

5165 Notes on Logic

Sentential Logic

Well-formed Formulas and Mathematical Induction

Definition 1. A *well-formed sequence* is a finite sequence $\alpha_1, \dots, \alpha_n$ of expressions such that each α_i is either:

- i) a sentence symbol
- ii) $(\neg\alpha_j)$, for some $j < i$, or
- iii) $(\alpha_j \rightarrow \alpha_k)$, for some $j, k < i$, or
- iv) $(\alpha_j \vee \alpha_k)$, for some $j, k < i$, or
- v) $\alpha_j \wedge \alpha_k$, for some $j, k < i$, or
- vi) $(\alpha_j \leftrightarrow \alpha_k)$, for some $j, k < i$.

An expression α is a *well-formed formula* (wff) if there is a well-formed sequence $\alpha_1, \dots, \alpha_n$ such that $\alpha = \alpha_n$.

Remark. It is clear that every non-empty initial subsequence of a well-formed sequence is also a well-formed sequence. It follows that if $\alpha_1, \dots, \alpha_n$ is a well-formed sequence, then each expression α_i in the sequence is a well-formed formula.

Proposition 2. *Every wff has one of the following forms:*

$$A, (\neg\alpha), (\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \rightarrow \beta), (\alpha \leftrightarrow \beta), \quad (*)$$

where A is a sentence symbol and α and β are wffs.

Proof. Suppose α is a wff. By definition, there is a well-formed sequence

$$\alpha_1, \dots, \alpha_n$$

such that $\alpha_n = \alpha$. By the definition of a well-formed sequence, α_n must have one of the forms (*). \square

Proposition 3. *Let S be a set of wffs. If*

- a) *Every sentence symbol is in S , and*
- b) *For every wff α and wff β ,*

if α and β are in S then each of $(\neg\alpha)$, $(\alpha \rightarrow \beta)$, $(\alpha \vee \beta)$, $\alpha \wedge \beta$, and $(\alpha \leftrightarrow \beta)$ is in S ,

then S is the set of all wffs.

Proof. Suppose S is a set of wffs which satisfies (a) and (b). Let α be a wff. We show that α is in S . Since α is a wff, there is a well-formed sequence $\alpha_1, \dots, \alpha_n$ such that $\alpha = \alpha_n$. We show by induction on i , for $1 \leq i \leq n$, that $\alpha_i \in S$.

Basis: $i = 1$. Then α is a sentence symbol, and so $\alpha \in S$ by (a).

Induction Step: $i > 1$. By the induction hypothesis, $\alpha_j \in S$ for every j such that $1 \leq j < i$. Since $\alpha_1, \dots, \alpha_n$ is a well-formed sequence, it satisfies properties (i)-(vi) above.

Case (i): α_i is a sentence symbol. Then $\alpha_i \in S$ by (a).

Case (ii): α_i is $(\neg\alpha_j)$, for some $j < i$. By induction hypothesis, $\alpha_j \in S$. But then by (b), $\alpha_i = (\neg\alpha_j) \in S$.

Case (iii): α is $(\alpha_j \rightarrow \alpha_k)$, for some $j, k < i$. By the induction hypothesis, both α_j and α_k are in S . But then by (b), $\alpha_i = (\alpha_j \rightarrow \alpha_k) \in S$.

The other cases are similar. \square

Exercise (Optional) 4. Let \mathcal{P} be the following collection of sets S of expressions:

$S \in \mathcal{P} \iff S$ is a set of expressions & S satisfies (a) and (b) of Prop. 2 .

Show that $\bigcap \mathcal{P}$ is the set of all wffs.¹

Theorem (Informal) 5. Assume that the set $\{A_1, \dots, A_n\}$ of sentence symbols is effectively decidable. Then the set of wffs is effectively decidable.

The following is a more precise statement:

Exercise (Optional) 6. Assign Gödel numbers to wffs in such a way that

- $\{\#A : A \text{ is a sentence symbol}\}$ is primitive recursive, and
- $\{\#\alpha : \alpha \text{ is a wff}\}$ is primitive recursive.

Your assignment of Gödel numbers should have the property:

There is an algorithm \mathcal{A} such that for $n \in \mathbb{N}$, if n is the Gödel number of a wff α , then \mathcal{A} on input n gives output α . \square

¹ Note: $\bigcap \mathcal{P}$ is the intersection of all members of \mathcal{P} . So for example, if $\mathcal{P} = \{A, B\}$, then

$$\bigcap \mathcal{P} = \bigcap \{A, B\} = A \cap B ,$$

and if $\mathcal{P} = \{A_1, \dots, A_n, \dots\}$, then

$$\bigcap \mathcal{P} = \bigcap \{A_1, \dots, A_n, \dots\} = \bigcap_{n=1}^{\infty} A_n .$$

Logically Valid Sentences and Logically Correct Reasoning

The sentence

The current outdoor temperature in Minneapolis is less than 32 degrees.

happens to be a true sentence. To realize that it is true one must understand the meaning of the words (temperature, etc.) in the sentence (as well as know something about the current temperature).

The sentence

The current outdoor temperature in Minnesota is less than 32 degrees, or it is not the case that the current outdoor temperature is less than 32 degrees.

also is a true sentence. However, in this case the truth of the sentence does not depend upon understanding the meaning of the words (temperature, etc.). Rather, the sentence is seen to be true simply by looking at the *form* of the sentence. This sentence has the form $A \vee \neg A$ and *every* such sentence is true.² Such a sentence is called *logically valid*, or simply *valid*.

Next consider the following form of reasoning:

Premise: The current outdoor temperature in Minneapolis is below 32 degrees.

Conclusion: It is snowing.

Now the conclusion might be true, or it might be false. Next consider the reasoning:

Premise: The current outdoor temperature in Minneapolis is below 32 degrees.

Conclusion: The current outdoor temperature in Minneapolis is below 32 degrees, or it is snowing.

The conclusion might be true, or it might be false. However, in this case the reasoning is seen to be correct, since *if* the premise is true, then the conclusion *must* also be true. We say in this case that the reasoning is *logically valid*, or simply *valid*. The fact that reasoning is logically valid does not mean that the conclusion of the reasoning is true. It simply means that *if* the premises (i.e. hypotheses) are true, then the conclusion must also be true. One of our goals is to gain insight into the *form* of reasoning which is valid.

Exercise 6. Determine whether the conclusion is correctly inferred from the premises, by making a suitable translation into the language of Sentential Logic.

I *Premises:*

² With apologies to Intuitionists.

- 1) If Jones did not meet Smith last night, then either Smith is the murderer, or Jones is lying.
- 2) If Smith is not the murderer, then Jones did not meet Smith last night, and the murder took place after midnight.
- 3) If the murder took place after midnight, then either Smith is the murderer or Jones is lying.

