

7.1 Structures and Satisfaction

Sections 5.3 to 6.4 derived a good deal, and some searches stalled. The converse of Proposition 5.6.2 could not be finished: every attempt ran into a side condition. The converse of Proposition 5.6.1 stalled in the same way, and so did the search for $\exists x P(x) \vdash P(c)$ in Exercise (C2: A witness is not a name, used later). But a search that fails proves nothing, since a cleverer derivation might exist. Chapter 4 met the same problem for the connectives and solved it with meaning: a row of a truth table that makes the premises true and the conclusion false rules out every derivation at once (Corollary 4.7.3). This section builds the meaning of a quantified formula: a **model**, a domain of objects with a meaning for every symbol. A model that makes the premises true and the conclusion false is a **counter-model**, the exact form of the counter-readings used since Section 5.3, and Section 7.2 uses it to settle the stalled searches.

Remark (Looking back). Section 4.4 gave each formula of Chapter 4 a truth value by recursion on the formula, one clause per connective, under a row. This section does the same for first-order formulas, with a structure and an assignment in place of a row. Clauses (S3)–(S5) below are Chapter 4’s clauses for $\perp$ and the connectives, read as conditions for satisfaction instead of truth under a row; (S6) and (S7), for the quantifiers, are the new ones.

7.1.1 Why a row is not enough

Chapter 4’s semantics gave each atom a truth value and computed the rest. Try it on the stalled sequent of Proposition 5.6.2’s converse,

$$\forall y \exists x R(x, y) \vdash \exists x \forall y R(x, y).$$

Neither side has a main connective among $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$: each is a quantified formula, and a truth table would have to treat each as one atom. The sequent would become $p \vdash q$, refuted by the row $p = \text{T}, q = \text{F}$. That “refutation” is worthless. The same reading refutes $\forall x P(x) \vdash \exists x P(x)$, which is derivable (Proposition 5.5.1). Treating a quantified formula as an atom throws away exactly the structure that the quantifier rules use.

So what does make $\forall y \exists x R(x, y)$ true or false? Section 5.1 read such sentences informally, **over a domain**: “for every y in the domain there is an x in the domain with $R(x, y)$ ”. To decide it we need three things.

- **A domain**: the objects the variables range over. Over $\mathbb{Z}$ the sentence may be true and over a three-element set false.
- **A meaning for each symbol**. R is a predicate symbol, a name with two slots and no meaning of its own (Section 5.2). Read as $>$ on $\mathbb{Z}$, the sentence says “every integer has an integer above it”, which is true; read as $<$ on $\mathbb{N}$, it says “every natural number has a natural number below it”, which is false at $y = 0$.
- **Values for the free variables**, if there are any. The formula $\exists y (y < x)$ is neither true nor false on $\mathbb{N}$ until we know which number x is.

The first two together are a **structure**; the third is an **assignment**. A structure with an assignment decides every formula, by clauses that read the quantifiers as Section 5.1 read them. That is all this section does; Section 7.2 uses the result.

7.1.2 Structures

Situate the notion first. A formula is built from **symbols**: constants such as c and 0 , function symbols such as f and $+$, predicate symbols such as P , R and $<$, each function and predicate symbol with its number of argument places, its **arity** (Section 5.2, Definition 5.2.1 and Definition 5.2.2). The symbol $=$ is special: it is part of the logic, with rules of its own (Definition 5.7.1), and it will always mean the same thing. A structure gives every other symbol a meaning.

A structure is a non-empty set of objects together with a meaning for each symbol: an object for each constant, an operation for each function symbol, and a relation for each predicate symbol.

Definition 7.1.1. Structure

Fix a collection of constants, function symbols and predicate symbols, each function and predicate symbol with its arity $n \geq 1$. A **structure** $\mathfrak{M}$ (also called a **model**) for these symbols consists of

- (M1) a **non-empty** set D , the **domain** of $\mathfrak{M}$;
- (M2) for each constant c , an element $c^{\mathfrak{M}} \in D$;
- (M3) for each function symbol f of arity n , a function $f^{\mathfrak{M}}: D^n \rightarrow D$, defined at **every** n -tuple of elements of D ;
- (M4) for each predicate symbol R of arity n , a set $R^{\mathfrak{M}}$ of n -tuples of elements of D .

The symbol $=$ is not interpreted by $\mathfrak{M}$: in every structure it means **identity**, “is the same element of D as”.

In words. (M1) says what the variables range over, and insists that there is at least one object. (M2) says which object each name names; two constants may name the same object. (M3) makes each function symbol an operation that can be applied to **any** objects of the domain and always returns an object of the domain. (M4) makes each predicate symbol a relation: $R^{\mathfrak{M}}$ is the set of tuples for which R “holds”. A one-place predicate symbol P becomes a subset $P^{\mathfrak{M}} \subseteq D$, a two-place one a set of ordered pairs, and so on. Here D^n is the set of n -tuples $(d_1, \dots, d_n)$ of elements of D ; this section uses only their everyday meaning.

