Introduction to Model Theory for C*-Algebras

Logic pre-reading

Anna Duwenig, Jennifer Pi

This is meant to be a quick summary of common mathematical notation/basic concepts from logic. In the first two sections, I will mostly provide just the definitions and notation, and give suggestions for further reading if you're unfamiliar with these concepts.

Sets

A set is a collection of objects. We write $x \in A$ so denote that x is an element of the set A, and similarly $x \notin A$ to denote that x is not an element of A.

We write $B \subseteq A$ to denote that B is a *subset* of A, i.e. that every element of B is also an element of A.

There are some basic operations on sets: for sets A, B,

1. We write their *union* as
$$A \cup B = \{x : x \in A \text{ or } x \in B\}.$$
2. We write their *intersection* as
$$A \cap B = \{x : x \in A \text{ and } x \in B\}.$$
3. If $B \subseteq A$, we write their *difference* as
$$A \setminus B = \{x : x \in A \text{ and } x \notin B\}.$$
4. If $B \subseteq A$, we can also write the *complement* of B as
$$B^c = \{x : x \in A \text{ and } x \notin B\}.$$
Note this is the same set as $A \setminus B$, but the notation B^c means we are treating A as an ambient space (so we do not specify A).
5. We can take the *Cartesian product* of A and B, written
$$A \times B = \{(x, y) : x \in A \text{ and } y \in B\}.$$

Here are some common sets and their notations:

1. $\mathbb{Z}$ for the integers.
2. $\mathbb{N}$ for the natural numbers.

3. $\mathbb{R}$ for the real numbers.

4. $\mathbb{C}$ for the complex numbers.

For more on sets, or for practice exercises if you are uncomfortable with any of these concepts, see Chapter 1 of Richard Hammack's *Book of Proof*, freely available on the internet.

Fundamentals

We set the following notation for the common logical connectives on statements:

1. The symbol $\wedge$ is read "and". [Think of $\bigcap$ for sets!]

2. The symbol $\vee$ is read "or". [Think of $\bigcup$ for sets!]

3. The symbol $\neg$ is read "not". [Think of complement for sets!]

We also have the following notation for quantifiers:

1. The symbol $\exists$ stands for "there exist(s)".

2. The symbol $\forall$ stands for "for all".

So for example, the statement

$$\forall y \in \mathbb{R}\, \exists x \in \mathbb{N} :\ y = x + 1$$

means that for every real number y there exists some natural number x so that $y = x + 1$.

For mathematical statements P, Q we also have:

1. $P \to Q$, which is read "P implies Q", if Q is true whenever P is true. [Sometimes this is also written $P \implies Q$.]

2. $P \leftrightarrow Q$, which is read "P if and only if Q", if $P \to Q$ and $Q \to P$. [Sometimes this is also written $P \iff Q$.]

For some more information on these logic fundamentals, you can read Chapter 2 of Richard Hammack's *Book of Proof*.

1 Model Theory

The material presented in this section is likely to be new for most students in the course. I follow the material presented in the first chapter of Tent and Ziegler's *A Course in Model Theory*.

By week 3 of the course, we will begin discussing model theory for continuous structures (namely, C^*-algebras). This section aims to provide some basic model theory material for classical (discrete) structures, to get you acquainted with these concepts more quickly when they appear in week 3. If you are very confident, or busy with the other pre-reading material, you could put off reading this section until the second half of week 2. But if you need time to digest definitions (as I do!), I would encourage you to read this now.

1.1 Structures

Definition 1.1 *A* language $\mathcal{L}$ *is a set of constant symbols, function symbols, and relation symbols.*

The function and relation symbols also come with an *arity* n, which tells us how many arguments the symbol takes. The symbols have no inherent meaning, but we usually choose symbols to invoke the meaning we usually associate; this is best understood by example.

Example 1.2 *The language of orders is* $\mathcal{L} = \{<\}$*, where* $<$ *is a binary relation symbol.*

Example 1.3 *The language of abelian groups is* $\mathcal{L} = \{0, +, -\}$*, where* 0 *is a constant symbol,* $+$ *is a binary function symbol, and* $-$ *is a unary function symbol.*

Example 1.4 *The language of rings is* $\mathcal{L} = \{0, +, -, 1, \cdot\}$*, where* $\{0, +, -\}$ *is the language of abelian groups,* 1 *is a constant symbol, and* $\cdot$ *is a binary function symbol.*

Languages only obtain meaning once they are interpreted in an appropriate structure.