Conclusion:

Smith is the murderer.

II *Premises:*

- 1) If capital investment remains constant, then government spending will increase or the level of unemployment will increase.
- 2) If government spending does not increase, then taxes will be reduced.
- 3) If taxes are reduced and capital investment remains constant, then the level of unemployment will not increase.

Conclusion:

Government spending will increase.

Disjunctive Normal Form and Conjunctive Normal Form

Exercise 7. Find a wff α in disjunctive normal form, and a wff β in conjunctive normal form which are equivalent to the given wff.

- a) $\neg A \rightarrow (\neg B \vee C) \rightarrow A$
- b) $((\neg A \rightarrow A \wedge B) \rightarrow C) \rightarrow B \wedge A$
- c) $\neg(A \rightarrow B) \rightarrow C$

Topology and the Compactness Theorem

Let

$$2^{\mathbb{N}} = \{f : f : \mathbb{N} \rightarrow \{0, 1\}\}.$$

Note that there is a one-to-one correspondence between $2^{\mathbb{N}}$ and the set of truth assignments. (For v a truth assignment, let

$$f_v(n) = ? .)$$

Let the set $\{0, 1\}$ have the discrete topology, and the set $2^{\mathbb{N}}$ have the product topology. Note that the set $2^{\mathbb{N}}$ is compact. (why?) With each set Δ of wffs we can associate a subset X_{Δ} of $2^{\mathbb{N}}$, where

$$X_{\Delta} = \{f_v : v \text{ satisfies } \Delta\}.$$

Exercise (For Math majors) 8. Use the fact that $2^{\mathbb{N}}$ is a compact space to give a proof of the Compactness Theorem.

Infinite Trees and the Compactness Theorem

We can identify with each finite sequence of 0's and 1's a unique node on the following infinite tree \mathbb{T} : An infinite path p through the tree \mathbb{T} corresponds to a unique infinite sequence of 0's and 1's. This in turn defines a unique function $f \in \mathbb{N}$, where

$$\begin{aligned} f(0) &\text{ is the first term of } p \\ f(1) &\text{ is the second term of } p \\ &\vdots \\ f(n) &\text{ is the } n + 1\text{-st term of } p \\ &\vdots \end{aligned}$$

Thus each infinite path p through the tree \mathbb{T} is identified with a unique member f of $2^{\mathbb{N}}$. Conversely, a function $f \in 2^{\mathbb{N}}$ defines a unique infinite sequence of 0's and 1's, namely the infinite sequence $(f(0), f(1), \dots, f(n), \dots)$, and this in turn is associated with a unique path p through the \mathbb{T} . Thus there is a one-to-one correspondence between $2^{\mathbb{N}}$ and paths through the tree \mathbb{T} .

Now consider a subtree \mathbb{S} of the tree \mathbb{T} obtained by chopping off certain branches of \mathbb{T} . \mathbb{S} is completely determined by specifying those finite sequences which are nodes of \mathbb{S} . In order for \mathbb{S} to be a tree it must have the following property:

If the finite sequence (s_1, \dots, s_n) is a node on \mathbb{S} , then every finite subsequence (s_1, \dots, s_i) , where $1 \leq i < n$, is also a node on \mathbb{S} .

Note that a subtree \mathbb{S} of \mathbb{T} might have only a finite number of nodes (in which case there are no infinite paths through \mathbb{S}).

Definition 9. Let \mathbb{S} be a subtree of \mathbb{T} . A *path of length n on \mathbb{S}* is a finite sequence (s_1, \dots, s_n) which is a node of \mathbb{S} .

Note that if (s_1, \dots, s_n) is a path of length n on \mathbb{S} , then (s_1, \dots, s_i) is a path of length i on \mathbb{S} for $1 \leq i < n$.

Exercise (König's Infinity Lemma) 10. Let \mathbb{S} be a subtree of the tree \mathbb{T} . Show that if for each $n \in \mathbb{N}$, \mathbb{S} has a path of length n , then \mathbb{S} has an infinite path.

Exercise 11. Use König's Infinity Lemma to prove the Compactness Theorem.

First-Order Logic

The Syntax of a First-Order Language

We start with the symbols of a first-order language. There are two types of symbols: *logical symbols*, and *parameters*.

Logical Symbols:

1. The two symbols (and). The *parentheses* are used for punctuation.
2. \rightarrow and \neg . These are the sentential connective symbols.
3. $v_1, v_2, \dots, v_n, \dots$. This is an enumerable list of symbols called *variables*.
4. \approx . This is the *identity* or *equality* symbol. It may, or may not be present in a particular first-order language.

Parameters

1. \forall . This is the *universal quantifier* symbol.
2. For each $n > 0$ there is a set (possibly empty) of objects called n -ary (or n -place) *relation* (or *predicate* symbols.
3. For each $n > 0$ there is a set (possibly empty) of objects called n -ary (or n -place) *function* symbols.
4. A set (possibly empty) of objects called *constant* symbols.

By a *symbol* we mean, temporarily, either a logical symbol or a parameter

Further requirements.

- 1) \approx is a 2-ary relation symbol.
- 2) We require that there must be at least one relation symbol.
- 3) The symbols are distinct, and no symbol is equal to a finite sequence of other symbols. For example, no 2-ary function symbol is also a 3-ary function symbol, no function symbol is a relation symbol, etc.

Remark. We have not put any requirements on the number of function symbols, relation symbols, or constant symbols. For example, it is possible that for a given first-order language there are three constant symbols, enumerably many function symbols, and uncountably many relation symbols!

Definition. An *expression* is a finite sequence of symbols. A *term sequence* is a finite sequence t_1, \dots, t_n of expressions such that each t_i is either

- a) a variable, or a constant symbol, or
- b) is of the form $f s_1 \cdots s_n$, where f is an n -ary function symbol and each of s_1, \dots, s_n occurs earlier in the sequence.

Definition. An expression t is a *term* if there is a term sequence t_1, \dots, t_k such that $t = t_k$.

Example. Suppose:

f is a 2-ary function symbol;
 g is a 3-ary function symbol;
 c_1 and c_2 are constant symbols.

Then $gfc_1c_2v_3c_1$ is a term since

$$v_3, c_1, c_2, fc_1c_2, gfc_1c_2v_3c_1$$

is a term sequence.

Definition. An expression is an *atomic formula* if it is of the form $Pt_1 \cdots t_k$, where t_1, \dots, t_k are terms, and P is a k -ary relation symbol.

Example. The expression $\approx v_7v_3$ (consisting of three symbols) is an atomic formula since \approx is a 2-ary relation symbol and v_7 and v_3 are terms. If c_1 and c_2 are constant symbols then $\approx c_2c_7$ is an atomic formula.