We usually describe a structure in one line, naming the domain and the meaning of each symbol, and write “ $\mathfrak{M} = (\mathbb{Z}, <)$ ” for the structure with domain $\mathbb{Z}$ in which the symbol $<$ means the usual order. When a symbol has its usual meaning on a familiar set, we say so and do not write $<^{\mathfrak{M}}$. When the domain is followed by a list of symbols, a semicolon separates the domain from the list, as in $(\mathbb{R}; 1, +, \cdot)$: each listed symbol has its usual meaning on that set, and only the listed symbols belong to the structure. With a single symbol we may also write a comma, as in $(\mathbb{Z}, <)$.

Examples, simplest first.

- *One element.* $D = \{a\}$, with $P^{\mathfrak{M}} = \{a\}$. This is the degenerate structure. Every variable must take the value a , so “for every element” and “there is an element” both mean “ a ”. It will reappear as a test case: a sentence true here and false in some other structure cannot be derivable (by the soundness statement in Section 7.2), and some of the chapter’s non-derivable sequents need at least two elements to refute.
- *Two elements, one relation.* $D = \{a, b\}$, with $R^{\mathfrak{M}} = \{(a, a), (b, b)\}$: the relation “is the same as”, written out as a set of pairs. A finite structure can be given completely by listing.
- *The integers with their order.* $D = \mathbb{Z}$, with $R^{\mathfrak{M}} = \{(x, y) \in \mathbb{Z}^2 : x > y\}$. Here the relation is described by a condition rather than a list.

- *Arithmetic.* $D = \mathbb{R}$, with the constants 0 and 1 naming zero and one, the function symbols $+$, $\cdot$ and two-place $-$ meaning addition, multiplication and subtraction, the one-place symbols $-$, $(\cdot)^2$ and $|\cdot|$ meaning negation, squaring and absolute value, and $<$, $\leq$, $>$, $\geq$ the usual order. Extended by the other familiar symbols with their usual meanings (numerals, halving, the predicates $\in A$, division with its value at a zero denominator fixed arbitrarily as in the non-example below, and a function symbol for each sequence or function an example uses), this is the structure in which the analysis examples of Chapters 5 and 6 have been read all along.

🕒 **Second reading** · *A non-example, and the repair used for $1/n$*

Non-example by minimal change. Take the arithmetic structure and add a one-place function symbol g , meant to be read as $x \mapsto 1/x$. This is **not** a structure: (M3) asks for a function defined at every element of D , and $1/0$ is not defined. Everything else in the description is fine; exactly clause (M3) fails, at the single point 0. There are two repairs. Shrink the domain to $(0, \infty)$, where $1/x$ is always defined; or keep $\mathbb{R}$ and give g any value at 0, say $g^{\mathfrak{M}}(0) := 0$. The second repair is the one this book uses for terms such as $1/n$: the value at 0 is fixed arbitrarily and is never looked at, because such terms are used only under a bound $n \in \mathbb{Z}_{>0}$ (as in Exercise C3 of Section 7.2), exactly as a sequence a_n was read in Section 5.1 as a function symbol whose values off $\mathbb{N}$ are never used.

The domain is non-empty because of Proposition 5.5.1; the subsection “Why domains are non-empty” explains why.

🕒 **Second reading** · *Why total functions, and why identity*

- *Total functions.* Clause (M3) is what makes every term name an object: the term $g(0)$ must have a value, or the formula $g(0) = g(0)$, which $=I$ derives from nothing, would have no truth value.
- *Identity for $=$.* The rule $=E$ replaces equals by equals **everywhere**. If a structure could read $=$ as some other relation, $=E$ would fail. On $\mathbb{Z}$, read $x = y$ as “ x and y have the same parity” and P as “is positive”: then $-2 = 2$ and $P(2)$ would both hold, while $P(-2)$, which $=E$ derives from them, fails. With $=$ fixed as identity, no such reading exists.

Warning. A structure must interpret **every** symbol that occurs in the formulas at hand. “Take $D = \mathbb{Z}$ ” is not yet a structure for $\forall y \exists x R(x, y)$: until R is given a meaning, the sentence has no truth value, and the same domain makes it true with R read as $>$ and false with R read as “ $x \cdot x = y$ ” (at $y = 2$). Always write the domain **and** the meaning of each symbol.

7.1.3 Drawing a finite structure

A finite structure whose symbols are one- and two-place predicate symbols can be drawn, as Section 5.1 drew its worlds. Each element is a dot. Beside a dot stand its name and the one-place

predicates it satisfies. For a two-place relation R , an arrow runs from d to e for each pair (d, e) in $R^{\mathfrak{M}}$, and a loop at d records the pair (d, d) . Deciding a sentence in a drawn structure is then reading the picture, dot by dot.

When the relation is a partial order $\leq$, arrows crowd the picture, and a lighter drawing is used. Section 7.4 defines partial orders; here it is enough that $\leq$ is reflexive and transitive and never holds both ways between two different elements. The **Hasse diagram** of such a structure puts d below e when $d \leq e$ and $d \neq e$, and draws a line only for the pairs with nothing strictly between them. The loops, and the pairs that follow by chaining lines upwards, are understood. In the right picture below, $1 \leq 6$ holds although no line joins 1 and 6: the chain 1, 2, 6 runs upwards.

