \in B_2$, and $(B_1, B_2)$ is a comparable partition of $\mathbb{Q}$, we obtain that $p < q$. And $p \in A_2$ and $p < q$ implies that $q \in A_2$. Because $q \in A_2 \cap C_1$, it follows that $(A_1, A_2) <_R (C_1, C_2)$, *i.e.*, $<_R$ is transitive. Thus $<_R$ is a strict order on $R$.

Assume that two elements $(A_1, A_2)$ and $(B_1, B_2)$ of $R$ are not identical, *i.e.*, $A_1 \neq B_1$. Thus there exists $q \in \mathbb{Q}$ such that ($q \in A_1$ and $q \notin B_1$) or ($q \notin A_1$ and $q \in B_1$). If $q \in A_1$ and $q \notin B_1$, then $q \in B_2$. Hence $q \in B_2 \cap A_1$, which implies that $(B_1, B_2) <_R (A_1, A_2)$. If $q \notin A_1$ and $q \in B_1$, then by the same way, we know that $(A_1, A_2) <_R (B_1, B_2)$. Hence every two elements of $R$ are comparable. Thus $(R, <_R)$ is a linearly ordered set.

Choose arbitrary element $(A_1, A_2)$ of $R$. Because $A_1$ and $A_2$ are nonempty, there exist $x \in A_1$ and $y \in A_2$. Then $x \in [x, \infty)_{\mathbb{Q}} \cap A_1$, which means that $\iota(x) <_R (A_1, A_2)$. And from $y \in A_2$, we know that $y \in A_2 \cap (-\infty, y+1)_{\mathbb{Q}}$, which means that $(A_1, A_2) <_R \iota(y+1)$. Therefore $\iota(x) <_R (A_1, A_2) <_R \iota(y+1)$. Thus $R$ is without endpoints. We already know that $\iota(\mathbb{Q})$ is dense in $R$, which implies that $R$ is dense directly. $\square$

**Theorem 4.1.2.** Let $S$ be a dense linearly ordered set without endpoints and $\iota : \mathbb{Q} \to S$ be an order-preserving map such that $\iota(\mathbb{Q})$ is dense in $S$. For a comparable partition $(S_1, S_2)$ of $S$, we define two subsets $\mathbb{Q}_1$ and $\mathbb{Q}_2$ of $\mathbb{Q}$ as follows :

$$\mathbb{Q}_1 := \{q \in \mathbb{Q} \mid \iota(q) \in S_1\}, \quad \mathbb{Q}_2 := \{q \in \mathbb{Q} \mid \iota(q) \in S_2\}.$$

Then $(\mathbb{Q}_1, \mathbb{Q}_2)$ is a comparable partition of $\mathbb{Q}$. Additionally, if for each comparable partition $(S_1, S_2)$ of $S$ there exists corresponding $m$ in $S$ such that $\iota(q_1) \leq_S m \leq_S \iota(q_2)$ for all $q_1 \in \mathbb{Q}_1$ and $q_2 \in \mathbb{Q}_2$, then $S$ is Dedekind-complete.

*Proof.* Since $(S_1, S_2)$ is a comparable partition of $S$, we know that both $S_1$ and $S_2$ are nonempty, and $s_1 <_S s_2$ for every $s_1 \in S_1$ and $s_2 \in S_2$, and $S_1 \cup S_2 = S$.

Because $S_1$ is nonempty, there is an element $s_1$ in $S_1$. And because $S$ does not have endpoints, there is an element $s'$ of $S$ such that $s' <_S s_1$. By Lemma 3.2.1, we see that $s' \in S_1$. Since $\iota(\mathbb{Q})$ is dense in $S$, there is $q_1$ in $\mathbb{Q}$ such that $s' <_S \iota(q_1) <_S s_1$. By Lemma 3.2.1, we see that $\iota(q_1) \in S_1$, which implies that $\mathbb{Q}_1$ is nonempty. In the same way, we can prove that $\mathbb{Q}_2$ is nonempty.

Choose arbitrary $q_1$ in $\mathbb{Q}_1$ and $q_2$ in $\mathbb{Q}_2$. Then $\iota(q_1) \in S_1$ and $\iota(q_2) \in S_2$. Since $(S_1, S_2)$ is a comparable partition of $S$, we know that $\iota(q_1) <_S \iota(q_2)$. For the order between $q_1$ and $q_2$, there are three possibilities : $q_1 < q_2$ or $q_1 = q_2$ or $q_2 < q_1$. Because $\iota$ is an order-preserving map, each cases implies that $\iota(q_1) <_S \iota(q_2)$ or $\iota(q_1) =_S \iota(q_2)$ or $\iota(q_2) <_S \iota(q_1)$, respectively. Since $S$ is a linearly ordered set, the only non-contradictable case is $q_1 < q_2$. Hence we show that $q_1 < q_2$ for every $q_1 \in \mathbb{Q}_1$ and $q_2 \in \mathbb{Q}_2$. And this shows that $\mathbb{Q}_1 \cap \mathbb{Q}_2 = \emptyset$.

We know that $\iota(q)$ is in $S$ for every $q \in \mathbb{Q}$. Since $S = S_1 \cup S_2$, we obtain that $\iota(q) \in S_1$ or $\iota(q) \in S_2$ for every $q \in \mathbb{Q}$, which means that $q \in \mathbb{Q}_1$ or $q \in \mathbb{Q}_2$ for every $q \in \mathbb{Q}$. Thus $\mathbb{Q}_1 \cup \mathbb{Q}_2 = \mathbb{Q}$. Therefore $(\mathbb{Q}_1, \mathbb{Q}_2)$ is a comparable partition of $\mathbb{Q}$.

Assume that for each comparable partition $(S_1, S_2)$ of $S$, there exists corresponding $m$ in $S$ such that $\iota(q_1) \leq_S m \leq_S \iota(q_2)$ for all $q_1 \in \mathbb{Q}_1$ and $q_2 \in \mathbb{Q}_2$. Since $S$ is a linearly ordered set, for each $x$ in $S$ such that $x \neq_S m$, we obtain that $x <_S m$ or $m <_S x$. Suppose that $x <_S m$. Because $\iota(\mathbb{Q})$ is dense in $S$, there is $q \in \mathbb{Q}$ such that $x <_S \iota(q) <_S m$. If $q \in \mathbb{Q}_2$, then $m \leq_S \iota(q)$ by assumption, which leads a contradiction. Hence $q \in \mathbb{Q}_1$, and so $\iota(q)$ is in $S_1$. By Lemma 3.2.1 and $x <_S \iota(q)$, we obtain that $x$ is in $S_1$. Thus if $x <_S m$, then $x$ is in $S_1$. In the similar way, we can show that if $m <_S x$ then $x$ is in $S_2$. In summary,

$$\begin{cases} x <_S m \implies x \in S_1, \\ x =_S m \implies x \in S_1 \text{ or } x \in S_2, \\ m <_S x \implies x \in S_2. \end{cases}$$

Thus, every element of $S_1$ is less than or equal to $m$, and every element of $S_2$ is greater than or equal to $m$. So if $m$ belongs to $S_1$, then $m$ is the greatest element of $S_1$; and if $m$ belongs to $S_2$, then $m$ is the least element of $S_2$. Therefore $S$ is Dedekind-complete. $\square$

**Theorem 4.1.3.** $R$ is Dedekind-complete.

*Proof.* Recall that $R$ is the set of all comparable partitions $(A, B)$ of $\mathbb{Q}$ such that $A$ does not have the greatest element. Let $(R_1, R_2)$ be an arbitrary comparable partition of $R$. We define two subsets $\mathbb{Q}_1$ and $\mathbb{Q}_2$ of $\mathbb{Q}$ as follows :

$$\mathbb{Q}_1 := \{q \in \mathbb{Q} \mid \iota(q) \in R_1\}, \quad \mathbb{Q}_2 := \{q \in \mathbb{Q} \mid \iota(q) \in R_2\}.$$

Then by Theorem 4.1.2, $(\mathbb{Q}_1, \mathbb{Q}_2)$ is a comparable partition of $\mathbb{Q}$.

If $\mathbb{Q}_1$ has the greatest element, say it $a$, then since $(\mathbb{Q}_1, \mathbb{Q}_2)$ is a comparable partition of $\mathbb{Q}$, it follows that $q_1 \leq a < q_2$ for every $q_1 \in \mathbb{Q}_1$ and $q_2 \in \mathbb{Q}_2$. Because $\iota$ is order-preserving, we obtain that $\iota(q_1) \leq_R \iota(a) <_R \iota(q_2)$ for every $q_1 \in \mathbb{Q}_1$ and $q_2 \in \mathbb{Q}_2$.