Definition. A *well-formed sequence* is a finite sequence $\alpha_1, \dots, \alpha_n$ of expressions such that each α_i is either

- a) an atomic well-formed formula, or
- b) is of the form $(\neg\beta)$ or $(\beta \rightarrow \gamma)$, where β and γ occur earlier in the list,
or
- c) is of the form $\forall v_i\beta$, where β occurs earlier in the list.

Definition. The expression α is a *well-formed formula* (wff) if there is a well-formed sequence $\alpha_1, \dots, \alpha_k$ such that $\alpha = \alpha_k$.

Example. The expression $(\neg\forall v_3 \approx v_1v_2)$ is a wff since

$$\approx v_1v_2, \forall v_3 \approx v_1v_2, (\neg\forall v_3 \approx v_1v_2)$$

is a well-formed sequence of expressions.

Models of Sentences**Definition 12.** The structure

- a) \mathfrak{A} is *finite* if $|\mathfrak{A}|$ is finite;
- b) \mathfrak{A} is *infinite* if $|\mathfrak{A}|$ is infinite;
- c) \mathfrak{A} is *countable* if $|\mathfrak{A}|$ is countable;
- d) \mathfrak{A} is *enumerable* if $|\mathfrak{A}|$ is enumerable;
- e) \mathfrak{A} is *uncountable* if $|\mathfrak{A}|$ is uncountable.

Definition 13. Let τ be a sentence, and Σ be a set of sentences of some first-order language.

- a) \mathfrak{A} is a *model* of τ if $\models_{\mathfrak{A}} \tau$.
- b) \mathfrak{A} is a *model* of Σ if \mathfrak{A} is a model of each σ in Σ .
- c) $\text{Mod}(\Sigma)$ is the collection of all models of Σ .

Exercise 14. Let P be a two place relation symbol, and Σ be the following set of sentences:

$$\begin{aligned} & \forall x \forall y \forall z (Pxy \wedge Pyz \longrightarrow Pxz) \\ & \exists x \forall y Pxy \\ & \exists z \forall y Pyz \\ & \exists x \exists y (\neg Pxy \wedge \neg Pyx) \\ & \forall x \forall y (Pxy \wedge Pyx \longrightarrow x \approx y) . \end{aligned}$$

Does Σ have a model? If it does find one. If it doesn't, explain why.**Exercise 15.** Let \mathcal{L} be the first-order language with \approx , and a 2-place predicate symbol P , but no other function or relation or constant symbols.Find a sentence σ of \mathcal{L} such that *σ is true in some infinite structure, but false in every finite structure (i.e. σ has an infinite model, but no finite model).*Explain carefully why the sentence σ you have found has the desired properties.**Exercise 16.** Choose an appropriate first-order language \mathcal{L} and find a set of sentences Σ of \mathcal{L} such that Σ has an uncountable model but no countable model.**Exercise 17.**

- a) Try to find a first-order language \mathcal{L} and a sentence σ of \mathcal{L} such that for each n , the sentence σ is true in some structure \mathfrak{A} with $|\mathfrak{A}| \geq n$, but σ is false in every infinite structure.

- b) Try to find a first-order language \mathcal{L} and a set Σ of sentences of \mathcal{L} such that for each n , Σ has a model in some structure \mathfrak{A} with $|\mathfrak{A}| \geq n$, but Σ has no infinite model.

Definition 18. Suppose $\mathfrak{A} = (A, R)$ is a structure, where R is a binary relation on A . Let $B \subseteq A$, and $b \in A$.

b is an R -minimal member of B , if

- 1) $b \in B$, and
- 2) $\forall a \in B [(a, b) \notin B]$.

Example. Let:

$$\begin{aligned} \mathfrak{R} &= (\mathbb{R}, <); \\ B_1 &= \{x : 0 \leq x < 3\}; \\ B_2 &= \{x : 0 < x < 3\}, \end{aligned}$$

where $<$ is the *less than* relation on \mathbb{R} . Then 0 is an $<$ -minimal member of B_1 , but there is no $<$ -minimal member of B_2 .

Definition 19. The structure $\mathfrak{A} = (A, R)$ (where R is a binary relation on $|A|$) is

- a) *wellfounded* if every non-empty subset of A has an R -minimal member;
- b) *wellordered* if it is wellfounded and ordered.

Example 20.

- a) $\mathfrak{R} = (\mathbb{R}, <)$ is not wellfounded.
- b) $\mathfrak{R}_{\geq 0} = (\{x \in \mathbb{R} : x \geq 0\}, <)$ (where $<$ is the *less than* relation on $\{x \in \mathbb{R} : x \geq 0\}$) is not wellfounded.
- c) $\mathfrak{N} = (\mathbb{N}, <)$ (where $<$ is the *less than* relation on \mathbb{N}) is wellordered.

Theorem 21. *The structure $\mathfrak{A} = (A, R)$ is not wellfounded iff there is an infinite sequence $a_0, a_1, \dots, a_n, \dots$ of members of A such that for all n , $a_{n+1} R a_n$.*

Proof. (\Leftarrow): Suppose that for all n , $a_{n+1} R a_n$. Let $B = \{a_0, \dots, a_n, \dots\}$. Then B has no R -minimal member.

(\Rightarrow): Suppose \mathfrak{A} is not wellfounded. Then there is some $B \neq \emptyset$ which has no R -minimal member. Choose $a_0 \in B$. a_0 is not R -minimal, and so there is some $a_1 \in B$ such that $a_1 R a_0$. Since a_1 is not R -minimal, there is an $a_2 \in B$ such that $a_2 R a_1$. Continuing, we see that for each $n \in B$, there is an $a_{n+1} \in B$ such that $a_{n+1} R a_n$. Thus there is an infinite sequence a_0, \dots, a_n, \dots of members of B such that for all n , $a_{n+1} R a_n$. \square

Question 22. Let \mathcal{L} be the first-order language with \approx , and a 2-place predicate symbol P , but no other function or relation or constant symbols.

- a) Does there exist a sentence σ of \mathcal{L} such that for every structure $\mathfrak{A} = (A, R)$,

$$\models_{\mathfrak{A}} \sigma \text{ iff } \mathfrak{A} \text{ is wellfounded?}$$

(The existence of such a sentence is equivalent to saying that the class of wellfounded structures is an EC class.)

- b) If the answer to (a) is *no*, does there exist a set Σ of sentences of \mathcal{L} such that for every structure $\mathfrak{A} = (A, R)$,

$$\mathfrak{A} \text{ is a model of } \Sigma \text{ iff } \mathfrak{A} \text{ is a wellfounded structure?}$$

(The existence of such a set of sentences is equivalent to saying that the class of wellfounded structures is an EC_{Δ} .)