A picture is a structure only when it says what every symbol means: the caption names the domain and each symbol. Figure F draws two structures. Left: $D = \{a, b, c\}$, $P^{\mathfrak{M}} = \{a\}$, $R^{\mathfrak{M}} = \{(a, b), (b, b), (c, a)\}$. Right: $D = \{1, 2, 3, 6\}$, with $x \leq y$ read as “ x divides y ”.

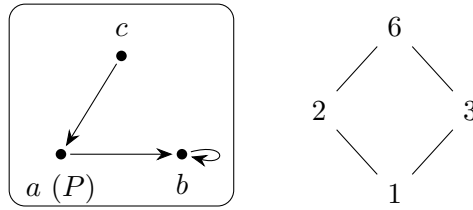


Figure F. An arrow diagram (left) and a Hasse diagram (right).

Quick check. In the left structure of Figure F, decide $\forall x \exists y R(x, y)$ and $\exists y \forall x R(x, y)$.

Solution. The first is true: every dot has an arrow out of it ($a \rightarrow b$, $b \rightarrow b$, $c \rightarrow a$). The second is false: no dot receives an arrow from all three. The dot b receives none from c , the dot a none from a or b , and the dot c none at all.

7.1.4 Assignments and the value of a term

A structure names the objects that **constants** stand for. Variables are different: their whole purpose is to range over the domain, so a structure gives them no fixed value. A separate object records what they stand for at a given moment.

Definition 7.1.2. Assignment

Let $\mathfrak{M}$ be a structure with domain D . An **assignment** in $\mathfrak{M}$ is a function ℓ that gives **every** variable x a value $\ell(x) \in D$. For an assignment ℓ , a variable x and an element $d \in D$, the assignment $\ell[x \mapsto d]$ is the one that gives x the value d and every other variable the value ℓ gives it:

$$\ell[x \mapsto d](y) := \begin{cases} d & \text{if } y \text{ is the variable } x, \\ \ell(y) & \text{if } y \text{ is any other variable.} \end{cases}$$

In words: an assignment is a choice of value for every variable at once, and $\ell[x \mapsto d]$ changes that choice at the one variable x , leaving the others alone. The assignment $\ell[x \mapsto d]$ is what the quantifiers will use: “for every x ” will mean “for every $d \in D$, with x set to d ”. Changes can be

stacked: $\ell[x \mapsto d][y \mapsto e]$ first sets x to d and then y to e . When x and y are different variables the order does not matter, since each change touches only its own variable; when they are the same variable, the later change wins.

There are always assignments, because D is non-empty: pick any $d_0 \in D$ and let every variable have the value d_0 . Like a valuation in Chapter 4 (Definition 4.4.1), an assignment fixes infinitely many values, and a formula will only ever look at finitely many of them.

With a structure and an assignment, every term names an element of the domain. Terms are built by the clauses of Definition 5.2.1, so the value is defined by recursion on terms, in the pattern of Theorem 4.3.4.

The value of a term is computed from its variables and constants upwards, one function symbol at a time.

For a structure $\mathfrak{M}$ and an assignment ℓ , the **value** $t^{\mathfrak{M},\ell} \in D$ of a term t is:

- $x^{\mathfrak{M},\ell} := \ell(x)$ for a variable x ;
- $c^{\mathfrak{M},\ell} := c^{\mathfrak{M}}$ for a constant c ;
- $f(t_1, \dots, t_n)^{\mathfrak{M},\ell} := f^{\mathfrak{M}}(t_1^{\mathfrak{M},\ell}, \dots, t_n^{\mathfrak{M},\ell})$ for a function symbol f of arity n .

The third clause is where (M3) is needed: $f^{\mathfrak{M}}$ must accept every tuple the terms $t_1, \dots, t_n$ might produce.

For example, in the arithmetic structure on $\mathbb{R}$, let $\ell(x) = 2$ and $\ell(y) = -3$. The term $x \cdot x + y$, which is $+(\cdot(x, x), y)$ written in the usual way, has value $2 \cdot 2 + (-3) = 1$. Under $\ell[y \mapsto 1]$ it has value $2 \cdot 2 + 1 = 5$, and under $\ell[x \mapsto 0]$ it has value -3 . The term $1 + 1$ has value 2 under every assignment, since it contains no variable.

7.1.5 Satisfaction

Now the central definition. We want to say when a structure and an assignment make a formula true. Formulas are built by the clauses of Definition 5.2.2, so we define it by recursion, one clause per way of building a formula, and each clause reads its connective or quantifier as Chapter 4 and Section 5.1 read it.

A formula is satisfied when the objects its terms name stand in the relations its predicates name, combined by the connectives as in a truth table, with “for every x ” and “there is an x ” ranging over the domain.