If $\mathbb{Q}_1$ does not have the greatest element, then since $(\mathbb{Q}_1, \mathbb{Q}_2)$ is a comparable partition of $\mathbb{Q}$, we obtain that $(\mathbb{Q}_1, \mathbb{Q}_2) \in R$. Let us denote $(\mathbb{Q}_1, \mathbb{Q}_2)$ by $m$. For each $q_1 \in \mathbb{Q}_1$, there exists $q_1' \in \mathbb{Q}_1$ such that $q_1 < q_1'$. Then $q_1' \in [q_1, \infty)_{\mathbb{Q}} \cap \mathbb{Q}_1$. Hence $\iota(q_1) <_R m$. And for every $q_2 \in \mathbb{Q}_2$, from $\mathbb{Q}_2 \cap \mathbb{Q}_1 = \emptyset$, we know that $[q_2, \infty)_{\mathbb{Q}} \cap \mathbb{Q}_1 = \emptyset$. It follows that (not $\iota(q_2) <_R m$) for every $q_2 \in \mathbb{Q}_2$. Hence $m \leq_R \iota(q_2)$ for every $q \in \mathbb{Q}_2$. Thus $\iota(q_1) <_R m \leq_R \iota(q_2)$ for every $q_1 \in \mathbb{Q}_1$ and $q_2 \in \mathbb{Q}_2$.

Therefore $R$ is Dedekind-complete by Theorem 4.1.2. $\square$

## 4.2 Uniqueness of the reals

In the previous section, we show the existence of the reals $R$, or equivalently, an algebraic structure corresponding to a straight line. In this section, we shall show the uniqueness of the reals (up to isomorphism).

**Theorem 4.2.1.** Suppose that $(S, <_S)$ is a dense linearly ordered set without endpoints, and that there is an order-preserving map $f : \mathbb{Q} \to S$ such that $f(\mathbb{Q})$ is dense in $S$, and that $(S, <_S)$ is Dedekind-complete. Then there exists a bijective order-preserving map $\overline{f} : R \to S$ which extends $f$, *i.e.*, $f(q) = \overline{f}(\iota(q))$ for all $q \in \mathbb{Q}$. Moreover, this extension $\overline{f}$ is unique.

$$\begin{tikzcd} \mathbb{Q} \arrow[d, "\iota"'] \arrow[dr, "f"] & \\ R \arrow[r, dashed, "\overline{f}"'] & S \end{tikzcd}$$

*Proof.* First, we show the uniqueness of this extension. Assume that $\overline{f}_1$ and $\overline{f}_2$ are two distinct extensions. Then there exists $r \in R$ such that $\overline{f}_1(r) \neq_S \overline{f}_2(r)$. Without loss of generality, assume that $\overline{f}_1(r) <_S \overline{f}_2(r)$. Since $f(\mathbb{Q})$ is dense in $S$, there is $q \in \mathbb{Q}$ such that $\overline{f}_1(r) <_S f(q) <_S \overline{f}_2(r)$. Thus $\overline{f}_1(r) <_S \overline{f}_1(\iota(q))$ and $\overline{f}_2(\iota(q)) <_S \overline{f}_2(r)$. It implies that $r <_R \iota(q)$ and $\iota(q) <_R r$. This leads a contradiction. Thus if there is an extension, it is unique.

We shall show that for every $(A, B) \in R$, there is unique $p \in S$ such that $f(a) <_S p \leq_S f(b)$ for all $a \in A$ and $b \in B$, *i.e.*, $f(a) <_S p$ for all $a \in A$ and $p \leq_S f(b)$ for all $b \in B$. We define two subsets $C, D$ of $S$ as follows:

$$C = \{c \in S \mid c \leq_S f(a) \text{ for some } a \in A\},$$
$$D = \{d \in S \mid f(b) <_S d \text{ for some } b \in B\}.$$

If $x \in C$, then there is $a \in A$ such that $x \leq_S f(a)$. Because $A$ does not have the greatest element, there is $a' \in A$ such that $a < a'$. Since $f$ is order preserving, we see that $f(a) <_S f(a')$. Thus $x <_S f(a')$. And by definition of $C$, we obtain that $f(a') \in C$. Therefore $C$ does not have the greatest element.

If $y \in D$, then there is $b \in B$ such that $f(b) <_S y$. Since $S$ is dense, there is $y' \in S$ such that $f(b) <_S y' <_S y$. Hence $y' \in D$ and $y' <_S y$. Thus $D$ does not have the least element.

If there is no $p \in S$ such that $f(a) <_S p \leq_S f(b)$ for all $a \in A$ and $b \in B$, then $C \cup D = S$. Thus we know that $(C, D)$ is a comparable partition of $S$. Since $S$ is Dedekind complete, $C$ has the greatest element or $D$ has the least element. It contradicts to our previous argument. Thus there exists $p \in S$ such that $f(a) <_S p \leq_S f(b)$ for all $a \in A$ and $b \in B$. If such $p$ is not unique, assume that there are two such elements $p_1, p_2$ in $S$ with $p_1 <_S p_2$. Since $f(\mathbb{Q})$ is dense in $S$, there is $q \in \mathbb{Q}$ such that $p_1 <_S f(q) <_S p_2$. Since $(A, B)$ is a comparable partition of $\mathbb{Q}$, we see that $q \in A$ or $q \in B$. If $q \in A$, then $p_1 <_S f(q)$ contradicts that $f(a) <_S p_1$ for all $a \in A$. If $q \in B$, then $f(q) <_S p_2$ contradicts that $p_2 \leq_S f(b)$ for all $b \in B$. Thus such $p$ is unique.

To define $\overline{f}$, for each $(A, B) \in R$, we assign $p$ in $S$ to $(A, B)$ satisfying that $f(a) <_S p \leq_S f(b)$ for all $a \in A$ and $b \in B$. By our previous argument, $\overline{f}$ is well defined. For each $q \in \mathbb{Q}$, we know that $\iota(q) = ((-\infty, q)_{\mathbb{Q}}, [q, \infty)_{\mathbb{Q}})$. Hence $\overline{f}(\iota(q))$ is equal to $p$ satisfying that $f(a) <_S p \leq_S f(b)$ for all $a \in (-\infty, q)_{\mathbb{Q}}$ and $b \in [q, \infty)_{\mathbb{Q}}$. If $p = f(q)$, then the condition is satisfied. By the uniqueness of $p$, we conclude that $\overline{f}(\iota(q)) = f(q)$.

For two distinct $(A_1, B_1), (A_2, B_2) \in R$, assume that $(A_1, B_1) <_R (A_2, B_2)$. Then there is $q \in \mathbb{Q}$ such that $q \in B_1 \cap A_2$. Let $p_i$ be $\overline{f}((A_i, B_i))$ for $i = 1, 2$. Then $f(a) <_S p_1 \leq_S f(b)$ for all $(a, b) \in A_1 \times B_1$ and $f(a) <_S p_2 \leq_S f(b)$ for all $(a, b) \in A_2 \times B_2$.

We derive the inequalities $p_1 \leq_S f(q)$ and $f(q) <_S p_2$. Hence $p_1 <_S p_2$. Thus $\overline{f}$ is an order preserving map. The fact that $\overline{f}$ is injective is also proved.

The only remaining goal is to show that $\overline{f}$ is surjective. For each $p \in S$, define $A_p$ and $B_p$ by $A_p = \{q \in \mathbb{Q} \mid f(q) <_S p\}$ and $B_p = \{q \in \mathbb{Q} \mid p \leq_S f(q)\}$. Then

$(A_p, B_p)$ is a comparable partition of $\mathbb{Q}$. And if $q_1 \in A_p$, *i.e.*, if $f(q_1) <_S p$, then since $f(\mathbb{Q})$ is dense in $S$, there is $q_2 \in \mathbb{Q}$ such that $f(q_1) <_S f(q_2) <_S p$. Thus $q_2 \in A_p$ and $q_1 <_S q_2$. Hence $A_p$ does not have the greatest element. Thus $(A_p, B_p)$ belongs to $R$. And by definition of $A_p$ and $B_p$, the condition $f(a) <_S p \leq_S f(b)$ for all $a \in A_p$ and $b \in B_p$ is satisfied, which implies that $\overline{f}((A_p, B_p)) = p$. Thus $\overline{f}$ is surjective. Therefore $\overline{f}$ is a bijective order preserving map satisfying that $\overline{f}(\iota(q)) = f(q)$ for all $q \in \mathbb{Q}$. $\square$

# Chapter 5

# Coq proof checking 1

In this chapter, we overview how we use Coq to construct the reals. In the following Coq codes, `Lemma` and `Theorem` and `Example` are all things that we need to prove. Due to a lake of space, we omit all proof codes in this paper, instead upload them in the Internet. [1]

In Coq codes, we first put the excluded-middle property by axiom because there are some occasions necessarily to use it. And then we make and prove some logical lemmas. *all_ssreflect* is a library that contains some useful tactics. *QArith* is a library that contains definitions and lemmas related to $\mathbb{Q}$. And $\vee$ and $\wedge$ are logical connectives that imply 'or' and 'and', respectively.

`From` *mathcomp* `Require Import` *all_ssreflect*.

`Require Import` *QArith*.

`Axiom` *excluded_middle* :
$\forall P$ : `Prop`, $P \vee$ *not* $P$.

`Lemma` *and_or_distr* ($A\ B\ C$ : `Prop`) :
$(A \wedge B) \vee C \leftrightarrow (A \vee C) \wedge (B \vee C)$.