Definability Within a Structure

Let:

$$\mathfrak{N}_{<} = (\mathbb{N}, <)$$

$$\mathfrak{N}_{+} = (\mathbb{N}, +)$$

$$\mathfrak{N}_{\times} = (\mathbb{N}, \times)$$

$$\mathfrak{N}_{+, \times} = (\mathfrak{N}, +, \times),$$

where $<$ is the ‘less than’ relation on \mathbb{N} , and $+$ and \times are the addition and multiplication functions on \mathbb{N} . (The structure $\mathfrak{N}_{+, \times}$ has in its associated language the symbols $\approx, \dot{+}, \dot{\times}$, but no other predicate, function, or constant symbols.)

Example 23.

- a) $v_1 \dot{<} v_2$ defines the relation $<$ in $\mathfrak{N}_{<}$. But the same wff $v_1 \dot{<} v_2$ also defines the 3-ary relation $\{(m, n, p) : m, n, p \in \mathbb{N} \ \& \ m < n\}$, the 4-ary relation $\{(m, n, p, q) : m, n, p, q \in \mathbb{N} \ \& \ m < n\}$, etc.
- b) $v_2 \dot{<} v_4$ defines the 4-ary relation $\{(m, n, p, q) : m, n, p, q \in \mathbb{N} \ \& \ n < q\}$, the 5-ary relation $\{(m, n, p, q, r) : m, n, p, q, r \in \mathbb{N} \ \& \ n < p\}$, etc.
- c) $\forall v_2 (v_2 \not\approx v_1 \rightarrow v_1 \dot{<} v_2)$ defines the set $\{0\}$.
- d) $\exists v_3 [\forall v_2 (v_2 \not\approx v_3 \rightarrow v_3 \dot{<} v_2) \wedge v_1 \not\approx v_3 \wedge \forall v_4 (v_4 \dot{<} v_1 \rightarrow v_4 \approx v_3)]$ defines $\{1\}$.

Definition 24. Let \mathfrak{A} be a structure for the language \mathcal{L} . The n -ary function $f: A^n \rightarrow A$ is *definable in* \mathfrak{A} if its graph is definable in \mathfrak{A} . For example, the binary function $f: A^2 \rightarrow A$ is definable in \mathfrak{A} if the 3-ary relation

$$\{(a, b, c) : f(a, b) = c\}$$

is definable.

Proposition 25. *Let \mathfrak{A} be a structure for the language \mathcal{L} with $A = |\mathfrak{A}|$.*

- a) \emptyset and A are definable.
- b) The relation $P^{\mathfrak{A}}$ is definable for each predicate symbol P . (In particular, the equality relation $=^A$ on A is definable, assuming \approx is in the language \mathcal{L} .)
- c) The function $f^{\mathfrak{A}}$ is definable, for each function symbol f .
- d) If the relation R is definable, then so is its complement.
- e) If each of the k -ary relations R_1, \dots, R_n is definable, then both the union, $R_1 \cup \dots \cup R_n$, and the intersection, $R_1 \cap \dots \cap R_n$, are definable.
- f) If the $n + 1$ -ary relation R is definable, then so are the n -ary relations Q and S , where

$$Q(\mathbf{x}) \iff \exists y \in A R(\mathbf{x}, y)$$

$$S(\mathbf{x}) \iff \forall y \in A R(\mathbf{x}, y).$$

Exercise 26. Let \mathfrak{A} be a structure for \mathcal{L} and $A = |\mathfrak{A}|$.

- a) Show that each projection function $Pr_i^n: A^n \rightarrow A$ is definable, where $Pr_i^n(a_1, \dots, a_n) = a_i$.
- b) Show that the class of functions definable on \mathfrak{A} is closed under composition. For example, show that if f and g are 1-ary functions which are definable in \mathfrak{A} , then h is also definable, where $h(a) = f(g(a))$.

The projection functions play the same role here that they do in recursive function theory.

Example 27.

- a) \leq is definable in \mathfrak{N}_+ ;
- b) $<$ is definable in \mathfrak{N}_+ ;
- c) $\{2n : n \in \mathbb{N}\}$ (the set of even integers in \mathbb{N}) is definable in \mathfrak{N}_+ .

Question 28.

- a) Is \times definable in \mathfrak{N}_+ ?
- b) Is $+$ definable in \mathfrak{N}_\times ?

Exercise 29.

- a) Show that every initial function is definable in $\mathfrak{N}_{+, \times}$.
- b) Suppose f is an $n + 1$ -ary function which is definable in $\mathfrak{N}_{+, \times}$ and g is defined by unbounded search from f . Show that g is definable in $\mathfrak{N}_{+, \times}$.

Question 30. We know from the above exercises that every initial function is definable in $\mathfrak{N}_{+, \times}$, and the functions definable in $\mathfrak{N}_{+, \times}$ are closed under composition and unbounded search. Is every recursive function definable in $\mathfrak{N}_{+, \times}$?

Exercise 31. Assume that every recursive function is definable in $\mathfrak{N}_{+, \times}$.

- a) Show that every recursively enumerable relation is definable in $\mathfrak{N}_{+, \times}$.
- b) Show that there are relations definable in $\mathfrak{N}_{+, \times}$ which are not recursively enumerable.

Substructures and Reducts

Definition (Substructure). Let $\mathfrak{A} = (A, \dots)$ and $\mathfrak{B} = (B, \dots)$ be structures for the first-order language \mathcal{L} .

\mathfrak{A} is a *substructure* of \mathfrak{B} (written $\mathfrak{A} \subseteq \mathfrak{B}$) if

- 1) $A \subseteq B$;
- 2) for every k -ary relation symbol P and every k -tuple $\bar{a} = (a_1, \dots, a_k)$ of elements of A ,

$$\bar{a} \in P^{\mathfrak{A}} \iff \bar{a} \in P^{\mathfrak{B}}$$

(Note that this is not the same as saying that $P^{\mathfrak{A}} \subseteq P^{\mathfrak{B}}$);

- 3) for every k -ary function symbol f and every k -tuple $\bar{a} = (a_1, \dots, a_k)$ of elements of A ,

$$f^{\mathfrak{A}}(\bar{a}) = f^{\mathfrak{B}}(\bar{a});$$

- 4) for each constant symbol c of \mathcal{L} ,

$$c^{\mathfrak{A}} = c^{\mathfrak{B}}.$$

If \mathfrak{A} is a substructure of \mathfrak{B} we also say that \mathfrak{B} is an *extension* of \mathfrak{A} .

The notion of substructure is used, for example, in Exercises 18, 19 on page 96 of Enderton.

Example. Let

$$\begin{aligned} \mathfrak{N} &= (\mathbb{N}, <^{\mathbb{N}}, +^{\mathbb{N}}, \times^{\mathbb{N}}) \\ \mathfrak{E} &= (\mathbb{E}, <^{\mathbb{E}}, +^{\mathbb{N}}, \times^{\mathbb{E}}). \end{aligned}$$

Then \mathfrak{E} is a substructure of \mathfrak{N} .