Definition 1.5 *Let* $\mathcal{L}$ *be a language. An* $\mathcal{L}$-structure *is a pair* $\mathcal{A} = (A, (Z^{\mathcal{A}})_{Z \in \mathcal{L}})$*, where* A *is a set and* $Z^{\mathcal{A}}$ *is the* $\mathcal{A}$*-interpretation of* Z*.*

The cardinality *of a structure is defined to be the cardinality of its universe* A*.*

Sometimes we call the set A the *universe* of $\mathcal{A}$.

The notion of $\mathcal{L}$-structure is again best illustrated by example.

Example 1.6 *Let* $\mathcal{L}_{\text{ord}}$ *be the language of orders. Then* $\mathcal{A} = (\mathbb{R}, (<^{\mathcal{A}}))$ *is an* $\mathcal{L}_{\text{ord}}$*-structure, where* $<^{\mathcal{A}}$ *is interpreted as the usual ordering on the real numbers, i.e.* $a < b$ *if and only if* $b - a$ *is a positive number. Of course,* $(\mathbb{Z}, (<^{\mathcal{A}}))$ *is also a valid example of an* $\mathcal{L}_{\text{ord}}$*-structure.*

Example 1.7 *Let* $\mathcal{L}_{\text{AbGrp}}$ *be the language of abelian groups. Then* $\mathcal{G} = (\mathbb{Z}, (0^{\mathcal{G}}, +^{\mathcal{G}}, -^{\mathcal{G}}))$ *is an* $\mathcal{L}_{\text{AbGrp}}$*-structure, where*

- $0^{\mathcal{G}} = 0 \in \mathbb{Z}$*;*
- $+^{\mathcal{G}}$ *is interpreted as the usual addition in the integers;*
- $-^{\mathcal{G}}$ *is interpreted as the map which takes an integer* n *to its additive inverse* $-n$*.*

Example 1.8 *Let $\mathcal{L}_{\text{Rng}}$ be the language of rings. Then $\mathcal{R} = (M_n(\mathbb{Z}), (0^{\mathcal{R}}, +^{\mathcal{R}}, -^{\mathcal{R}}, 1^{\mathcal{R}}, .^{\mathcal{R}}))$ is an $\mathcal{L}_{\text{Rng}}$-structure, where*

- *$0^{\mathcal{R}}$ is interpreted as the $(n \times n)$-matrix of all zeroes;*
- *$+^{\mathcal{R}}$ is interpreted as the usual entrywise matrix addition;*
- *$-^{\mathcal{R}}$ is the interpreted as the map which takes a matrix A to its additive inverse $-A$ (i.e. multiplies entrywise by -1);*
- *$1^{\mathcal{R}}$ is interpreted as the $(n \times n)$ identity matrix;*
- *$.^{\mathcal{R}}$ is interpreted as the usual matrix multiplication.*

Note that in both of the previous examples, the symbols are interpreted to mean what we usually think they mean, and that the resulting structures are indeed a group and a ring, respectively. However, I will again point out that the symbols themselves are absent of meaning, as the following example shows:

Example 1.9 *Let $\mathcal{B} = (\mathbb{Z}, (0^{\mathcal{G}}, +^{\mathcal{G}}, -^{\mathcal{G}}))$ be the $\mathcal{L}_{\text{AbGrp}}$-structure with $\mathbb{Z}$ as the underlying set and with:*

- *$0^{\mathcal{G}}$ interpreted as* 123,
- *$+^{\mathcal{G}}$ interpreted as multiplication of integers, and*
- *$-^{\mathcal{G}}$ interpreted as the map that sends every integer to* 47.

This is a perfectly valid $\mathcal{L}_{\text{AbGrp}}$-structure! Of course, when we do mathematics we will usually want some amount of structure to be assumed, and the structure $\mathcal{B}$ will not satisfy the formal axioms for groups, which we present later in Example 1.32 after we've discussed formulas and theories.