`Lemma` *or_and_distr* ($A\ B\ C$ : `Prop`) :

[1] https://github.com/DoyunNam/Coq_Reals/blob/main/Coq_Reals.v

$(A \vee B) \wedge C \leftrightarrow (A \wedge C) \vee (B \wedge C)$.

Lemma *and_comm* ($P$ $Q$ : Prop) :

$P \wedge Q \leftrightarrow Q \wedge P$.

Lemma *or_trans* ($A$ : Prop) ($B$ : Prop) ($C$ : Prop) :

$(A \vee B) \vee C \leftrightarrow A \vee (B \vee C)$.

Lemma *contrapositive* ($P$ $Q$ : Prop) :

$(P \rightarrow Q) \rightarrow (\mathit{not}\ Q \rightarrow \mathit{not}\ P)$.

Lemma *imply_not_or* ($P$ $Q$ : Prop) :

$(P \rightarrow Q) \leftrightarrow (\mathit{not}\ P \vee Q)$.

Lemma *not_not_equiv* ($P$ : Prop) :

$P \leftrightarrow (\mathit{not}\ (\mathit{not}\ P))$.

Lemma *all_prop* ($S$ : Set) ($P$ : $S \rightarrow$ Prop) :

$(\forall\, x : S, (P\ x)) \leftrightarrow \mathit{not}\ (\exists\, x : S, \mathit{not}\ (P\ x))$.

Lemma *not_all_prop* ($S$ : Set) ($P$ : $S \rightarrow$ Prop) :

$\mathit{not}\ (\forall\, x : S, (P\ x)) \leftrightarrow \exists\, x : S, \mathit{not}\ (P\ x)$.

Lemma *not_exists_prop* ($S$ : Set) ($P$ : $S \rightarrow$ Prop) :

$\mathit{not}\ (\exists\, x : S, (P\ x)) \leftrightarrow \forall\, x : S, \mathit{not}\ (P\ x)$.

Lemma *not_imply_equiv* ($P$ $Q$ : Prop) :

$\mathit{not}\ (P \rightarrow Q) \leftrightarrow \mathit{not}\ (\mathit{not}\ P \vee Q)$.

Lemma *not_or* ($P$ $Q$ : Prop) :

$\mathit{not}\ (P \vee Q) \leftrightarrow \mathit{not}\ P \wedge \mathit{not}\ Q$.

Lemma *equiv_not_equiv1* ($P$ $Q$ : Prop) :

$(P \leftrightarrow Q) \rightarrow (\mathit{not}\ P \leftrightarrow \mathit{not}\ Q)$.

Lemma *equiv_not_equiv2* ($P$ $Q$ : Prop) :

$(\mathit{not}\ P \leftrightarrow \mathit{not}\ Q) \rightarrow (P \leftrightarrow Q)$.

Lemma *equiv_not_equiv* ($P$ $Q$ : Prop) :

$(P \leftrightarrow Q) \leftrightarrow (not\ P \leftrightarrow not\ Q)$.

`Lemma` *not_and* $(P\ Q :$ `Prop`$)$ :

$not\ (P \land Q) \leftrightarrow not\ P \lor not\ Q$.

`Lemma` *all_or_pro_distr* $(S :$ `Set`$)$ $(P\ Q : S \rightarrow$ `Prop`$)$ :

$(\forall\ x : S, (P\ x \lor Q\ x)) \rightarrow$

$(\forall\ x : S, P\ x) \lor (\exists\ x : S, Q\ x)$.

Like these logical lemmas, if necessary, we make lemmas and prove them; and use them in the course of proving some theorems. Since Coq library does not contain every logically true statement, in many times, we need to define lemmas and prove them. For example,

`Lemma` *Zlt_le_0* $(n : Z)$ :

$(0 < n)\%Z \rightarrow (0 \leq n)\%Z$.

`Lemma` *Qlt_le* $(a\ b : Q)$ :

$a < b \rightarrow a \leq b$.

`Lemma` *Qlt_plus_transpose* $(a\ b\ c : Q)$ :

$a - b < c \leftrightarrow a < b + c$.

These three lemmas are trivial in natural language. However, in Coq, we need to prove them if we want to use them and they are not in the Coq library. For briefness, we shall omit obvious lemmas.

And we define relation, reflexive, irreflexive, and so on. For general situations, we define *compatible_eq_lt* : if $w \sim_X x$, $y \sim_X z$, and $w <_X y$, then $x <_X z$, where $\sim_X$ is an equivalence relation on $X$.

`Definition` *relation* $(X :$ `Set`$)$ :=

$X \rightarrow X \rightarrow$ `Prop`.

`Definition` *reflexive* $\{X :$ `Set`$\}$ $(R : relation\ X)$ :=

$\forall\ a : X, (R\ a\ a)$.

`Definition` *irreflexive* $\{X :$ `Set`$\}$ $(R : relation\ X)$ :=

$\forall\, a : X,\ not\ (R\ a\ a).$

```
Definition symmetric {X : Set} (R : relation X) :=
∀ a b : X, (R a b) → (R b a).

Definition antisymmetric {X : Set} (R : relation X) :=
∀ a b : X, (R a b) → (R b a) → a = b.

Definition asymmetric {X : Set} (R : relation X) :=
∀ a b : X, (R a b) → not (R b a).

Definition transitive {X : Set} (R : relation X) :=
∀ a b c : X, (R a b) → (R b c) → (R a c).

Definition strict_order {X : Set} (R : relation X) :=
(irreflexive R) ∧ (asymmetric R) ∧ (transitive R).

Definition equivalence {X : Set} (R : relation X) :=
(reflexive R) ∧ (symmetric R) ∧ (transitive R).

Definition compatible_eq_lt {X : Set} (Xlt Xeq : relation X) :=
∀ w x y z : X, (Xeq w x) → (Xeq y z) → (Xlt w y) → (Xlt x z).

Definition total_order {X : Set} (Xlt Xeq : relation X) :=
∀ x y : X, (Xlt x y) ∨ (Xeq x y) ∨ (Xlt y x).

Definition without_endpoints {X : Set} (Xlt : relation X) :=
∀ x : X, (∃ y, Xlt y x) ∧ (∃ z, Xlt x z).

Definition dense {X : Set} (Xlt : relation X) :=
∀ x y : X, (Xlt x y) →
∃ z : X, (Xlt x z) ∧ (Xlt z y).

Record dlos := mkdlos {
  X : Set;
  Xlt : relation X;
  Xeq : relation X;
```

*eq* : *equivalence Xeq*;

*st* : *strict_order Xlt*;

*cp* : *compatible_eq_lt Xlt Xeq*;

*to* : *total_order Xlt Xeq*;

*den* : *dense Xlt*;

*we* : *without_endpoints Xlt*;

}.

And in the above, we make a Record structure *dlos*. The Record structure *dlos* is similar to an ordered 9-tuples (*X, Xlt, ..., den, we*). Each *X*, *Xlt*, *Xeq*, ... is like a coordinate function. If *S* is a *dlos*, then *X S* is a set, and *Xlt S* is a relation defined on *X S*, and so on.

If *S* is a *dlos*, then *Xeq S* is a relation on *X S*. And *eq S* implies that *equivalence Xeq S* is true. Hence *Xeq S* is an equivalence relation on *X S*. Similarly, *Xlt S* is a strict order on *X S*.

In the below, we make an axiom whose name is *function*.

```
Axiom function :
∀ S : dlos, ∀ f : (X S) → bool,
∀ p q : X S, (Xeq S) p q → f p = f q.
```

```
Lemma Xlt_not (S : dlos) (x y : X S) :
Xlt S x y → not (Xlt S y x ∨ Xeq S y x).
```

```
Example Q_equivalence :
equivalence Qeq.
```

```
Example Q_strict_order :
strict_order Qlt.
```

```
Example Q_compatible_eq_lt :
compatible_eq_lt Qlt Qeq.
```

```
Example Q_total_order :
```

*total_order Qlt Qeq*.

```
Example
```
*Q_dense* :

*dense Qlt*.

`Example` *Q_without_endpoints* :

*without_endpoints Qlt*.

`Definition` *Q_dlos* :=

{|

*X* := *Q*;

*Xlt* := *Qlt*;

*Xeq* := *Qeq*;

*eq* := *Q_equivalence*;

*st* := *Q_strict_order*;

*cp* := *Q_compatible_eq_lt*;

*to* := *Q_total_order*;

*den* := *Q_dense*;

*we* := *Q_without_endpoints*

|}.

In the above, we proved that $\mathbb{Q}$ is a dense linearly ordered set without endpoints.

And for a comparable partition $(A, B)$ of some dense linearly ordered set $X$, there is a corresponding function $f : X \to \{0, 1\}$ such that $f(x) = 0$ if $x \in A$ and $f(x) = 1$ if $x \in B$. ( This function $f$ is equal to the characteristic function $\chi_B$). Since both $A$ and $B$ are nonempty, the map $f$ is not a constant function. And since $(A, B)$ is a comparable partition, it follows that $f$ is monotonically increasing. Thus each comparable partition corresponds to a non-constant, monotonically increasing function from $X$ into $\{0, 1\}$. And we can easily prove that this correspondence is bijective.