Exercise. Let

$$\begin{aligned} \mathfrak{B} &= (B, P^{\mathfrak{B}}), \text{ and} \\ \mathfrak{A} &= (A, P^{\mathfrak{A}}), \end{aligned}$$

where

$$B = \mathbb{N} \text{ and } P^{\mathfrak{B}} = \mathbb{E}$$

$$A = \{0, 1, 2, 3\} \text{ and } P^{\mathfrak{A}} = \{0, 1\} .$$

Is \mathfrak{A} a substructure of \mathfrak{B} ?

Definition (Reduct). Let \mathcal{L} be a first-order language and let \mathcal{L}' be the first-order language obtained from \mathcal{L} by eliminating from \mathcal{L} some of the relation symbols, function symbols, or constant symbols. Let $\mathfrak{A} = (A, \dots)$ be a structure for \mathcal{L} and $\mathfrak{B} = (B, \dots)$ be a structure for \mathcal{L}' . \mathfrak{B} is a *reduct* of \mathfrak{A} if

- 1) $A = B$ (so \mathfrak{A} and \mathfrak{B} have the same universe);
- 2) For each relation symbol P , function symbol f , and constant symbol c of \mathcal{L}' ,

$$P^{\mathfrak{A}} = P^{\mathfrak{B}} ;$$

$$f^{\mathfrak{A}} = f^{\mathfrak{B}} ;$$

$$c^{\mathfrak{A}} = c^{\mathfrak{B}} .$$

Example. Let

$$\mathfrak{N} = (\mathbb{N}, <^{\mathbb{N}}, +^{\mathbb{N}}, \times^{\mathbb{N}}, 0, 1) ;$$

$$\mathfrak{B} = (\mathbb{N}, +^{\mathbb{N}}) .$$

So \mathfrak{N} is a structure for the language \mathcal{L} which has symbols \approx , $<$, $+$, \times , $\mathbf{0}$, and $\mathbf{1}$. And \mathfrak{B} is a structure for the language \mathcal{L}' which has \approx , and $+$. \mathfrak{B} is a reduct of \mathfrak{N} . Note that the numbers 0 and 1 are still in the universe of \mathfrak{B} even though the constant symbols $\mathbf{0}$ and $\mathbf{1}$ are not part of the language \mathcal{L}' .

Graphs

Definition. Let \mathcal{L} be a first-order language with \approx and a binary relation symbol E . A structure $\mathfrak{A} = (A, E^{\mathfrak{A}})$ for \mathcal{L} is a *graph* if:

- a) $E^{\mathfrak{A}}$ is *irreflexive*, i.e. $\forall a \in A (a, a) \notin E^{\mathfrak{A}}$, and
- b) $E^{\mathfrak{A}}$ is *symmetric*, i.e. $\forall a \in A \forall b \in A [(a, b) \in E^{\mathfrak{A}} \implies (b, a) \in E^{\mathfrak{A}}]$.

A graph is *finite* if its universe is finite.

Remark. Let σ be the sentence $\forall x \neg Exx \wedge \forall x \forall y (Exy \implies Eyx)$. Then for every structure \mathfrak{A} ,

$$\mathfrak{A} \in \text{Mod}(\sigma) \text{ iff } \mathfrak{A} \text{ is a graph.}$$

Remark. If $\mathfrak{A} = (A, E^{\mathfrak{A}})$ is a finite graph we can draw a picture of it by drawing a line between each elements a and b just in case $(a, b) \in E^{\mathfrak{A}}$.

Example. Let $\mathfrak{A}_3 = (A_3, E^{\mathfrak{A}_3})$ and $\mathfrak{A}_4 = (A_4, E^{\mathfrak{A}_4})$ be the graphs, where

$$A_3 = \{0, 1, 2, 3\} ;$$
$$E^{2^3} = \{(0, 2), (2, 0), (1, 3), (3, 1), (1, 2), (2, 1)\} ,$$

and

$$A_4 = \{0, 1, 2, 3, 4\} ;$$
$$E_4 = \{(0, 2), (2, 0), (1, 3), (3, 1), (2, 4), (4, 2), (0, 4), (4, 0)\} .$$

We can picture these graphs as follows:

Definition. Let $\mathfrak{A} = (A, E^{\mathfrak{A}})$ be a graph.

a) A *path* between elements a and b of A is a finite sequence a_1, \dots, a_k of elements of A such that

$$a = a_1 \text{ and } b = a_k, \text{ and} \\ \text{for each } i \text{ such that } 0 < i < k, \text{ we have } (a_i, a_{i+1}) \in E^{\mathfrak{A}}.$$

b) \mathfrak{A} is a *connected graph* if for every a and b in A there is some path between a and b .

Remark. The graph \mathfrak{A}_3 is connected, but the graph \mathfrak{A}_4 is not connected.

We can generalize the graphs \mathfrak{A}_3 and \mathfrak{A}_4 as follows:

Definition. For $n > 0$, let $\mathfrak{A}_n = (A_n, E^{\mathfrak{A}_n})$ be the finite structure with

$$A = \{0, \dots, n\}, \text{ and} \\ E^{\mathfrak{A}_n} = \{(i, j) : 0 \leq i, j \leq n \ \& \ |i - j| = 2\} \\ \cup \{(0, n), (n, 0)\} \cup \{(1, n-1), (n-1, 1)\}.$$

Remark. The graph \mathfrak{A}_n is connected iff n is odd, i.e. iff the universe of \mathfrak{A}_n has an even number of members.

Question. Is there a sentence σ of the language for graphs such that for every graph \mathfrak{A} ,

$$\mathfrak{A} \text{ is connected iff } \models_{\mathfrak{A}} \sigma ?$$

Finite Model Theory

For every mathematical statement which talks about all structures, there is a corresponding statement about finite structures. Finite Model Theory is concerned with properties of finite structures. This area has received a lot of interest recently because fundamental open problems in theoretical computer science can be formulated elegantly in terms of finite model theory. A reference is Ebbinghaus and Flum, *Finite Model Theory*, Springer-Verlag, 1995.

There is a question about finite models which corresponds to the last question above:

Question. Is there a sentence σ of the language of graphs such that for every *finite* graph \mathfrak{A} ,

$$\mathfrak{A} \text{ is connected iff } \models_{\mathfrak{A}} \sigma ?$$

Definition. Let \mathcal{L} be a first-order language with \approx , a binary relation symbol $<$, and constant symbols \min and \max . A structure for \mathcal{L} , is a *finite ordered structure* if for some n it is isomorphic to the structure $\mathfrak{A} = (A, <^{\mathfrak{A}}, \min^{\mathfrak{A}}, \max^{\mathfrak{A}})$, where

$A = \{0, \dots, n\}$, and
 $<^{\mathfrak{A}}$ is the usual ‘less than’ relation on A , and
 $\min^{\mathfrak{A}} = 0$ and $\max^{\mathfrak{A}} = n$.