We should also discuss maps between structures. In the following, we will fix a language $\mathcal{L}$ and two $\mathcal{L}$-structures $\mathcal{A}$ and $\mathcal{B}$.

Definition 1.10 *A map $h\colon A \to B$ is called a* homomorphism *if for all $a_1, \ldots, a_n \in A$, h respects all the symbols of $\mathcal{L}$ In other words, if for all constants c, n-ary function symbols f, and relation symbols R from $\mathcal{L}$, we have*

$$\begin{aligned} h(c^{\mathcal{A}}) &= c^{\mathcal{B}} \\ h(f^{\mathcal{A}}(a_1, \ldots, a_n)) &= f^{\mathcal{B}}(h(a_1), \ldots, h(a_n)) \\ R^{\mathcal{A}}(a_1, \ldots, a_n) &\implies R^{\mathcal{B}}(h(a_1), \ldots, h(a_n)) \end{aligned}$$

A homomorphism between $\mathcal{L}$-structures is denoted by $h\colon \mathcal{A} \to \mathcal{B}$.

If in addition h is injective and $R^{\mathcal{A}}(a_1, \ldots, a_n) \iff R^{\mathcal{B}}(h(a_1), \ldots, h(a_n))$ for all $a_1, \ldots, a_n \in A$, then h is an embedding. *An* isomorphism *is a surjective embedding.*

Definition 1.11 *We call $\mathcal{A}$ a* substructure *of $\mathcal{B}$ if $A \subseteq B$ (inclusion of universes) and if the inclusion map is an embedding from $\mathcal{A}$ to $\mathcal{B}$. We denote this by $\mathcal{A} \subseteq \mathcal{B}$. In this case, we will also say $\mathcal{B}$ is an* extension *of $\mathcal{A}$.*

Definition 1.12 *Suppose that S is a subset of the universe A of $\mathcal{A}$. The* substructure generated by S in $\mathcal{A}$ *is formed by first including the interpretations of all constant symbols $c^{\mathcal{A}}$, and then closing under (the interpretation of) all function symbols. Formally, define $\tilde{S} = S \cup \{c^{\mathcal{A}}\}$, and then close under all operations $f^{\mathcal{A}}$.*

Exercise 1.13 (Lemma 1.1.7 of Tent-Ziegler) *If $\mathcal{A}$ is generated by a set S, then every homomorphism $h\colon \mathcal{A} \to \mathcal{B}$ is determined by its values on S.*

Exercise 1.14 (Optional; Lemma 1.1.8 of Tent-Ziegler) *Let $h\colon \mathcal{A} \to \mathcal{A}'$ be an isomorphism, and let $\mathcal{B}$ an extension of $\mathcal{A}$. Then there exists an extension $\mathcal{B}'$ of $\mathcal{A}'$ and an isomorphism $g\colon \mathcal{B} \to \mathcal{B}'$ that extends h.*

1.2 Formulas

Definition 1.15 *An* $\mathcal{L}$-term *is a word (i.e. sequence of symbols) built out of constants, the function symbols of $\mathcal{L}$, and the variables $v_0, v_1, \ldots$ according to the following rules:*

1. *We declare every variable v_i and every constant c to be an $\mathcal{L}$-term.*
2. *If f is an n-ary function symbol and $t_1, \ldots, t_n$ are $\mathcal{L}$-terms, then we declare $f(t_1, \ldots, t_n)$ to be an $\mathcal{L}$-term.*

If there is no ambiguity as to which language we are talking about, we will often just say term *instead of $\mathcal{L}$-term.*

Note that I've written $f(t_1, \ldots, t_n)$ rather than $ft_1 \ldots t_n$ for readability, and in general we will do this, e.g. we write

$$(x + y) \cdot (z + w),$$

even though formally this should read

$$\cdot(+xy)(+zw).$$

We can then interpret terms by substituting elements in for the variables; this is called an *assignment*: Let $\mathcal{A}$ be an $\mathcal{L}$-structure, and let $\vec{a} = (a_0, a_1, \ldots) \in A^{\mathbb{N}}$. Then an $\mathcal{L}$-term t defines an element $t^{\mathcal{A}}[\vec{a}]$ of A in the natural way by replacing each instance of v_i with a_i. (To see this formally, see Definition 1.2.2 of Tent and Ziegler, but I feel this is clearer written informally).