In Coq, *bool* is a set {*false, true*}. We define $f(x) =$ *false* if $x \in A$ and $f(x) =$ *true* if $x \in B$. Then we may understand the following three definitions.

`Definition` *mono_inc* $\{S : dlos\}$ $(f : (X\ S) \to bool) :=$

$\forall\, p\ q : X\ S,\ (Xlt\ S)\ p\ q \to$

$(f\ p = false \wedge f\ q = false) \vee$

$(f\ p = false \wedge f\ q = true) \vee$

$(f\ p = true \wedge f\ q = true).$

`Definition` *not_const* $\{S : dlos\}$ $(f : (X\ S) \to bool) :=$

$(\exists\, p : X\ S,\ (f\ p) = false) \wedge$

$(\exists\, q : X\ S,\ (f\ q) = true).$

`Definition` *comparable_partition* $\{S : dlos\}$ $(f : (X\ S) \to bool) :=$

$(mono\_inc\ f) \wedge (not\_const\ f).$

And *not_havemax* means that there is no greatest element of $f^{-1}(false)$, and *havemax* means that there is a greatest element of $f^{-1}(false)$. Similarly, *not_havemin* implies that there is no least element of $f^{-1}(true)$, and *havemin* implies that there is a least element of $f^{-1}(true)$.

`Definition` *not_havemax* $\{S : dlos\}$ $(f : (X\ S) \to bool) :=$

$\forall\, p : X\ S,\ (f\ p) = false$

$\to (\exists\, q : X\ S,\ (Xlt\ S)\ p\ q \wedge (f\ q) = false).$

`Definition` *havemax* $\{S : dlos\}$ $(f : (X\ S) \to bool) :=$

$(\exists\, x : X\ S,$

$f\ x = false \wedge (\forall\, y : X\ S,\ (Xlt\ S)\ x\ y \to f\ y = true)).$

`Definition` *not_havemin* $\{S : dlos\}$ $(f : (X\ S) \to bool) :=$

$\forall\, q : X\ S,\ (f\ q) = true$

$\to (\exists\, p : X\ S,\ (Xlt\ S)\ p\ q \wedge (f\ p) = true).$

`Definition` *havemin* $\{S : dlos\}$ $(f : (X\ S) \to bool) :=$

$(\exists\, y : X\ S,$

$f\ y = true \wedge (\forall\, x : X\ S,\ (Xlt\ S)\ x\ y \to f\ x = false)).$

And as we know, Dedekind-completeness is defined as follows : for every comparable partition $(A, B)$, $A$ has the greatest element or $B$ has the least element. And to construct $R$, we define *CondR*.

*X Q_dlos* is equal to *Q* as a set. Hence *f : X Q_dlos* $\rightarrow$ *bool* is equal to *f : Q* $\rightarrow$ *bool*. And we defined *comparable_partition f* by (*mono_inc f*) $\wedge$ (*not_const f*) before. Thus *CondR f* in Coq corresponds to a comparable partition $(A, B)$ of $\mathbb{Q}$ such that $A$ does not have the greatest element.

`Definition` *Dedekind_complete* (*S* : *dlos*) :=
$\forall$ (*f* : (*X S*) $\rightarrow$ *bool*),
(*comparable_partition f*) $\rightarrow$ (*havemax f*) $\vee$ (*havemin f*).

`Definition` *CondR* (*f* : (*X Q_dlos*) $\rightarrow$ *bool*) :=
*mono_inc f* $\wedge$ *not_const f* $\wedge$ *not_havemax f*.

`Lemma` *havemax_total* {*S* : *dlos*} (*f* : (*X S*) $\rightarrow$ *bool*) :
*havemax f* $\leftrightarrow$ *not* (*not_havemax f*).

`Lemma` *havemin_total* {*S* : *dlos*} (*f* : (*X S*) $\rightarrow$ *bool*) :
*havemin f* $\leftrightarrow$ *not* (*not_havemin f*).

`Lemma` *mono_inc'* {*S* : *dlos*} (*f* : (*X S*) $\rightarrow$ *bool*) :
(*mono_inc f* $\leftrightarrow$
$\forall$ *p q* : *X S*, (*f p* = *false* $\rightarrow$ *f q* = *true* $\rightarrow$ (*Xlt S*) *p q*)).

*less_part* and *greater_part* corresponds to Lemma 3.2.1.

`Lemma` *less_part* {*S* : *dlos*} (*f* : (*X S*) $\rightarrow$ *bool*) (*p q* : *X S*) :
*mono_inc f* $\rightarrow$ (*Xlt S*) *p q* $\rightarrow$ *f q* = *false* $\rightarrow$ *f p* = *false*.
`Lemma` *greater_part* {*S* : *dlos*} (*f* : (*X S*) $\rightarrow$ *bool*) (*p q* : *X S*) :
*mono_inc f* $\rightarrow$ (*Xlt S*) *p q* $\rightarrow$ *f p* = *true* $\rightarrow$ *f q* = *true*.
`Lemma` *classify_comp_part* {*S* : *dlos*} (*f* : (*X S*) $\rightarrow$ *bool*) :
(*comparable_partition f*) $\rightarrow$
(*havemax f* $\wedge$ *not_havemin f*) $\vee$
(*not_havemax f* $\wedge$ *havemin f*) $\vee$

(*not_havemax f* $\wedge$ *not_havemin f*).

And we make $R$ as follows. And then we define *Req* and *Rlt*. Note that if we corresponds *f r1* to $(A_1, A_2)$ and *f r2* to $(B_1, B_2)$, then *(f r1) q = true* means that $q \in A_2$, and *(f r2) q = false* means that $q \in B_1$. Hence $q \in A_2 \cap B_1$, which implies that $(A_1, A_2) <_R (B_1, B_2)$. Thus *Rlt* is well defined.

`Record` *R* := *mkReal* {

  *f* : (*X Q_dlos*) $\rightarrow$ *bool*;

  *Cond* : *CondR f*;

}.

`Definition` *Req* (*r1 r2* : *R*) :=

$\forall$ *q* : *Q*, (*f r1*) *q* = (*f r2*) *q*.

`Definition` *Rlt* (*r1 r2* : *R*) :=

$\exists$ *q* : *Q*, (*f r1*) *q* = *true* $\wedge$ (*f r2*) *q* = *false*.

`Theorem` *R_equivalence* :

*equivalence Req*.

`Theorem` *R_strict_order* :

*strict_order Rlt*.

`Theorem` *R_compatible_eq_lt* :

*compatible_eq_lt Rlt Req*.

`Theorem` *R_total_order* :

*total_order Rlt Req*.

*Qle_bool* is a function of type *Q* $\rightarrow$ *Q* $\rightarrow$ *bool* defined by as follows : *Qle_bool p q* = *true* if $p \leq q$, and *Qle_bool p q* = *false* if $q < p$. As defined above, *CondR f* is a property corresponding that $A$ does not have the greatest element for a comparable partition $(A, B)$ of $\mathbb{Q}$. For each $q \in \mathbb{Q}$, we see that *Qle_bool q* is a function from *Q* into *bool*, and in the below, *inject_Q* is a structure corresponding $\iota : \mathbb{Q} \rightarrow R$ in our previous chapter.

```
Lemma CondR_Q (q : Q) :
CondR (Qle_bool q).
Definition inject_Q (q : Q) : R :=
{| f := (Qle_bool q) ;
    Cond := (CondR_Q q)
|}.
Theorem inject_Q_eq (p q : Q) :
p == q → Req (inject_Q p) (inject_Q q).
Theorem inject_Q_order_preserve (p q : Q) :
p < q → Rlt (inject_Q p) (inject_Q q).
Lemma inject_Q_order_reverse (p q : Q) :
Rlt (inject_Q p) (inject_Q q) → p < q.
Theorem inject_Q_dense (a b : R) :
(Rlt a b) →
∃ q : Q, (Rlt a (inject_Q q)) ∧ (Rlt (inject_Q q) b).
Theorem R_dense :
dense Rlt.
Theorem R_without_endpoints :
without_endpoints Rlt.
```

And so far, we show that $R$ is a dense linearly ordered set without endpoints. And in the below, we define a *dlos* structure *R_dlos* which represents $R$.

```
Definition R_dlos :=
{|
  X := R;
  Xlt := Rlt;
  Xeq := Req;
  eq := R_equivalence;
```

*st* := *R_strict_order*;

*cp* := *R_compatible_eq_lt*;

*to* := *R_total_order*;

*den* := *R_dense*;

*we* := *R_without_endpoints*

|}.

And then, we prove that *R* is Dedekind-complete. We construct a *dlos* (dense linearly ordered set without endpoints) structure *R_dlos*; and show that *inject_Q* is an order-preserving map from *Q* to *R* such that *inject_Q(Q)* is dense in *R*; and prove that *R* is Dedekind-complete. Therefore we prove the existence of the reals in Coq.