Definition 7.1.3. Satisfaction

Let $\mathfrak{M}$ be a structure with domain D and ℓ an assignment in $\mathfrak{M}$. The relation $\mathfrak{M}, \ell \models \varphi$, read “ $\mathfrak{M}$ **satisfies** φ under ℓ ”, is defined by recursion on the formula φ :

- (S1) $\mathfrak{M}, \ell \models R(t_1, \dots, t_n)$ if $(t_1^{\mathfrak{M},\ell}, \dots, t_n^{\mathfrak{M},\ell}) \in R^{\mathfrak{M}}$;
- (S2) $\mathfrak{M}, \ell \models t_1 = t_2$ if $t_1^{\mathfrak{M},\ell}$ and $t_2^{\mathfrak{M},\ell}$ are the **same** element of D ;
- (S3) $\mathfrak{M}, \ell \models \perp$ **never** holds;
- (S4) $\mathfrak{M}, \ell \models \neg\varphi$ if $\mathfrak{M}, \ell \models \varphi$ does not hold;
- (S5) $\mathfrak{M}, \ell \models \varphi \wedge \psi$ if both $\mathfrak{M}, \ell \models \varphi$ **and** $\mathfrak{M}, \ell \models \psi$; $\mathfrak{M}, \ell \models \varphi \vee \psi$ if at least one of them holds; $\mathfrak{M}, \ell \models \varphi \rightarrow \psi$ unless $\mathfrak{M}, \ell \models \varphi$ holds and $\mathfrak{M}, \ell \models \psi$ does not; $\mathfrak{M}, \ell \models \varphi \leftrightarrow \psi$ if both hold or neither does;
- (S6) $\mathfrak{M}, \ell \models \forall x \varphi$ if $\mathfrak{M}, \ell[x \mapsto d] \models \varphi$ **for every** $d \in D$;

(S7) $\mathfrak{M}, \ell \models \exists x \varphi$ if $\mathfrak{M}, \ell[x \mapsto d] \models \varphi$ for **at least one** $d \in D$.

When $\mathfrak{M}, \ell \models \varphi$ does not hold we write $\mathfrak{M}, \ell \not\models \varphi$. For a list Γ of formulas, $\mathfrak{M}, \ell \models \Gamma$ means that $\mathfrak{M}, \ell \models \gamma$ for **every** formula γ of Γ .

Read it clause by clause. (S1): an atomic formula is satisfied exactly when the values of its terms form a tuple in the relation that interprets its predicate symbol. (S2): an equation is satisfied when both sides name the same object; this is where “= means identity” enters. (S3)–(S5) are the $\perp$ half of (V1) and clauses (V2)–(V6) of Chapter 4’s Definition 4.4.2, with “is true under v ” replaced by “is satisfied by $\mathfrak{M}$ under ℓ ”: $\perp$ is never satisfied, $\neg$ flips, $\wedge$ needs both, $\vee$ needs one, $\rightarrow$ fails only with a satisfied hypothesis and an unsatisfied conclusion, $\leftrightarrow$ needs agreement. (S6) and (S7) are the new clauses. To decide $\forall x \varphi$, we run through the elements d of the domain, set x to d and leave every other variable alone, and ask whether φ is satisfied each time. For $\exists x \varphi$ one success is enough. In symbols, for (S6) and (S7):

$$\begin{aligned} \mathfrak{M}, \ell \models \forall x \varphi & \quad \text{iff} \quad \text{for every } d \in D, \quad \mathfrak{M}, \ell[x \mapsto d] \models \varphi, \\ \mathfrak{M}, \ell \models \exists x \varphi & \quad \text{iff} \quad \text{there is } d \in D \text{ with } \mathfrak{M}, \ell[x \mapsto d] \models \varphi. \end{aligned}$$

A trace, step by step. Decide $\forall x \exists y (x < y)$ in two structures, both with $<$ the usual order: $\mathfrak{M}_1$ with domain $\mathbb{N}$, and $\mathfrak{M}_2$ with domain $\{1, 2, 3\}$. Let ℓ be any assignment.

In $\mathfrak{M}_1$: by (S6) we must check, for every $d \in \mathbb{N}$, that $\mathfrak{M}_1, \ell[x \mapsto d] \models \exists y (x < y)$. Fix d . By (S7) it is enough to find one $e \in \mathbb{N}$ with $\mathfrak{M}_1, \ell[x \mapsto d][y \mapsto e] \models x < y$. Take $e = d + 1$. Under $\ell[x \mapsto d][y \mapsto d + 1]$ the terms x and y have the values d and $d + 1$, and $d < d + 1$, so (S1) holds. Hence the sentence is satisfied. Notice that the witness $e = d + 1$ depends on d : (S7) is checked **inside** (S6), once for each d .

In $\mathfrak{M}_2$: (S6) asks the same for $d = 1, 2, 3$. At $d = 1$ and $d = 2$ the witnesses $e = 2$ and $e = 3$ work. At $d = 3$ no $e \in \{1, 2, 3\}$ has $3 < e$, so (S7) fails for $\ell[x \mapsto 3]$, and (S6) fails. So $\mathfrak{M}_2 \not\models \forall x \exists y (x < y)$, and the failure has a name: the element 3 has nothing above it.