You can try to prove the following by induction on complexity of the terms, to check your understanding:

Lemma 1.16 (Lemma 1.2.4 of Tent-Ziegler) *If $t, t_1, \ldots, t_n$ are terms, then*

$$t(t_1, \ldots, t_n)^{\mathcal{A}}[\vec{a}] = t^{\mathcal{A}}\left[t_1^{\mathcal{A}}[\vec{a}], \ldots, t_n^{\mathcal{A}}[\vec{a}]\right].$$

Lemma 1.17 (Lemma 1.2.5 of Tent-Ziegler) *Let $h\colon \mathcal{A} \to \mathcal{B}$ be a homomorphism and $t(v_1, \ldots, v_n)$ a term. Then we have for all $a_1, \ldots, a_n$ from A,*

$$t^{\mathcal{B}}[h(a_1), \ldots, h(a_n)] = h(t^{\mathcal{A}}[a_1, \ldots, a_n]).$$

Finally we define $\mathcal{L}$-formulas. These capture all "basic mathematical statements" we can make about an $\mathcal{L}$-structure. We inherently always have variables $v_0, v_1, \ldots$, and the equality symbol $=$, so we don't specify them in the language, or in the definition of formulas below.

Definition 1.18 *There are two types of* atomic $\mathcal{L}$-formulas*:*

1. $t_1 = t_2$*, where* t_1, t_2 *are* $\mathcal{L}$*-terms, and*
2. $Rt_1 \dots t_n$*, where* R *is an* n*-ary relation symbol and* $t_1, \dots, t_n$ *are terms.*

In general, $\mathcal{L}$-formulas *are built inductively from atomic* $\mathcal{L}$*-formulas and the following list:*

3. $\neg\psi$*, where* ψ *is an* $\mathcal{L}$*-formula,*
4. $(\psi_1 \wedge \psi_2)$*, where* ψ_1 *and* ψ_2 *are* $\mathcal{L}$*-formulas,*
5. $\exists x\, \psi(x)$*, where* ψ *is an* $\mathcal{L}$*-formula and* x *is a variable.*

These formulas have complexity defined by the number of occurrences of $\neg$, $\exists$, *and* $\wedge$*. This allows us to do arguments based on induction on (the complexity of) formulas.*

Note that in items 4 and 5 above, we could have equivalently used $\psi_1 \vee \psi_2$ and $\forall x\, \psi(x)$ instead, because we included negation in item 3. (Verify this if you're not sure about the last sentence: rewrite any expression involving $\exists x\, \psi$ using $\neg$ and $\forall x$.) Thus, we continue to use $\vee$ and $\forall$ as convenient shorthand, even though they are not formally included in the definition of $\mathcal{L}$-formulas.

Exercise 1.19 *Given any* $\mathcal{L}$*-formulas* ψ_1 *and* ψ_2*, rewrite* $(\psi_1 \to \psi_2)$ *using only the list given in Definition 1.18.*

For an $\mathcal{L}$-structure $\mathcal{A}$, we can discuss the notion of a tuple of elements of A, denoted $\vec{b}$, *satisfying* a formula in a structure, via the *assignment* that we discussed earlier for terms.

Definition 1.20 *Let* $\mathcal{A}$ *be an* $\mathcal{L}$*-structure. Then for all* $\mathcal{L}$*-formulas* ϕ *and all assignments* $\vec{b}$*, we define the relation*

$$\mathcal{A} \models \phi[\vec{b}]$$

recursively over ϕ*:*

1. $\mathcal{A} \models t_1 = t_2 \iff t_1^{\mathcal{A}}[\vec{b}] = t_2^{\mathcal{A}}[\vec{b}]$*;*
2. $\mathcal{A} \models Rt_1 \dots t_n[\vec{b}] \iff R^{\mathcal{A}}(t_1^{\mathcal{A}}[\vec{b}], \dots, t_n^{\mathcal{A}}[\vec{b}])$*;*
3. $\mathcal{A} \models \neg\psi[\vec{b}] \iff \mathcal{A} \not\models \psi[\vec{b}]$*;*
4. $\mathcal{A} \models (\psi_1 \wedge \psi_2)[\vec{b}] \iff \mathcal{A} \models \psi_1[\vec{b}]$ *and* $\mathcal{A} \models \psi_2[\vec{b}]$*;*
5. $\mathcal{A} \models (\exists x\, \psi(x))[\vec{b}] \iff$ *there exists* $a \in A$ *such that* $\mathcal{A} \models \psi[\vec{b}, a/x]$*, where the notation* a/x *means that we make the assignment* a *in for every occurrence of the variable* x*.*