`Theorem` *R_Dedekind_complete* :
*Dedekind_complete R_dlos*.

# Chapter 6

# Construction of the reals 2

In chapter 4 we show the existence and uniqueness of the reals by hand, and in chapter 5 we prove the existence of the reals by Coq. In this chapter, we construct the reals in another way, and define addition and multiplication, and show that the reals is the Dedekind-complete ordered field. Recall the definition of an ordered field.

**Definition 6.0.1.** If $(S, <)$ is a linearly ordered set and if $(S, +, \times)$ is a field, then $S$ is called an *ordered field* if it satisfies the following conditions:

(a) For $x, y \in S$, if $0 < x$ and $0 < y$ then $0 < x \times y$

(b) For $x, y, z \in S$, if $x < y$ then $x + z < y + z$.

## 6.1 Nested intervals

This section summarizes definitions, lemmas, and theorems. We prove every lemmas and theorems by Coq in the next section.

**Definition 6.1.1.** Let $(a_n)$ and $(b_n)$ be rational sequences. If $(a_n)$ and $(b_n)$ satisfies the following properties, then the pair $((a_n), (b_n))$ is called a *nested interval*.

(a) $\exists m \in \mathbb{N},\ \forall n \in \mathbb{N},\ m \leq n \implies a_n \leq b_n$.

(b) $\forall m, n \in \mathbb{N},\ \exists p \in \mathbb{N},\ n \leq p$ and $b_p - a_p < \frac{1}{m}$.

(c) $\exists m \in \mathbb{N},\ \forall n, p \in \mathbb{N},\ m \leq n \leq p \implies a_n \leq a_p$.

(d) $\exists m \in \mathbb{N},\ \forall n, p \in \mathbb{N},\ m \leq n \leq p \implies b_p \leq b_n$.

And we denote $I$ the set of all nested intervals.

As a comparable partition $(A, B)$ of $\mathbb{Q}$ corresponds to a point in a straight line $L$ such that not less than every point of $A$ and not greater than every point of $B$, We may consider a nested interval $((a_n), (b_n))$ corresponds to a point $p$ in a straight line $L$ such that $p$ is not less than every $a_n$ and not greater than every $b_n$. The condition (a), (c), (d) of a nested interval contains common phrase $\exists m \in \mathbb{N}$, because it helps to define multiplication of two nested intervals.

**Notation.** For a nested interval $A = ((a_n), (b_n))$, We can choose $m_1, m_2, m_3$ of $\mathbb{N}$ in the condition (a), (c), (d) of Definition 6.1.1. And let $m$ be $\max\{m_1, m_2, m_3\}$. Then after $m$-th term, the sequence $(a_n)$ is increasing, $(b_n)$ is decreasing, and $a_n \leq b_n$ for each $m \leq n$. We shall use this $m$ frequently. For convenience, we denote this $m$ by $m_A$ for a nested interval $A$.

**Lemma 6.1.1.** For a nested interval $A = ((a_n), (b_n))$, the following statement is true.

$$\forall n, p \in \mathbb{N},\ m_A \leq n \leq p \implies a_n \leq b_p.$$

*Proof.* If $m_A \leq n \leq p$, then we obtain that $a_n \leq a_p$ and $a_p \leq b_p$, which implies that $a_n \leq b_p$. $\square$

**Definition 6.1.2.** For two nested intervals $A = ((a_n), (b_n))$ and $X = ((x_n), (y_n))$, we define a binary relation $<_I$ as follows :

$$A <_I X \iff \forall m \in \mathbb{N},\ \exists n \in \mathbb{N},\ m \leq n \text{ and } b_n < x_n.$$

**Theorem 6.1.2.** A binary relation $<_I$ is a strict order on $I$.

*Proof.* Let $A = ((a_n), (a'_n))$, $B = ((b_n), (b'_n))$, and $C = ((c_n), (c'_n))$ be nested intervals. Assume that $A <_I B$ and $B <_I C$. Let $m$ be $\max\{m_A, m_B, m_C\}$. Then there exists $n$ such that $m \leq n$ and $a'_n < b_n$, and exists $p$ such that $n \leq p$ and $b'_p < c_p$. Since $m \leq n \leq p$ and $m_B \leq m$, we obtain that $b_n \leq b'_p$ by Lemma 6.1.1. Hence $a'_n < c_p$. Since we know that after $m$-th term, sequence $(a'_n)$ is decreasing, and $(c_n)$ is increasing, we obtain that $a'_t < c_t$ for all $p \leq t$. Thus $A <_I C$, *i.e.*, $<_I$ is transitive.

If $A <_I A$ for a nested interval $A = ((a_n), (a'_n))$, then there is $n \in \mathbb{N}$ such that $m_A \leq n$ and $a'_n < a_n$. It contradicts to the definition of $m_A$. Thus $<_I$ is irreflexive. Hence $<_I$ is a strict order on $I$. □

**Definition 6.1.3.** For two nested intervals $A$ and $X$, we define a binary relation $=_I$ as follows : $A =_I X \iff$ (not $A <_I X$) and (not $X <_I A$).

**Theorem 6.1.3.** A binary relation $=_I$ is an equivalence relation on $I$.

*Proof.* Reflexivity and symmetry is proved trivially. Assume that $A =_I B$ and $B =_I C$. We want to show that $A =_I C$. For this, it is enough to prove that (not $A <_I B$) and (not $B <_I C$) implies (not $A <_I C$).

We set $A = ((a_n), (a'_n))$, $B = ((b_n), (b'_n))$, and $C = ((c_n), (c'_n))$. (not $A <_I B$) implies that

$$\exists m_1 \in \mathbb{N},\ \forall n \in \mathbb{N},\ m_1 \leq n \implies b_n \leq a'_n. \tag{6.1}$$

and (not $B <_I C$) implies that

$$\exists m_2 \in \mathbb{N},\ \forall n \in \mathbb{N},\ m_2 \leq n \implies c_n \leq b'_n. \tag{6.2}$$

Let $m^*$ be $\max\{m_B, m_1, m_2\}$. If $A <_I C$, then there is $p \in \mathbb{N}$ such that $m^* \leq p$ and $a'_p < c_p$. Then from (6.1), (6.2), and the definition of $m^*$, we obtain that

$$\forall n \in \mathbb{N},\ p \leq n \implies b_n \leq b_p \leq a'_p < c_p \leq b'_p \leq b'_n.$$

Since $0 < c_p - a'_p \leq b'_n - b_n$ for all $p \leq n$, the nested interval $B$ cannot satisfy the condition (b) of Definition 6.1.1, which leads a contradiction. Hence (not $A <_I C$) is true. □

In this way, we can prove them by natural language. The remaining theorems and lemmas are proved in the next chapter by using Coq. Thus we skip to prove them by natural language, and only mention them.

**Theorem 6.1.4.** Let $A, B, C, D$ be nested intervals. If $A =_I B$ and $C =_I D$ and $A <_I C$, then $B <_I D$.

**Theorem 6.1.5.** For arbitrary two nested intervals $A$ and $X$ we obtain that $A <_I X$ or $A =_I X$ or $X <_I A$, *i.e.*, the strict order $<_I$ (with $=_I$) is a total order on $I$.

**Remark.** We can easily show that only one of $A <_I X$ or $A =_I X$ or $X <_I A$ is true. By asymmetry of $<_I$, both $A <_I X$ and $X <_I A$ cannot happen at the same time. Assume that $A <_I X$ and $A =_I X$. Then by Theorem 6.1.4, for $A =_I X$, $X =_I A$, and $A <_I X$, we obtain $X <_I A$. And asymmetry of $<_I$ leads a contradiction. Thus our claim is proved.

**Definition 6.1.4.** For each rational number $q$, there is a constant sequence $(q)$ (that is a rational sequence such that every term is $q$). Then we can easily check that $((q),(q))$ is a nested interval for every $q \in \mathbb{Q}$. Let $\iota : \mathbb{Q} \to I$ denote a map which assigns $((q),(q))$ to $q$.

**Theorem 6.1.6.** For two rational numbers $p$ and $q$, if $p < q$ then $\iota(p) <_I \iota(q)$.

**Theorem 6.1.7.** If $A, B$ are two nested intervals and if $A <_I B$, then there is $q \in \mathbb{Q}$ such that $A <_I \iota(q) <_I B$; in other words, $\iota(\mathbb{Q})$ is dense in $I$.

**Theorem 6.1.8.** $I$ is dense.

**Definition 6.1.5** (Translation)**.** For each nested interval $((a_n),(b_n))$ and for every rational number $t$, we can easily show that $((a_n + t),(b_n + t))$ is also a nested interval. Let $\phi : \mathbb{Q} \times I \to I$ be a map that sends $(t,((a_n),(b_n)))$ to $((a_n + t),(b_n + t))$.