Question. Is there a sentence σ of the language \mathcal{L} with \approx , and $<$, and \min and \max such that for every finite ordered structure \mathfrak{A} ,

$$|\mathfrak{A}| \text{ has an even number of members} \quad \text{iff} \quad \models_{\mathfrak{A}} \sigma ?$$

It turns out that the answers to the above questions about finite structures is given in terms of games.

The Ehrenfeucht Game

We assume in this section that \mathcal{L} is a first-order language that has no function symbols.

Definition. Let $\mathfrak{A} = (A, \dots)$ and $\mathfrak{B} = (B, \dots)$ be two finite structures for \mathcal{L} . The function p is a *partial isomorphism* from \mathfrak{A} to \mathfrak{B} if:

$\text{dom}(p) \subseteq A$ and $\text{ran}(p) \subseteq B$, and
 p is one-to-one, and
for every constant symbol c of \mathcal{L} , we have $p(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$, and
for every k -ary relation symbol P of \mathcal{L} , and k -tuple a_1, \dots, a_k of elements of A which are in the domain of p , we have

$$(a_1, \dots, a_k) \in P^{\mathfrak{A}} \iff (p(a_1), \dots, p(a_k)) \in P^{\mathfrak{B}} .$$

Definition. Let \mathcal{L} be a first-order language which has no function symbols, and $\mathfrak{A} = (A, \dots)$ and $\mathfrak{B} = (B, \dots)$ be finite structures for this language. Let $\bar{a} = a_1 \dots a_s$ be an s -tuple of elements of A , and $\bar{b} = b_1 \dots b_s$ be an s -tuple of elements of B , we write $\bar{a} \mapsto \bar{b}$ as an abbreviation for the relation p such that

- a) $p(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$ for each constant c of \mathcal{L} , and
- b) $p(a_i) = b_i$ (more precisely, $(a_i, b_i) \in p$) for $1 \leq i \leq s$.

p might not be a function. However, if the a_1, \dots, a_s are distinct then p is a function.

Definition (The Ehrenfeucht Game). Let:

- a) \mathfrak{A} and \mathfrak{B} be structures of \mathcal{L} ;
- b) \bar{a} be a tuple from A and \bar{b} be a tuple of the same length from B ;
- c) $m \in \mathbb{N}$.

The Ehrenfeucht game $G_m(\mathfrak{A}, \bar{a}, \mathfrak{B}, \bar{b})$ has two players, the *spoiler* and the *duplicator*. The players alternate, each making m moves in a play of the game. In the i -th move the spoiler first selects either the structure \mathfrak{A} or the structure \mathfrak{B} , and chooses an element of the domain of the chosen structure. The duplicator then responds with its i -th move by choosing an element of the domain of the other structure. At the end of m moves of each player, two sequences of length m have been constructed; a sequence $\bar{e} = e_1, \dots, e_m$ of elements from A , and a sequence $\bar{f} = f_1, \dots, f_m$ of elements from B .

- a) The duplicator *wins* the *play* of the game if the relation $\bar{a}\bar{e} \mapsto \bar{b}\bar{f}$ is a partial isomorphism from \mathfrak{A} to \mathfrak{B} .
- b) The spoiler *wins* the *play* otherwise.
- c) A player (either the duplicator or spoiler) has a *winning strategy* if he can win each play (by following his ‘strategy’) regardless of the moves of the other player. In this case we say that that player *wins the game* $G_m(\mathfrak{A}, \bar{a}, \mathfrak{B}, \bar{b})$.
- d) If \bar{a} and \bar{b} are the empty sequence we denote the game by $G_m(\mathfrak{A}, \mathfrak{B})$.
- e) If $m = 0$, then there are no plays of the game and the duplicator wins iff $\bar{a} \mapsto \bar{b}$ is a partial isomorphism from \mathfrak{A} to \mathfrak{B} .

Definition. For a first-order wff ϕ , the quantifier rank of ϕ , written $\text{qr}(\phi)$, is the number of occurrences of the universal quantifier symbol \forall which occur in ϕ . (If ϕ is written in abbreviated form with both \forall and \exists , the quantifier rank of ϕ would be the total number of occurrences of \forall and \exists .)

Definition. For structures \mathfrak{A} and \mathfrak{B} of a first-order language \mathcal{L} , we write $\mathfrak{A} \equiv_m \mathfrak{B}$ if for every sentence σ with $\text{qr}(\sigma) \leq m$,

$$\models_{\mathfrak{A}} \sigma \text{ iff } \models_{\mathfrak{B}} \sigma .$$

The main result is:

Theorem. *Let \mathfrak{A} and \mathfrak{B} be structures for \mathcal{L} . The duplicator wins the game $G_m(\mathfrak{A}, \mathfrak{B})$ iff $\mathfrak{A} \equiv_m \mathfrak{B}$.*

Example. Let $\mathfrak{A} = (\mathbb{R}, <^{\mathbb{R}})$ and $\mathfrak{B} = (\mathbb{Q}, <^{\mathbb{Q}})$. Then for each m , the duplicator wins the game $G_m(\mathfrak{A}, \mathfrak{B})$. It follows that for each m we have $(\mathbb{R}, <^{\mathbb{R}}) \equiv_m (\mathbb{Q}, <^{\mathbb{Q}})$, and hence $(\mathbb{R}, <^{\mathbb{R}}) \equiv (\mathbb{Q}, <^{\mathbb{Q}})$.

Later we shall get two other proofs of this fact!