The assignment matters for free variables. In $\mathfrak{M}_1 = (\mathbb{N}, <)$, with $\leq$ also read as usual, take the formula $\forall y (x \leq y)$, in which x is free. Under an assignment with $\ell(x) = 0$ it is satisfied, since every $e \in \mathbb{N}$ has $0 \leq e$ (recall that $\mathbb{N} = \{0, 1, 2, \dots\}$). Under an assignment with $\ell(x) = 1$ it is not: $e = 0$ gives $1 \leq 0$, which fails. The same formula in the same structure is satisfied under one assignment and not under another; it is a property of x , “ x is least”, not a statement.

○ **Second reading** · *Satisfaction is well defined; degenerate cases*

Well defined. The clauses define, for each formula, which pairs $(\mathfrak{M}, \ell)$ satisfy it, by recursion on the formula: (S1)–(S3) settle the atomic formulas and $\perp$, and each of (S4)–(S7) settles a compound formula from its immediate parts. One point needs a word. Clauses (S6) and (S7) consult the immediate part φ under **other** assignments $\ell[x \mapsto d]$, not under ℓ . So the recursion must define satisfaction of each formula under **all** assignments at once, and then (S6) reads off the answer for φ at every $\ell[x \mapsto d]$. The recursion is still on the formula alone, and Theorem 4.3.4, applied to formulas built by the clauses of Definition 5.2.2, gives exactly one such relation.

Degenerate cases. In the one-element structure $D = \{a\}$, every assignment gives every variable the value a , so $\ell[x \mapsto a] = \ell$ and (S6) and (S7) both reduce to “ $\mathfrak{M}, \ell \models \varphi$ ”. So $\forall x \varphi$ and $\exists x \varphi$ are satisfied together or not at all, and every equation $x = y$ is satisfied. In any structure,

$\forall x (x = x)$ is satisfied, since every d is the same element as itself; and $\exists x \neg(x = x)$ never is.

Satisfaction as a game. Section 5.1 read a sentence whose quantifiers all stand in front as a game between a challenger, who tries to make it false, and a defender, who tries to make it true. The clauses extend that game to every formula, one move per clause. At $\forall x \varphi$ (S6) the challenger picks an element $d \in D$ for x ; at $\exists x \varphi$ (S7) the defender picks. At $\varphi \wedge \psi$ the challenger picks the side to be checked, and at $\varphi \vee \psi$ the defender does; $\varphi \rightarrow \psi$ is played as $\neg\varphi \vee \psi$, and $\varphi \leftrightarrow \psi$ as the two implications joined by $\wedge$. At $\neg\varphi$ the players change roles (S4), since making $\neg\varphi$ true is making φ false. At an atomic formula the game ends, and the defender wins exactly when (S1) or (S2) holds; at $\perp$ the defender never wins (S3). A sentence is true in $\mathfrak{M}$ exactly when the defender can answer every challenge and win: read clause by clause, that is what (S1)–(S7) say. In the trace above, the defender’s answer $e = d + 1$ to each challenge d is such a way of winning on $\mathbb{N}$; on $\{1, 2, 3\}$ the challenge $d = 3$ has no answer.

Quick check. Play the game for $\exists y \forall x (x \leq y)$ on $(\mathbb{N}, \leq)$ and on $(\{1, 2, 3\}, \leq)$. Who moves first, and who wins?

Solution. The defender moves first, since $\exists$ comes first: the order of the quantifiers is the order of the moves. On $\mathbb{N}$, whatever e the defender names, the challenger answers $x = e + 1$ and wins, so the sentence is false. On $\{1, 2, 3\}$ the defender names 3, and every challenge x has $x \leq 3$, so the sentence is true.

Quick check. Is $\forall x \exists y (y \cdot y = x)$ satisfied in $\mathbb{R}$, in $[0, \infty)$, and in $\mathbb{C}$, each with $\cdot$ the usual product?

Solution. In $\mathbb{R}$, no: at $d = -1$, (S7) asks for a real e with $e \cdot e = -1$, and every real square is ≥ 0 . In $[0, \infty)$, yes: for each $d \geq 0$, $e = \sqrt{d}$ is in the domain and $e \cdot e = d$; and $[0, \infty)$ with $\cdot$ is a structure, since a product of non-negative reals is non-negative (M3). In $\mathbb{C}$, yes: every complex number has a complex square root (school algebra of complex numbers: $r(\cos \theta + i \sin \theta)$ with $r \geq 0$ has the square root $\sqrt{r}(\cos \frac{\theta}{2} + i \sin \frac{\theta}{2})$). The sentence is the same all three times; only the structure changes.

7.1.6 Only the free variables matter

The example of $\forall y (x \leq y)$ above showed ℓ mattering. In the first example it did not: the verdict on $\forall x \exists y (x < y)$ was reached for “any assignment ℓ ”. That is no accident. A sentence has no free variables (Definition 5.2.4), and every variable it contains is bound, so every value ℓ gives is overwritten by (S6) or (S7) before it is looked at. We prove the general fact, which is the first-order version of Chapter 4’s Lemma 4.4.1: satisfaction depends only on the values of the **free** variables.