**Lemma 6.1.9.** If $t$ is a positive rational number, then $A <_I \phi(t, A)$ for every $A \in I$.

**Lemma 6.1.10.** If $t$ is a negative rational number, then $\phi(t, A) <_I A$ for every $A \in I$.

**Theorem 6.1.11.** For each nested interval $A$, there exist nested interval $B$ and $C$ such that $B <_I A <_I C$.

In summary, The set of all nested intervals $I$ (with $<_I$ and $=_I$) is a dense linearly ordered set without endpoints. And $\iota : \mathbb{Q} \to I$ is an order preserving map such that $\iota(\mathbb{Q})$ is dense in $I$.

**Theorem 6.1.12.** $I$ is Dedekind-complete.

*Proof.* Let $(I_1, I_2)$ be a comparable partition of $I$. Define two subsets $\mathbb{Q}_1$ and $\mathbb{Q}_2$ of $\mathbb{Q}$ as follows:

$$\mathbb{Q}_1 = \{q \in \mathbb{Q} \mid \iota(q) \in I_1\}, \quad \mathbb{Q}_2 = \{q \in \mathbb{Q} \mid \iota(q) \in I_2\}.$$

Then $(\mathbb{Q}_1, \mathbb{Q}_2)$ is a comparable partition of $\mathbb{Q}$ by Theorem 4.1.2. For every $n \in \mathbb{N}$, there exists unique $c_n \in \mathbb{Z}$ such that $\frac{c_n}{n} \in \mathbb{Q}_1$ and $\frac{c_n+1}{n} \in \mathbb{Q}_2$. Let $J_n$ be a closed interval $[\frac{c_n}{n}, \frac{c_n+1}{n}]$ (in $\mathbb{Q}$). Since $J_n = [\frac{2c_n}{2n}, \frac{2c_n+2}{2n}]$ and $J_{2n} = [\frac{c_{2n}}{2n}, \frac{c_{2n}+1}{2n}]$ and $\frac{c_{2n}}{2n} \in \mathbb{Q}_1$ and $\frac{c_{2n}+1}{2n} \in \mathbb{Q}_2$, it follows that $c_{2n}$ must be $2c_n$ or $2c_n + 1$. In any case, we obtain that $J_{2n} \subset J_n$ for all $n \in \mathbb{N}$. Let $a_n$ be $\frac{c_{2^n}}{2^n}$ and $b_n$ be $\frac{c_{2^n}+1}{2^n}$ for all $n \in \mathbb{N}$, *i.e.*, $[a_n, b_n] = J_{2^n}$. Since $J_{2^{n+1}} \subset J_{2^n}$, we obtain that $[a_{n+1}, b_{n+1}] \subset [a_n, b_n]$. Hence $(a_n)$ is an increasing sequence, and $(b_n)$ is a decreasing sequence. Moreover $a_n < b_n$ and $b_n - a_n = \frac{1}{2^n}$ for all $n \in \mathbb{N}$. Let us denote $((a_n), (b_n))$ by $m$. Then $m$ is a nested interval by the previous argument.

If $\mathbb{Q}_1$ has the greatest element, let $\alpha$ denote it. Then $q_1 \leq \alpha < q_2$ for all $q_1 \in \mathbb{Q}_1$ and $q_2 \in \mathbb{Q}_2$. Hence $\iota(q_1) \leq_I \iota(\alpha) < \iota(q_2)$ for all $q_1 \in \mathbb{Q}_1$ and $q_2 \in \mathbb{Q}_2$. If $\mathbb{Q}_2$ has the least element, then we can progress in the same way. Assume that $\mathbb{Q}_1$ does not have the greatest element and $\mathbb{Q}_2$ does not have the least element. For arbitrary $q_1 \in \mathbb{Q}_1$, there is $q_1' \in \mathbb{Q}_1$ such that $q_1 < q_1'$. And there is $n \in \mathbb{N}$ such that $\frac{1}{2^n} < q_1' - q_1$. Since $q_1' \in \mathbb{Q}_1$ and $b_n \in \mathbb{Q}_2$, we know that $q_1' < b_n$. Thus $b_n - a_n = \frac{1}{2^n} < q_1' - q_1 < b_n - q_1$, which implies that $q_1 < a_n$. Since $(a_n)$ is increasing, we obtain that $\iota(q_1) <_I m$. Similarly we can show that $m < \iota(q_2)$ for every $q_2 \in \mathbb{Q}_2$. Hence $I$ is Dedekind-complete by Theorem 4.1.2. $\square$

## 6.2 Addition of nested intervals

**Definition 6.2.1.** For each two nested intervals $A = ((a_n), (b_n))$ and $X = ((x_n), (y_n))$, we define a binary operation $+_I$ as follows:

$$A +_I X := ((a_n + x_n), (b_n + y_n)).$$

**Theorem 6.2.1.** If $A$ and $X$ are nested intervals, then so is $A +_I X$; thus $+_I$ is a binary operation on $I$.

**Theorem 6.2.2.** For $p, q \in \mathbb{Q}$, we obtain that $\iota(p + q) =_I \iota(p) +_I \iota(q)$.

**Theorem 6.2.3.** For each $A, B, C, D \in I$, if $A =_I B$ and $C =_I D$ then $A +_I C =_I B +_I D$.

**Theorem 6.2.4.** $(I, +_I)$ is commutative, *i.e.*, $A +_I B =_I B +_I A$ for every $A, B \in I$.

**Theorem 6.2.5.** $(I, +_I)$ is associative, *i.e.*, $(A +_I B) +_I C =_I A +_I (B +_I C)$ for every $A, B, C \in I$.

**Definition 6.2.2.** Let $0_I$ denote $\iota(0)$.

**Theorem 6.2.6.** For each $A \in I$, $A +_I 0_I =_I A$.

**Definition 6.2.3.** If $((a_n), (b_n))$ is a nested intervals, then so is $((-b_n), (-a_n))$. Let $- : I \to I$ be a map that assigns $((-b_n), (-a_n))$ to $((a_n), (b_n))$.

**Theorem 6.2.7.** For each $A \in I$, $A +_I (-A) =_I 0_I$.

**Theorem 6.2.8.** For each $A, B, C \in I$, if $A <_I B$ then $A +_I C < B +_I C$.

## 6.3 Multiplication of nested intervals

Because $<_I$ is a total order on $I$, for each $A \in I$ we obtain that $A <_I 0_I$ or $A =_I 0_I$ or $0_I <_I A$.

**Definition 6.3.1.** For each two nested intervals $A = ((a_n), (b_n))$ and $X = ((x_n), (y_n))$, we define a binary operation $\times_I$ as follows :

$$A \times_I X := \begin{cases} ((a_n x_n), (b_n y_n)) & \text{if } 0_I < A \text{ and } 0_I < X, \\ ((b_n x_n), (a_n y_n)) & \text{if } 0_I < A \text{ and } X < 0_I, \\ ((a_n y_n), (b_n x_n)) & \text{if } A < 0_I \text{ and } 0_I < X, \\ ((b_n y_n), (a_n x_n)) & \text{if } A < 0_I \text{ and } X < 0_I, \\ 0_I \quad \text{otherwise.} \end{cases}$$

We may think that if $A <_I 0_I$ and if $0_I <_I B$ then $A \times_I B$ must be $-((-A) \times_I B)$. The above definition is made by this way.

**Theorem 6.3.1.** If $A$ and $X$ are nested intervals, then so is $A \times_I X$; thus $\times_I$ is a binary operation on $I$.

**Theorem 6.3.2.** For $p, q \in \mathbb{Q}$, we obtain that $\iota(p \times q) =_I \iota(p) \times_I \iota(q)$.

**Theorem 6.3.3.** For each $A, B, C, D \in I$, if $A =_I B$ and $C =_I D$ then $A \times_I C =_I B \times_I D$.

**Theorem 6.3.4.** $(I, \times_I)$ is commutative, *i.e.*, $A \times_I B =_I B \times_I A$ for every $A, B \in I$.

**Theorem 6.3.5.** $(I, \times_I)$ is associative, *i.e.*, $(A \times_I B) \times_I C =_I A \times_I (B \times_I C)$ for every $A, B, C \in I$.

**Definition 6.3.2.** Let $1_I$ denote $\iota(1)$.

**Theorem 6.3.6.** For each $A \in I$, $A \times_I 1_I =_I A$.

**Definition 6.3.3.** For each nested interval $A = ((a_n), (b_n))$ satisfying that not $(A =_I 0_I)$, we define a unary operation $/_I$ as follows :

$$/_I A := ((1/b_n), (1/a_n)),$$

where if $a_n = 0$ for some $n$ then assign $1/a_n$ to 0; similarly to $b_n$.

**Theorem 6.3.7.** If $A$ is a nested interval, then so is $/_I A$; thus $/_I$ is a unary operation on $I$.

**Theorem 6.3.8.** If $A \in I$ and not ($A =_I 0_I$), then $A \times_I (/_I A) =_I 1_I$.

**Theorem 6.3.9.** For each $A, B, C \in I$, $A \times_I (B +_I C) =_I A \times_I B +_I A \times_I C$.

**Theorem 6.3.10.** For each $A, B \in I$, if $0_I <_I A$ and $0_I <_I B$ then $0_I <_I A \times_I B$.

Therefore we conclude that $I$ is a Dedekind-complete ordered field.