If $\mathcal{A} \models \phi[\vec{b}]$ *holds, we say* $\vec{b}$ *satisfies* ϕ *(in* $\mathcal{A}$*).*

Informally, the above definition basically says a structure $\mathcal{A}$ "believes" $\phi[\vec{b}]$, which we denote by $\mathcal{A} \models \phi[\vec{b}]$. It's the formal way of writing down recursively that we know a formula is true in $\mathcal{A}$.

Example 1.21 *Let $\mathcal{L}$ be the language of abelian groups, and let $\mathcal{A}$ be any structure which is an abelian group (for example, Example 1.7). On the other hand let $\mathcal{B}$ be any structure which is a nonabelian group (for example, the invertible matrices $GL_2(\mathbb{C})$ with $+$ interpreted as matrix multiplication). Define terms $t_1 := v_1 + v_2$ and $t_2 := v_2 + v_1$. Then $\mathcal{A} \models t_1 = t_2$, since for any assignment $\vec{b}$ we have the interpretations agree: $t_1^{\mathcal{A}}[\vec{b}] = t_2^{\mathcal{A}}[\vec{b}]$.*

On the other hand, $\mathcal{B} \not\models t_1 = t_2$, since there exist invertible matrices that do not commute under matrix multiplication.

Example 1.22 *Let $\mathcal{G}$ be the structure of the usual group of integers $\mathbb{Z}$, as in Example 1.7. Then,*

$$\mathcal{G} \models \forall x \exists y \, (x + y = 0).$$

In plain language, this says that every element of $\mathcal{G}$ has an inverse.
Let's also do an example using an assignment. Let

$$\psi(x_1, x_2) \text{ be the formula } x_1 = x_2 + x_2.$$

Then, for example, $\mathcal{G} \models \phi(2, 1)$ but $\mathcal{G} \not\models \phi(0, 4)$.

We say a variable x is a *free variable* in ϕ if it occurs at a place which is not within the scope of a quantifier $\exists$, $\forall$ (i.e. we can make an assignment for x). Otherwise we say the variable x is *bound* (it is bound by the quantifier that it occurs with).

Example 1.23 *In the formula $\phi(x, y)$ given by*

$$\exists x \, (y = x + x),$$

the variable x is bound and the variable y is free.

Definition 1.24 *An $\mathcal{L}$-formula ϕ without free variables is called a* sentence. *If ϕ is a sentence and if $\mathcal{A}$ is an $\mathcal{L}$-structure for which $\phi^{\mathcal{A}}[\vec{b}]$ is satisfied in $\mathcal{A}$ for some (equivalently, all) assignments $\vec{b}$, then we write $\mathcal{A} \models \phi$.*

In that case we say $\mathcal{A}$ is a model *of ϕ. If Σ is a set of sentences, then $\mathcal{A}$ is a model of Σ, denoted*

$$\mathcal{A} \models \Sigma,$$

if $\mathcal{A}$ is a model of all sentences in Σ.

Example 1.25 *The formula $x = 2y - 1$ is a sentence in any language, while $\exists x \, (x = 2y - 1)$ is not a sentence, since the variable y is free.*

In the language of groups, the formula $\forall x \exists y \, (x + y = 0)$ is a sentence, but $\exists y \, (x + y = 0)$ is not a sentence, since the variable x is free.

A formula is in *negation normal form* if it is built out of atomic formulas and their negations using only $\wedge$, $\vee$, $\exists$, $\forall$.

Definition 1.26 *A formula in negation normal form that does not contain any existential quantifier is called* universal. *Conversely, formulas in negation normal form without universal quantifiers are called* existential.