Say that two assignments ℓ and ℓ' **agree on** a set of variables if $\ell(y) = \ell'(y)$ for every variable y in it. We need the free variables of a formula (Definition 5.2.3), in the recursive description

given after that definition in Section 5.2: every variable occurring in an atomic formula is free there; $\neg\varphi$ has the free variables of φ ; $\varphi \circ \psi$, for a binary connective $\circ$, has those of φ and those of ψ ; and $\forall x \varphi$ and $\exists x \varphi$ have those of φ other than x .

Lemma 7.1.1. Free Variables Coincidence Lemma

Let $\mathfrak{M}$ be a structure and φ a formula, and let ℓ and ℓ' be assignments in $\mathfrak{M}$ that agree on every free variable of φ . Then

$$\mathfrak{M}, \ell \models \varphi \quad \text{if and only if} \quad \mathfrak{M}, \ell' \models \varphi.$$

In words: to decide a formula, only the values of its free variables matter.

○ **Second reading** · *The proof: an induction for all assignments at once*

Idea. Structural induction on φ , as for Lemma 4.4.1, after a first induction on terms: a term's value depends only on the values of its variables. The connective cases are Chapter 4's. The quantifier case is the only new one, and it explains why the property must be stated **for all** pairs of assignments: (S6) compares $\ell[x \mapsto d]$ with $\ell'[x \mapsto d]$, which are new assignments, so the induction hypothesis must apply to them. They agree at x , since both give it d , and at every other free variable of φ , since those are free in $\forall x \varphi$.

Proof. Terms first. We show: if ℓ and ℓ' agree on every variable occurring in a term t , then $t^{\mathfrak{M}, \ell} = t^{\mathfrak{M}, \ell'}$. We use structural induction on terms (Definition 5.2.1; the pattern of Theorem 4.3.1). For a variable x , the variable x occurs in x , so $x^{\mathfrak{M}, \ell} = \ell(x) = \ell'(x) = x^{\mathfrak{M}, \ell'}$. For a constant c , both values are $c^{\mathfrak{M}}$. For $f(t_1, \dots, t_n)$, every variable of t_i occurs in $f(t_1, \dots, t_n)$, so the hypothesis for t_i gives $t_i^{\mathfrak{M}, \ell} = t_i^{\mathfrak{M}, \ell'}$ for each i ; applying $f^{\mathfrak{M}}$ to equal tuples gives equal values.

Formulas. We apply structural induction to the property $C(\varphi)$: *for all assignments ℓ, ℓ' in $\mathfrak{M}$ that agree on the free variables of φ , $\mathfrak{M}, \ell \models \varphi$ if and only if $\mathfrak{M}, \ell' \models \varphi$.*

Atomic formulas. Let φ be $R(t_1, \dots, t_n)$, and let ℓ, ℓ' agree on its free variables. Every variable occurring in φ is free in it (Definition 5.2.3), so ℓ and ℓ' agree on the variables of each t_i , and by the claim for terms $t_i^{\mathfrak{M}, \ell} = t_i^{\mathfrak{M}, \ell'}$. The two tuples in (S1) are equal, so one lies in $R^{\mathfrak{M}}$ exactly when the other does. The case $t_1 = t_2$ is the same with (S2). For $\perp$, neither side holds, by (S3).

Connectives. Assume $C(\varphi)$ and $C(\psi)$, and let ℓ, ℓ' agree on the free variables of $\varphi \wedge \psi$. These include the free variables of φ and those of ψ , so $\mathfrak{M}, \ell \models \varphi$ if and only if $\mathfrak{M}, \ell' \models \varphi$, and likewise for ψ . By (S5), satisfaction of $\varphi \wedge \psi$ is decided by these two answers through the same rule for ℓ and for ℓ' , so $C(\varphi \wedge \psi)$ holds. The cases $\neg\varphi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, and $\varphi \leftrightarrow \psi$ are the same argument with (S4) and the other parts of (S5), as in the proof of Lemma 4.4.1.

Universal quantifier. Assume $C(\varphi)$, let x be a variable, and let ℓ, ℓ' agree on the free variables of $\forall x \varphi$. Let $d \in D$. We claim that $\ell[x \mapsto d]$ and $\ell'[x \mapsto d]$ agree on the free variables of φ . At x , both give d . A free variable y of φ other than x is a free variable of $\forall x \varphi$ (Definition 5.2.3),

so $\ell[x \mapsto d](y) = \ell(y) = \ell'(y) = \ell'[x \mapsto d](y)$. By $C(\varphi)$, applied to these two assignments,

$$\mathfrak{M}, \ell[x \mapsto d] \models \varphi \quad \text{iff} \quad \mathfrak{M}, \ell'[x \mapsto d] \models \varphi.$$

This holds for every $d \in D$, so by (S6) $\mathfrak{M}, \ell \models \forall x \varphi$ if and only if $\mathfrak{M}, \ell' \models \forall x \varphi$. This proves $C(\forall x \varphi)$.

Existential quantifier. The same argument, with (S7) in place of (S6): the displayed equivalence holds for each d , so there is a d making the left side hold exactly when there is one making the right side hold.