Definition. Let \mathcal{L} be the first-order language with \approx , and a 2-ary relation symbol $<$, and constant symbols \min and \max . A structure $\mathfrak{A} = (A, <^A, \min^A, \max^A)$ is an *finite, ordered* structure if \mathfrak{A} is a finite structure and $<^A$ is a linear ordering on A , with \min^A the smallest member and \max^A the largest member. Each finite, ordered structure is, for some n , isomorphic to the structure $\mathfrak{A}_n = (A_n, <^A, \min^{A_n}, \max^{A_n})$, where

$$\begin{aligned} A_n &= \{0, \dots, n\}; \\ <^{A_n} &\text{ is the 'less than' relation on } A_n; \\ \min^{A_n} &= 0 \text{ and } \max^{A_n} = n. \end{aligned}$$

Proposition. *For a given m , if n and k are bigger than 2^m , then the duplicator wins the game $G_m(\mathfrak{A}_n, \mathfrak{A}_k)$, and so $\mathfrak{A}_n \equiv_m \mathfrak{A}_k$.*

Corollary. *Let \mathcal{L} be the language for ordered structures. There does not exist a sentence σ of \mathcal{L} such that for every finite ordered structure \mathfrak{A} ,*

$$\models_{\mathfrak{A}} \sigma \text{ iff } |\mathfrak{A}| \text{ has an even number of members.}$$

Proof. Suppose there is such a sentence σ . Let $m = \text{qr}(\sigma)$. Choose n odd and k even, both larger than 2^m . Then $\models_{\mathfrak{A}_n} \sigma$ since n is odd, and $\models_{\mathfrak{A}_k} \neg\sigma$ since k is even. But $\mathfrak{A}_n \equiv_m \mathfrak{A}_k$, and so

$$\models_{\mathfrak{A}_n} \sigma \text{ iff } \models_{\mathfrak{A}_k} \sigma,$$

which is a contradiction. □

Exercise.

For each of the following pairs \mathcal{A} and \mathcal{B} of graphs, find the largest m such that the duplicator wins the game $G_m(\mathcal{A}, \mathcal{B})$. Also find the largest m' such that $\mathcal{A} \equiv_{m'} \mathcal{B}$. What is the relationship between m and m' ?

a) \mathcal{A} \mathcal{B}

b) \mathcal{A} \mathcal{B}

c) \mathcal{A} \mathcal{B}

d) \mathcal{A} \mathcal{B}

Example. Let $\tau = \emptyset$, and $\mathcal{A} = (A)$ and $\mathcal{B} = (B)$ be τ -structures. If $\|A\| \geq m$ and $\|B\| \geq m$ then the duplicator wins $G_m(\mathcal{A}, \mathcal{B})$.

Example.

- a) Let $\mathcal{A} = (A, <^A, \min^A, \max^A)$ and $\mathcal{B} = (B, <^B, \min^B, \max^B)$ be the ordered structures with $A = \{0, 1, 2, 3\}$ and $B = \{0, 1, 2, 3, 4\}$, and the natural orderings. Then the spoiler wins $G_2(\mathcal{A}, \mathcal{B})$.
- b) Let $\mathcal{A} = (A, <^A, \min^A, \max^A)$ and $\mathcal{B} = (B, <^B, \min^B, \max^B)$ be the ordered structures with $A = \{0, 1, 2, 3, 4\}$ and $B = \{0, 1, 2, 3, 4, 5\}$, and the natural orderings. Then the duplicator wins $G_2(\mathcal{A}, \mathcal{B})$.

Lemma.

- a) If $\mathcal{A} \cong \mathcal{B}$ then the duplicator wins $G_m(\mathcal{A}, \mathcal{B})$ for each m .
- a) If the duplicator wins $G_{m+1}(\mathcal{A}, \mathcal{B})$ and $\|A\| \leq m$ then $\mathcal{A} \cong \mathcal{B}$.

Lemma. *Let:*

\mathcal{A} and \mathcal{B} be τ -structures,
 $\bar{a} \in A^s$, $\bar{b} \in B^s$, and
 $m > 0$.

- a) The duplicator wins $G_0(\mathcal{A}, \bar{a}, \mathcal{B}, \bar{b})$ iff $\bar{a} \mapsto \bar{b}$ is a partial isomorphism.
- b) The duplicator wins $G_m(\mathcal{A}, \bar{a}, \mathcal{B}, \bar{b})$ iff

$$\begin{aligned} & \forall a \in A \exists b \in B \text{ [the duplicator wins } G_{m-1}(\mathcal{A}, \bar{a}a, \mathcal{B}, \bar{b}b)] \text{ \&} \\ & \forall b \in B \exists a \in A \text{ [the duplicator wins } G_{m-1}(\mathcal{A}, \bar{a}a, \mathcal{B}, \bar{b}b)] . \end{aligned}$$

- c) If the duplicator wins $G_m(\mathcal{A}, \bar{a}, \mathcal{B}, \bar{b})$ and $m' < m$, then the duplicator wins $G_{m'}(\mathcal{A}, \bar{a}, \mathcal{B}, \bar{b})$.

Definition. Let \mathcal{A} be a τ -structure, $\bar{a} \in A^s$, and $\bar{v} = v_1, \dots, v_s$. The formulas $\phi_{\mathcal{A}, \bar{a}}^m(\bar{v})$ are defined by the recursion:

$$\begin{aligned} \phi_{\mathcal{A}, \bar{a}}^0(\bar{v}) &= \bigwedge \{ \phi(\bar{v}) : (\phi \text{ is atomic or negated atomic}) \ \& \ \mathcal{A} \models \phi[\bar{a}] \} ; \\ \phi_{\mathcal{A}, \bar{a}}^m(\bar{v}) &= \bigwedge \{ \exists v_{s+1} \phi_{\mathcal{A}, \bar{a}a}^{m-1}(\bar{v}, v_{s+1}) : a \in A \} \wedge \forall v_{s+1} \bigvee \{ \phi_{\bar{a}a}^{m-1}(\bar{v}, v_{s+1}) : a \in A \} . \end{aligned}$$

In order for the above sentences $\phi_{\mathcal{A}, \bar{a}}^m$ to be first-order, we need to know that the above conjunctions are actually the conjunctions of finitely many formulas. So the following lemma should really be proved simultaneously with the above definition.

Lemma. For all m and s ,

$$\{\phi_{\mathcal{A},\bar{a}}^m : \mathcal{A} \text{ is a } \tau\text{-structure \& } \bar{a} \in A^s\} \text{ is finite .}$$

When \mathcal{A} is fixed, we sometimes write $\phi_{\bar{a}}^m$ for $\phi_{\mathcal{A},\bar{a}}^m$.

Definition. $\text{Part}(\mathcal{A}, \mathcal{B})$ is the set of partial isomorphisms from \mathcal{A} to \mathcal{B} .

Lemma.

- a) $\text{qr}(\phi_{\mathcal{A},\bar{a}}^m) = m$;
- b) $\mathcal{A} \models \phi_{\mathcal{A},\bar{a}}^m[\bar{a}]$;
- c) For every \mathcal{B} and every $\bar{b} \in B$,

$$\mathcal{B} \models \phi_{\mathcal{A},\bar{a}}^0[\bar{b}] \text{ iff } \bar{a} \mapsto \bar{b} \in \text{Part}(\mathcal{A}, \mathcal{B}) .$$

Theorem (Ehrenfeucht). The following are equivalent:

- a) the duplicator wins $G_m(\mathcal{A}, \bar{a}, \mathcal{B}, \bar{b})$;
- b) $\mathcal{B} \models \phi_{\mathcal{A},\bar{a}}^m[\bar{b}]$;
- c) for every ϕ with $\text{qr}(\phi) \leq m$,

$$\mathcal{A} \models \phi[\bar{a}] \text{ iff } \mathcal{B} \models \phi[\bar{b}] .$$

Corollary. The following are equivalent:

- a) The duplicator wins $G_m(\mathcal{A}, \mathcal{B})$;
- b) $\mathcal{B} \models \phi_{\mathcal{A}}^m$;
- c) $\mathcal{A} \equiv_m \mathcal{B}$.