Exercise 1.27 *Let $h\colon \mathcal{A} \to \mathcal{B}$ be an embedding of $\mathcal{L}$-structures. Then, for all existential formulas $\phi(x_1, \ldots, x_n)$ and all $a_1, \ldots, a_n \in A$, we have*

$$\mathcal{A} \models \phi[a_1, \ldots, a_n] \implies \mathcal{B} \models \phi[h(a_1), \ldots, h(a_n)].$$

Hint: Go by induction on the complexity of ϕ.

Finally, this is one of the most important concepts from this prereading: the notion of an *ultraproduct*. These allow us to take some collection of structures and form a new structure which has properties given by "a large majority" of the structures. To define this formally, we first define a filter; these "capture large sets" from the index set.

Definition 1.28 *A* filter *$\mathcal{F}$ on a set I is a nonempty collection of subsets of I which does not contain the empty set and is closed under intersections and supersets. In mathematical symbols:*

- *$\emptyset \notin \mathcal{F}$;*
- *for $A, B \in \mathcal{F}$, we have $A \cap B \in \mathcal{F}$;*
- *if $A \in \mathcal{F}$ and $A \subseteq C \subseteq I$, then $C \in \mathcal{F}$.*

A filter $\mathcal{F}$ is called an ultrafilter *if for every $A \subseteq I$ we have $A \in \mathcal{F}$ or $I \setminus A \in \mathcal{F}$. (Equivalently, an ultrafilter is a maximal filter, under the ordering given by containment.)*

For a family $(\mathcal{A}_i : i \in I)$ of $\mathcal{L}$-structures and $\mathcal{U}$ an ultrafilter on I, we define the *ultraproduct*, denoted $\prod_{\mathcal{U}} \mathcal{A}_i$ as follows: On the Cartesian product $\prod_{i \in I} \mathcal{A}_i$, the ultrafilter $\mathcal{U}$ defines an equivalence relation $\sim_{\mathcal{U}}$ by

$$(a_i)_{i \in I} \sim_{\mathcal{U}} (b_i)_{i \in I} \iff \{i \in I : a_i = b_i\} \in \mathcal{U}.$$

On the set of equivalence classes $(a_i)_{\mathcal{U}}$ we define an $\mathcal{L}$-structure $\prod_{\mathcal{U}} \mathcal{A}_i$ as follows:

- For constants $c \in \mathcal{L}$, put $c^{\prod_{\mathcal{U}} \mathcal{A}_i} = (c^{\mathcal{A}_i})_{\mathcal{U}}$. In words: the $\prod_{\mathcal{U}} \mathcal{A}_i$-interpretation of c is defined to be the $\sim_{\mathcal{U}}$-equivalence class of the tuple $(c^{\mathcal{A}_i})_{i \in I}$ of $\mathcal{A}_i$-interpretations of c.
- For n-ary function symbols $f \in \mathcal{L}$, we put

$$f^{\prod_{\mathcal{U}} \mathcal{A}_i}((a_i^1)_{\mathcal{U}}, \ldots, (a_i^n)_{\mathcal{U}}) = (f^{\mathcal{A}_i}(a_i^1, \ldots, a_i^n))_{\mathcal{U}}.$$

- For n-ary relation symbols $R \in \mathcal{L}$, we put

$$R^{\prod_{\mathcal{U}} \mathcal{A}_i}((a_i^1)_{\mathcal{U}}, \ldots, (a_i^n)_{\mathcal{U}}) \iff \{i \in I : R^{\mathcal{A}_i}(a_i^1, \ldots, a_i^n)\} \in \mathcal{U}.$$

Exercise 1.29 (Ultraproducts and Łos's Theorem)

1. *Show that the ultraproduct $\prod_{\mathcal{U}} \mathcal{A}_i$ is well-defined as a structure, i.e. there is a well-defined interpretation of function and relation symbols in $\prod_{\mathcal{U}} \mathcal{A}_i$.*
2. *Prove Łos's Theorem: for any $\mathcal{L}$-formula ϕ we have*

$$\prod_{\mathcal{U}} \mathcal{A}_i \models \phi((a_i^1)_{\mathcal{U}}, \ldots (a_i^n)_{\mathcal{U}}) \iff \{i \in I : \mathcal{A}_I \models \phi(a_i^1, \ldots, a_i^n)\} \in \mathcal{F}.$$

Hint: go by induction on complexity of ϕ.