Every clause of Definition 5.2.2 has been covered, so $C(\varphi)$ holds for every formula φ . This proves the lemma. ■

Quick check. In $(\mathbb{N}, \leq)$, does whether ℓ satisfies $\forall y (x \leq y)$ depend on $\ell(y)$?

Solution. No: y is not free in the formula, so by the lemma only $\ell(x)$ matters. The formula holds exactly when $\ell(x) = 0$.

The lemma has a consequence that the rest of the section uses constantly.

Sentences. If φ is a sentence, it has no free variables, so **any** two assignments agree on its free variables, and the lemma says that $\mathfrak{M}, \ell \models \varphi$ either holds for every assignment ℓ or for none. We then write $\mathfrak{M} \models \varphi$, and say that φ is **true in $\mathfrak{M}$** (or that $\mathfrak{M}$ is a model of φ), in the first case, and $\mathfrak{M} \not\models \varphi$, φ is **false in $\mathfrak{M}$** , in the second. At least one assignment exists because D is non-empty, so a sentence is always exactly one of true and false in $\mathfrak{M}$. This is the precise form of Section 5.1's informal “true over a domain”.

Only finitely many values matter. A formula has finitely many free variables. To decide $\mathfrak{M}, \ell \models \varphi$, it is enough to know ℓ on those. So we describe an assignment by the values that matter (“ $\ell(x) = 2$ ”), exactly as Chapter 4 described a valuation by the values of a few atoms.

Warning. The same sentence can be true in one structure and false in another, and “true” without a structure means nothing. $\forall x \exists y (x < y)$ is true in $(\mathbb{N}, <)$ and false in $(\{1, 2, 3\}, <)$, as traced above. A sentence of predicate logic is like a formula of Chapter 4 with atoms: it has a truth value only once the meaning of its symbols has been fixed.

7.1.7 Semantic entailment

Chapter 4 said that Γ entails ψ when every row that makes Γ true makes ψ true (Definition 4.5.3). The row is now a structure with an assignment.

An argument is valid when every structure, under every assignment, that satisfies all the premises also satisfies the conclusion.

Definition 7.1.4. Semantic Entailment in Predicate Logic

Let Γ be a list of formulas and φ a formula. We say Γ **semantically entails** φ , and write $\Gamma \models \varphi$, if for **every** structure $\mathfrak{M}$ for the symbols occurring in Γ and φ , and **every** assignment ℓ in $\mathfrak{M}$,

$$\mathfrak{M}, \ell \models \Gamma \quad \Rightarrow \quad \mathfrak{M}, \ell \models \varphi.$$

We write $\models \varphi$ when Γ is empty. If $\Gamma \models \varphi$ fails, we write $\Gamma \not\models \varphi$; a structure $\mathfrak{M}$ with an assignment ℓ such that $\mathfrak{M}, \ell \models \Gamma$ and $\mathfrak{M}, \ell \not\models \varphi$ is a **counter-model** to $\Gamma \models \varphi$.

In words: “every structure” means every domain and every meaning of the symbols, so an entailment holds for reasons of form and not because of what R happens to mean. “Every assignment” covers every value of the free variables; when Γ and φ are sentences, the assignment makes no difference, by [Lemma 7.1.1](#). And a counter-model is the analogue of Chapter 4’s refuting row: one structure, with one assignment, where the premises hold and the conclusion does not. The symbol $\models$ is the one of Chapter 4, and the context always shows which is meant: a sequent of formulas with quantifiers or predicate symbols is read with this definition.

A structure may interpret symbols that do not occur in Γ or φ . The clauses of [Definition 7.1.3](#) consult only the symbols of the formula at hand, so such extra meanings change nothing.

First examples of entailment. The argument Section 5.1 opened with, “every positive real has a square root; 2 is positive; so 2 has a square root”, is $\forall x (P(x) \rightarrow Q(x)), P(c) \vdash Q(c)$, derived in Section 5.3. Truth tables called it invalid, since they saw three unrelated atoms. It is valid in the new sense:

$$\forall x (P(x) \rightarrow Q(x)), P(c) \models Q(c).$$

Let $\mathfrak{M}, \ell$ satisfy both premises, and let $a = c^{\mathfrak{M}}$. By (S6) for the first premise, with $d = a$, we get $\mathfrak{M}, \ell[x \mapsto a] \models P(x) \rightarrow Q(x)$. Under $\ell[x \mapsto a]$ the term x has value a , and under ℓ the term c has value a ; so by (S1), $\mathfrak{M}, \ell[x \mapsto a] \models P(x)$ holds if and only if $a \in P^{\mathfrak{M}}$ if and only if $\mathfrak{M}, \ell \models P(c)$, which is the second premise. So by (S5) $\mathfrak{M}, \ell[x \mapsto a] \models Q(x)$, that is, $a \in Q^{\mathfrak{M}}$, and by (S1) again $\mathfrak{M}, \ell \models Q(c)$. Chapter 4’s objection to this argument is gone.

7.1.8 Proving an entailment by meaning

To show $\Gamma \models \varphi$, fix an arbitrary structure and assignment satisfying Γ , and show that they satisfy φ . The first proof below writes every step with the clause that licenses it; the second condenses it to what a mathematician writes.