Corollary. If $\|A\| \leq m$, then for all \mathcal{B} ,

$$\mathcal{B} \models \phi_{\mathcal{A}}^{m+1} \text{ iff } \mathcal{A} \cong \mathcal{B} .$$

Theorem. Let $\phi(v_1, \dots, v_s)$ be a formula with $\text{qr}(\phi) \leq m$. Then

$$\models \phi \longleftrightarrow \bigvee \{\phi_{\mathcal{A},\bar{a}}^m : (\mathcal{A} \text{ is a structure}) \& \bar{a} \in A \& \mathcal{A} \models \phi[\bar{a}]\} .$$

Theorem. Let K be a class of τ -structures. The following are equivalent:

- a) K is not axiomatizable in first-order logic;
- b) for each m , there are finite structures \mathcal{A} and \mathcal{B} such that

$$\mathcal{A} \in K, \mathcal{B} \notin K, \text{ and } \mathcal{A} \equiv_m \mathcal{B} .$$

Completeness and Undecidability: A Preview

Assume that \mathcal{L} is a first-order language with enumerably many symbols, and the additional property that the set of wffs is effectively decidable. That is, there is an algorithm for deciding whether a given string of symbols of \mathcal{L} is a wff.

Question 31.

- a) Is the set of valid wffs of \mathcal{L} effectively enumerable?
- b) If the answer to (a) is 'yes' is the set of valid wffs effectively decidable?
- c) Does the answer to (a) and (b) depend upon the language \mathcal{L} ?

Definition 32. Let \mathfrak{A} be a structure for \mathcal{L} . Let $Th(\mathfrak{A})$ be the set of all sentences of \mathcal{L} which are true in \mathfrak{A} , that is

$$Th(\mathfrak{A}) = \{\sigma : \sigma \text{ is a sentence of } \mathcal{L} \ \& \ \models_{\mathfrak{A}} \sigma\} .$$

Question 33. Is $Th(\mathfrak{A})$ effectively decidable? Does the answer depend upon \mathfrak{A} ?

Exercise 34. Suppose that $Th(\mathfrak{A})$ is effectively enumerable. Show that $Th(\mathfrak{A})$ is effectively decidable.

The following is a summary of the idea used to answer Question 30 (a). An effectively decidable set A of logical axioms of \mathcal{L} is presented, together with a rule of inference. A *proof* is then defined as a finite sequence $\alpha_1, \dots, \alpha_n$ of wffs such that each α_i is either a member of A or is obtained from wffs in the sequence by the rule of inference. The set of proofs is then effectively decidable. A *theorem* is a wff ϕ such that there exists a proof $\alpha_1, \dots, \alpha_n$ such that $\phi = \alpha_n$. By effectively listing all proofs, we see that the set of theorems is effectively enumerable. Let $\vdash \phi$ mean that ϕ is a theorem. So $\{\phi : \vdash \phi\}$ is effectively enumerable.

Theorem (Gödel Completeness) For every wff ϕ of \mathcal{L} ,

$$\vdash \phi \text{ iff } \models \phi .$$

It follows that the answer to Question 30 (a) is 'yes'!

Quotient Structures

Let k be a positive integer and \equiv_k be the binary relation on \mathbb{N}

$$m \equiv_k n \iff k \mid (m - n).$$

If $m \equiv_k n$, we say that m and n are *congruent modulo k* .

Exercise 35. Show that \equiv_k is an equivalence relation on \mathbb{N} , where the equivalence classes are

$$\begin{aligned} [0] &= \{0, k, 2k, 3k, 4k, \dots\} \\ [1] &= \{1, k+1, 2k+1, 3k+1, 4k+1, \dots\} \\ &\vdots \\ [k-1] &= \{k-1, 2k-1, 3k-1, 4k-1, \dots\}. \end{aligned}$$

For example, if $k = 3$, then the equivalence classes are

$$\begin{aligned} [0] &= \{0, 3, 6, 9, \dots\} \\ [1] &= \{1, 4, 7, 10, \dots\} \\ [2] &= \{2, 5, 8, 11, 14, \dots\}. \end{aligned}$$

Let \mathcal{L} be the first order language with function symbols $\dot{+}$ and $\dot{\times}$, and a 2-place predicate symbol E . Let \mathfrak{N} be the structure

$$\mathfrak{N} = (\mathbb{N}, +, \times, \equiv_k)$$

for \mathcal{L} , where $E^{\mathfrak{N}} = \equiv_k$.

Exercise 36. Show that if

$$\begin{aligned} m_1 &\equiv_k m_2, \text{ and} \\ n_1 &\equiv_k n_2, \end{aligned}$$

then

$$\begin{aligned} m_1 + n_1 &\equiv_k m_2 + n_2, \text{ and} \\ m_1 \times n_1 &\equiv_k m_2 \times n_2. \end{aligned}$$

Conclude that \equiv_k is a congruence relation for \mathfrak{N} .

Let \mathfrak{N}/E be the quotient structure. For example, let $k = 3$. Then

$$|\mathfrak{N}/E| = \{[0], [1], [2]\},$$

and $\dot{+}^{\mathfrak{N}/E}$ is the function which satisfies

$$\begin{aligned} [1] \dot{+}^{\mathfrak{N}/E} [2] &= [0] \\ [2] \dot{+}^{\mathfrak{N}/E} [2] &= [1] , \end{aligned}$$

etc., and

$$[2] \dot{\times}^{\mathfrak{N}/E} [2] = [1] ,$$

etc. The function $h: \mathbb{N} \rightarrow \{[0], [1], [2]\}$, where $h(n) = [n]$, is a homomorphism of \mathfrak{N} onto \mathfrak{N}/E .

Observe that

$$\begin{aligned} ([m], [n]) \in E^{\mathfrak{N}/E} &\iff (m, n) \in E^{\mathfrak{N}} \\ &\iff m \equiv_3 n \\ &\iff [m] = [n] , \end{aligned}$$

i.e. $E^{\mathfrak{N}/E}$ is the equality relation on the universe $\{[0], [1], [2]\}$ of the quotient structure.

Exercise 37. Let

$$\mathfrak{N}_{<} = (\mathbb{N}, +, \times, \equiv_3, <) .$$

Is \equiv_3 a congruence relation for $\mathfrak{N}_{<}$?