1.3 Theories

Definition 1.30 *An $\mathcal{L}_{\text{ord}}$-*theory *T is a set of $\mathcal{L}$-sentences.*

Example 1.31 *Let $\mathcal{L} = \{<\}$ be the language of orders as in Example 1.2. Then the following captures the theory of dense linear orders without endpoints, usually denoted* **DLO**.
Axioms for linear ordering:

- $\forall x(x \not< x)$
- $\forall x, y, z\,(x < y \wedge y < z \implies x < z)$
- $\forall x, y\,(x < y \vee y < x \vee y = x)$

Density:

- $\forall x, z\,(x < z \to \exists y\,(x < y < z))$

No endpoints:

- $\forall y\,\exists x\,(x < y)$
- $\forall y\,\exists x\,(x > y)$

Then, for the $\mathcal{L}_{\text{ord}}$-structure $\mathcal{A} = (\mathbb{R}, (<^{\mathcal{A}}))$ from Example 1.6, we have $\mathcal{A} \models$ **DLO**, *while for $\mathcal{B} = (\mathbb{Z}, (<^{\mathcal{B}}))$, we have $\mathcal{B} \not\models$* **DLO**, *since density is not satisfied.*

Example 1.32 *Let $\mathcal{L}_{\text{AbGrp}}$ be the language of abelian groups from Example 1.3. Then the theory of abelian groups, denoted* **AbG**, *has the axioms:*

- $\forall x, y, z\,\big((x + y) + z = x + (y + z)\big)$
- $\forall x\,\big(0 + x = x\big)$
- $\forall x\,\big((-x) + x = 0\big)$
- $\forall x, y\,\big(x + y = y + x\big)$

Then, for the $\mathcal{L}_{\text{AbGrp}}$-structure $\mathcal{G}$ from Example 1.7, we have $\mathcal{G} \models$ **AbGrp**.

Example 1.33 *Let $\mathcal{L}_{\text{Rng}}$ be the language of rings from Example 1.4. Then the theory of rings, denoted* **Ring**, *has the axioms:*

- **AbG**
- $\forall x, y, z\,\big((xy)z = x(yz)\big)$
- $\forall x\,\big(1x = x\big)$
- $\forall x, y, z\,\big(x(y + z) = xy + xz\big)$

Then, for the $\mathcal{L}_{\text{Rng}}$-structure $\mathcal{R}$ from Example 1.8, we have $\mathcal{R} \models$ **Ring**.

We now have a concept which verifies that the language we write down is actually useful for studying a class of objects. For example, when we want to study abelian groups, we insist that the $\mathcal{L}_{\mathbf{AbGrp}}$-structures of interest are only those that model the theory **AbG**. [This gets rid of silly examples like Example 1.9].

Definition 1.34 *Two $\mathcal{L}$-structures $\mathcal{A}$ and $\mathcal{B}$ are called* elementary equivalent*, denoted*

$$\mathcal{A} \equiv \mathcal{B},$$

if they have the same theory; more explicitly, for all $\mathcal{L}$-sentences ϕ,

$$\mathcal{A} \models \phi \iff \mathcal{B} \models \phi.$$

Isomorphic structures are always elementarily equivalent, but the converse holds only for finite structures.

Exercise 1.35 *Show that if $\mathcal{A}$ is a finite $\mathcal{L}$-structure and $\mathcal{B}$ is elementarily equivalent to $\mathcal{A}$, then $\mathcal{A}$ and $\mathcal{B}$ are isomorphic.*

Exercise 1.36 *Two structures $\mathcal{A}$ and $\mathcal{B}$ are* partially isomorphic *if there is a non-empty set $\mathcal{I}$ of isomorphisms between substructures of $\mathcal{A}$ and $\mathcal{B}$ with the* back-and-forth *property:*

- *For every $f \in \mathcal{I}$ and $a \in A$ there is an extension of f in $\mathcal{I}$ with a in its domain.*
- *For every $f \in \mathcal{I}$ and $b \in B$ there is an extension of f in $\mathcal{I}$ with b in its image.*

Show that partially isomorphic structures are elementarily equivalent.