# Chapter 7

# Coq proof checking 2

In the below, we define each condition of a nested interval (Definition 6.1.1), and make a structure for a nested interval. And then we define $<_I$ and $=_I$ on $I$ named as *Ilt* and *Ieq*, respectively. And we eventually show that $I$ (with $<_I$ and $=_I$) is a dense linearly ordered set without endpoints.

`Definition` *compare* ($f\ g$ : *positive* $\rightarrow Q$) :=
$\exists$ *m* : *positive*, ($\forall$ *n* : *positive*,
($m \leq n$)%*positive* $\rightarrow f\ n \leq g\ n$).

`Definition` *get_closer* ($f\ g$ : *positive* $\rightarrow Q$) :=
$\forall$ *m n* : *positive*, ($\exists$ *p* : *positive*,
($n \leq p$)%*positive* $\wedge$ *g p* - *f p* < 1 # *m*).

`Definition` *increasing* ($f$ : *positive* $\rightarrow Q$) :=
$\exists$ *m* : *positive*, ($\forall$ *n p* : *positive*,
($m \leq n$)%*positive* $\rightarrow$ ($n \leq p$)%*positive* $\rightarrow f\ n \leq f\ p$).

`Definition` *decreasing* ($g$ : *positive* $\rightarrow Q$) :=
$\exists$ *m* : *positive*, ($\forall$ *n p* : *positive*,
($m \leq n$)%*positive* $\rightarrow$ ($n \leq p$)%*positive* $\rightarrow g\ p \leq g\ n$).

`Record` *I* := *mkI* {

```
  l : positive → Q;
  r : positive → Q;
  comp : compare l r;
  clo : get_closer l r;
  inc : increasing l;
  dec : decreasing r;
}.
Definition Ilt (a b : I) :=
∀ m : positive, (∃ n : positive,
(m ≤ n)%positive ∧ (r a) n < (l b) n).
Definition Ieq (a b : I) :=
compare (l b) (r a) ∧ compare (l a) (r b).
Lemma not_Ilt_equiv (a b : I) :
not (Ilt a b) ↔ compare (l b) (r a).

Theorem I_strict_order :
strict_order Ilt.
Lemma Ieq_trans_half (a b c : I) :
not (Ilt a b) → not (Ilt b c) → not (Ilt a c).
Theorem I_equivalence :
equivalence Ieq.
Theorem I_total_order :
total_order Ilt Ieq.
Theorem I_compatible_eq_lt :
compatible_eq_lt Ilt Ieq.
Definition const (q : Q) : positive → Q :=
fun ⇒ q.
```

```
Lemma compare_const (q : Q) :
compare (const q) (const q).

Lemma get_closer_const (q : Q) :
get_closer (const q) (const q).
Lemma increasing_const (q : Q) :
increasing (const q).

Lemma decreasing_const (q : Q) :
decreasing (const q).

Definition const_I (q : Q) : I :=
{|
   l := const q;
   r := const q;
   comp := compare_const q;
   clo := get_closer_const q;
   inc := increasing_const q;
   dec := decreasing_const q;
|}.

Theorem const_I_order_preserve (p q : Q) :
p < q → Ilt (const_I p) (const_I q).
Theorem const_I_order_reverse (p q : Q) :
Ilt (const_I p) (const_I q) → p < q.

Theorem const_I_dense (a b : I) :
(Ilt a b) →
∃ q : Q, (Ilt a (const_I q)) ∧ (Ilt (const_I q) b).
Theorem I_dense :
dense Ilt.

Definition translation (t : Q) (f : positive → Q) :=
```

```
fun q ⇒ f(q) + t.

Lemma compare_translation (t : Q) (a : I) :
compare (translation t (l a)) (translation t (r a)).

Lemma get_closer_translation (t : Q) (a : I) :
get_closer (translation t (l a)) (translation t (r a)).

Lemma increasing_translation (t : Q) (a : I) :
increasing (translation t (l a)).

Lemma decreasing_translation (t : Q) (a : I) :
decreasing (translation t (r a)).

Definition translation_I (t : Q) (a : I) : I:=
{|
  l := translation t (l a);
  r := translation t (r a);
  comp := compare_translation t a;
  clo := get_closer_translation t a;
  inc := increasing_translation t a;
  dec := decreasing_translation t a;
|}.

Lemma translation_gt (t : Q) (a : I) :
0 < t → Ilt a (translation_I t a).

Lemma translation_lt (t : Q) (a : I) :
t < 0 → Ilt (translation_I t a) a.

Theorem I_without_endpoints :
without_endpoints Ilt.

Definition I_dlos :=
{|
  X := I;
```

```
  Xlt := Ilt;
  Xeq := Ieq;
  eq := I_equivalence;
  st := I_strict_order;
  cp := I_compatible_eq_lt;
  to := I_total_order;
  den := I_dense;
  we := I_without_endpoints
|}.

Declare Scope I_scope.
Open Scope I_scope.

Notation "x < y" := (Ilt x y) : I_scope.
Notation "x == y" := (Ieq x y) : I_scope.
Notation "1" := (const_I 1) : I_scope.
Notation "0" := (const_I 0) : I_scope.
```

And we define addition of two nested intervals below. And then we show that $(I, +_I)$ is an abelian group. Additionally, we prove that addition preserves order in $I$, *i.e.*, $A <_I B \implies A +_I C <_I B +_I C$ for all $A, B, C \in I$.

```
Definition seq_plus (f g : positive → Q) :=
fun n : positive ⇒ (f n) + (g n).

Lemma compare_plus (a b : I) :
compare (seq_plus (l a) (l b)) (seq_plus (r a) (r b)).

Lemma get_closer_plus (a b : I) :
get_closer (seq_plus (l a) (l b)) (seq_plus (r a) (r b)).
Lemma increasing_plus (a b : I) :
increasing (seq_plus (l a) (l b)).

Lemma decreasing_plus (a b : I) :
```

*decreasing* (*seq_plus* (*r a*) (*r b*)).

`Definition` *Iplus* (*a b* : *I*) : *I* :=

{|

*l* := *seq_plus* (*l a*) (*l b*);

*r* := *seq_plus* (*r a*) (*r b*);

*comp* := *compare_plus a b*;

*clo* := *get_closer_plus a b*;

*inc* := *increasing_plus a b*;

*dec* := *decreasing_plus a b*;

|}.

`Notation` "x + y" := (*Iplus x y*) : *I_scope*.

`Theorem` *Iplus_Ieq_compatible* :

$\forall$ *a b c d* : *I*, *a* == *b* $\rightarrow$ *c* == *d* $\rightarrow$ *a* + *c* == *b* + *d*.

`Theorem` *Iplus_comm* :

$\forall$ *a b* : *I*, *a* + *b* == *b* + *a*.

`Theorem` *Iplus_assoc* :

$\forall$ *a b c* : *I*, (*a* + *b*) + *c* == *a* + (*b* + *c*).

`Theorem` *Iplus_0_r* :

$\forall$ *a* : *I*, *a* + 0 == *a*.

`Definition` *seq_opp* (*f* : *positive* $\rightarrow$ *Q*) :=

`fun` *n* : *positive* $\Rightarrow$ - (*f n*).

`Lemma` *compare_opp* (*a* : *I*) :

*compare* (*seq_opp* (*r a*)) (*seq_opp* (*l a*)).

`Lemma` *get_closer_opp* (*a* : *I*) :

*get_closer* (*seq_opp* (*r a*)) (*seq_opp* (*l a*)).

`Lemma` *increasing_opp* (*a* : *I*) :

*increasing* (*seq_opp* (*r a*)).

```
Lemma decreasing_opp (a : I) :
decreasing (seq_opp (l a)).
```

```
Definition Iopp (a : I) : I :=
{|
  l := seq_opp (r a);
  r := seq_opp (l a);
  comp := compare_opp a;
  clo := get_closer_opp a;
  inc := increasing_opp a;
  dec := decreasing_opp a;
|}.
```

```
Notation "- x" := (Iopp x) : I_scope.
```

```
Theorem Iplus_opp_r :
∀ a : I, a + (- a) == 0.
Theorem Iplus_order_compatible :
∀ a b c, a < b → a + c < b + c.
```

We define multiplication of $I$.

Previously, we already show that $A <_I B$ or $A =_I B$ or $B <_I A$ for every $A$, $B$ in $I$. It implies that $0_I <_I A$ or $A <_I 0_I$ or $A =_I 0_I$ for each $A \in I$. However, if there is no constructive way, then we cannot determine whether $0_I <_I A$ or not in Coq. We want to define multiplication of $I$ by dividing into several cases. Hence, the axiom *I_dec* below helps us to define multiplication.

```
Definition seq_mult (f g : positive → Q) :=
fun n : positive ⇒ (f n) × (g n).
```

```
Lemma pos_compare_mult (a b : I) :
0 < a → 0 < b → compare (seq_mult (l a) (l b)) (seq_mult (r a) (r b)).
```

```
Lemma pos_get_closer_mult (a b : I) :
```

$0 < a \to 0 < b \to$ *get_closer* (*seq_mult* (*l a*) (*l b*)) (*seq_mult* (*r a*) (*r b*)).

```
Lemma pos_increasing_mult (a b : I) :
0 < a → 0 < b → increasing (seq_mult (l a) (l b)).
Lemma pos_decreasing_mult (a b : I) :
0 < a → 0 < b → decreasing (seq_mult (r a) (r b)).
```

```
Axiom I_dec :
∀ a : I, ({0 < a} + {a < 0}) + {a == 0}.
```

*I_dec a* tells us that $0 < a$ or $a < 0$ or $a == 0$. First, *inleft* (`left` *H*) is the case that $0 < a$. And *inleft* (`right` *H*) is the case that $a < 0$. Finally, *inright H* is the case that $a == 0$. Using these, we define left and right rational sequences of a multiplication of two nested intervals, respectively.

```
Definition I_seq_mult_l (a b : I) :=
match (I_dec a) with
 | inleft (left H) ⇒
    match (I_dec b) with
     | inleft (left H) ⇒ seq_mult (l a) (l b)
     | inleft (right H) ⇒ seq_opp (seq_mult (r a) (r (- b)))
     | inright H ⇒ const 0
    end
 | inleft (right H) ⇒
    match (I_dec b) with
     | inleft (left H) ⇒ seq_opp (seq_mult (r (- a)) (r b))
     | inleft (right H) ⇒ seq_mult (l (- a)) (l (- b))
     | inright H ⇒ const 0
    end
 | inright H ⇒ const 0
end.
```

```
Definition I_seq_mult_r (a b : I) :=
match (I_dec a) with
  | inleft (left H) ⇒
      match (I_dec b) with
        | inleft (left H) ⇒ seq_mult (r a) (r b)
        | inleft (right H) ⇒ seq_opp (seq_mult (l a) (l (- b)))
        | inright H ⇒ const 0
      end
  | inleft (right H) ⇒
      match (I_dec b) with
        | inleft (left H) ⇒ seq_opp (seq_mult (l (- a)) (l b))
        | inleft (right H) ⇒ seq_mult (r (- a)) (r (- b))
        | inright H ⇒ const 0
      end
  | inright H ⇒ const 0
end.
Lemma I_compare_mult (a b : I) :
compare (I_seq_mult_l a b) (I_seq_mult_r a b).
Lemma I_get_closer_mult (a b : I) :
get_closer (I_seq_mult_l a b) (I_seq_mult_r a b).
Lemma I_increasing_mult (a b : I) :
increasing (I_seq_mult_l a b).
Lemma I_decreasing_mult (a b : I) :
decreasing (I_seq_mult_r a b).
Definition Imult (a b : I) : I :=
{|
   l := I_seq_mult_l a b ;
   r := I_seq_mult_r a b ;
```

*comp* := *I_compare_mult a b* ;

*clo* := *I_get_closer_mult a b* ;

*inc* := *I_increasing_mult a b* ;

*dec* := *I_decreasing_mult a b* ;

|}.

`Notation` "x *_I y " := (*Imult x y*) (`at level` 60, `right associativity`).

`Theorem` *Ieq_mult_compatible* :

$\forall$ *a b c d* : *I*, *a* == *b* $\rightarrow$ *c* == *d* $\rightarrow$ *a* $\times$_*I c* == *b* $\times$_*I d*.

`Theorem` *Imult_comm* :

$\forall$ *a b* : *I*, *a* $\times$_*I b* == *b* $\times$_*I a*.

`Theorem` *Imult_assoc* :

$\forall$ *a b c* : *I*, (*a* $\times$_*I b*) $\times$_*I c* == *a* $\times$_*I* (*b* $\times$_*I c*).

`Theorem` *Imult_1_r* :

$\forall$ *a* : *I*, *a* $\times$_*I* 1 == *a*.

`Definition` *seq_inv* (*f* : *positive* $\rightarrow$ *Q*) :=

`fun` *n* : *positive* $\Rightarrow$ / (*f n*).

`Lemma` *pos_compare_inv* (*a* : *I*) :

0 < *a* $\rightarrow$ *compare* (*seq_inv* (*r a*)) (*seq_inv* (*l a*)).

`Lemma` *pos_get_closer_inv* (*a* : *I*) :

0 < *a* $\rightarrow$ *get_closer* (*seq_inv* (*r a*)) (*seq_inv* (*l a*)).

`Lemma` *pos_increasing_inv* (*a* : *I*) :

0 < *a* $\rightarrow$ *increasing* (*seq_inv* (*r a*)).

`Lemma` *pos_decreasing_inv* (*a* : *I*) :

0 < *a* $\rightarrow$ *decreasing* (*seq_inv* (*l a*)).

`Lemma` *neg_compare_inv* (*a* : *I*) :

*a* < 0 $\rightarrow$ *compare* (*seq_inv* (*r a*)) (*seq_inv* (*l a*)).

`Lemma` *neg_get_closer_inv* (*a* : *I*) :

```
a < 0 → get_closer (seq_inv (r a)) (seq_inv (l a)).

Lemma neg_increasing_inv (a : I) :
a < 0 → increasing (seq_inv (r a)).

Lemma neg_decreasing_inv (a : I) :
a < 0 → decreasing (seq_inv (l a)).

Definition Iinv (a : I) : I :=
match (I_dec a) with
  | inleft (left H) ⇒
    {|
        l := seq_inv (r a) ;
        r := seq_inv (l a) ;
        comp := (pos_compare_inv a H) ;
        clo := (pos_get_closer_inv a H) ;
        inc := (pos_increasing_inv a H) ;
        dec := (pos_decreasing_inv a H) ;
    |}
  | inleft (right H) ⇒
    {|
        l := seq_inv (r a) ;
        r := seq_inv (l a) ;
        comp := (neg_compare_inv a H) ;
        clo := (neg_get_closer_inv a H) ;
        inc := (neg_increasing_inv a H) ;
        dec := (neg_decreasing_inv a H) ;
    |}
  | inright H ⇒ const_I 0
end.
```

Notation " / a" := (*Iinv a*).

Theorem *Imult_inv_r* :

$\forall$ *a* : *I*, *not* (*a* == 0) $\rightarrow$ *a* $\times$_*I* (/ *a*) == 1.

Theorem *Imult_plus_distr_r* :

$\forall$ *a b c* : *I*, *a* $\times$_*I* (*b* + *c*) == (*a* $\times$_*I b*) + (*a* $\times$_*I c*).

# Chapter 8

# Conclusion

In this paper, we first characterize a straight line by an intuitive approach and formalize what a straight line is. Especially, we define the Dedekind-completeness and show that it is equivalent to the least-upper-bound-property. After that, we show the existence of the reals, and prove the uniqueness of the reals. Then, we use Coq to prove the existence of the reals. In this way, we study a straight line, or the reals in the order sense.

Next, we define a nested interval. And we prove that the set of all nested intervals is a Dedekind-complete ordered field. We omit proofs in natural language and prove them by using Coq. We see again advantages of using proof assistant programs like Coq: for example, time saving and accurate proof checking. And it also helps people whether the proof one writes is really correct or not.

# Bibliography


[1] Thomas Hales. *Mathematics in the Age of the Turing Machine*. 2013. URL: https://arxiv.org/abs/1302.2898.

[2] Benjamin C. Pierce et al. *Logical Foundations*. Vol. 1. Software Foundations. Electronic textbook, 2022. URL: https://softwarefoundations.cis.upenn.edu.

[3] Karel Hrbacek; Thomas Jech. *Introduction to set theory*. CRC Press, 1999.

[4] Richard Dedekind. *Essays on the theory of numbers : I. Continuity and irrational numbers. II. The nature and meaning of number*. Dover Publications, 1963.

# 초 록


이 논문에서는 리하르트 데데킨트의 업적을 바탕으로 직선에 대한 직관적인 사실에 근거하여 직선이 무엇인지 정의한다. 그리고 증명보조기의 한 예인 Coq를 소개한다. 더불어 직선과 대응하는 대수적 구조인 실수에 두 연산, 덧셈과 곱셈을 정의한다. 마지막으로 이렇게 정의한 실수 구조가 완비순서체임을 Coq를 이용해서 보인다.



**주요어**: 실수, 데데킨드 완비성, 증명보조기, Coq

**학번**: 2018-24398

# 감사의 글

Otto van Koert 교수님과 국웅 교수님, 서인석 교수님, 수리과학부 행정실 선생님, 수리과학부 대학원 행정조교님, 연구실 동료들, 함께 공부한 2018년 전기 대학원생 친구들, 석사 수업을 담당해주신 교수님들 덕분에 이 논문을 쓸 수 있었습니다.

그리고 부모님, 동생, 친척, 정순모 교수님, 박신 선생님, 친구들, 하동우 선생님, 박정민 선생님, 할머니와 외할머니, 그 외 함께해 주시고 도움을 주신 많은 분들 덕분에 오늘날까지 수학을 공부하면서 잘 살아올 수 있었습니다.

이 분들과 이 논문을 읽어주신 분들께 감사의 말씀을 드립니다.