Example. A Semantic Proof, Clause by Clause

Show that $\exists x \forall y R(x, y) \models \forall y \exists x R(x, y)$.

Solution. Each step ends with the clause that licenses it.

1. Let $\mathfrak{M}$ be a structure with domain D and ℓ an assignment with $\mathfrak{M}, \ell \models \exists x \forall y R(x, y)$ ([Definition 7.1.4](#)).
2. There is $d \in D$ with $\mathfrak{M}, \ell[x \mapsto d] \models \forall y R(x, y)$ (S7).
3. To show $\mathfrak{M}, \ell \models \forall y \exists x R(x, y)$, let $e \in D$; we must show $\mathfrak{M}, \ell[y \mapsto e] \models \exists x R(x, y)$ (S6).
4. From item 2, with this e : $\mathfrak{M}, \ell[x \mapsto d][y \mapsto e] \models R(x, y)$ (S6).
5. Under $\ell[x \mapsto d][y \mapsto e]$ the terms x, y have the values d, e , so $(d, e) \in R^{\mathfrak{M}}$ (S1).
6. Take the witness d for x : under $\ell[y \mapsto e][x \mapsto d]$ the values are again d, e , so $\mathfrak{M}, \ell[y \mapsto e] \models \exists x R(x, y)$.

$e][x \mapsto d] \models R(x, y)$ by item 5 (S1), and so $\mathfrak{M}, \ell[y \mapsto e] \models \exists x R(x, y)$ (S7).

7. Since e was arbitrary, $\mathfrak{M}, \ell \models \forall y \exists x R(x, y)$ (S6).

Condensed. Suppose some d is R -related to everything. Then for each e , the element d itself is related to e , so each e has something related to it.

Section 5.6 derived the same sequent (Proposition 5.6.2); this is its meaning. The converse fails, and Section 7.2 draws why.

Quick check. In item 6, why are $\ell[y \mapsto e][x \mapsto d]$ and $\ell[x \mapsto d][y \mapsto e]$ the same assignment?

Solution. The variables x and y are different, and each change touches only its own variable, so the order of the two changes does not matter (the paragraph after Definition 7.1.2).

Example. The Drinker Sentence, by Meaning

Show that $\models \exists x (P(x) \rightarrow \forall y P(y))$: in every structure, some element has the property that if it has P , everything has P .

Solution. Let $\mathfrak{M}$ and ℓ be given, with domain D .

Case 1: $P^{\mathfrak{M}} = D$. Choose any $d \in D$, which exists because D is non-empty (M1). Under $\ell[x \mapsto d]$ the formula $\forall y P(y)$ is satisfied, by (S6), so the implication is satisfied, by (S5); hence the sentence holds, by (S7).

Case 2: some $d \in D$ is not in $P^{\mathfrak{M}}$. Under $\ell[x \mapsto d]$ the hypothesis $P(x)$ fails (S1), so the implication is satisfied (S5), and (S7) gives the sentence.

The case split is on a question about the whole structure, whether $P^{\mathfrak{M}} = D$, and the witness depends on its answer; this is why no witness can be named in advance, and why a derivation needs a detour (Section 6.3 derived it, with D, n, m in place of P, x, y , Exercise (C1: Some index decides, used later)). Case 1 is where the non-empty domain is used, as in the next subsection.

7.1.9 Why domains are non-empty

Clause (M1) of Definition 7.1.1 demands a non-empty domain. Here is the reason. Proposition 5.5.1 derives $\forall x P(x) \vdash \exists x P(x)$ in three lines: $\forall E$ at the variable x , then $\exists I$. The matching entailment holds for our structures:

$$\forall x P(x) \models \exists x P(x).$$

Let $\mathfrak{M}, \ell \models \forall x P(x)$. Since D is non-empty, choose some $d \in D$. By (S6), $\mathfrak{M}, \ell[x \mapsto d] \models P(x)$, and this d is the one element (S7) asks for, so $\mathfrak{M}, \ell \models \exists x P(x)$.

Now suppose empty domains were allowed. In a structure with $D = \emptyset$, (S6) says $\forall x P(x)$ is satisfied, since “for every $d \in \emptyset$ ” has nothing to check; and (S7) says $\exists x P(x)$ is not, since there

is no d at all. So the sequent would be derivable and yet fail in a structure, and the rules would lead from a true premise to a false conclusion. (Worse, no assignment into an empty domain exists, so even the clauses could not be applied as written.) One of two things had to give: the rule $\forall E$ at a variable, or empty domains. The book, like most of mathematics, keeps the rules and makes domains non-empty.

Nothing is lost. Statements about an empty **set** are handled by the bounded quantifiers of Definition 5.1.1, inside a non-empty domain. The bounded forms $\forall x \in \emptyset \varphi$ and $\exists x \in \emptyset \varphi$ abbreviate $\forall x (x \in \emptyset \rightarrow \varphi)$ and $\exists x (x \in \emptyset \wedge \varphi)$, and in a structure with domain $\mathbb{R}$ in which the one-place predicate symbol $\in \emptyset$ (Section 5.2) is interpreted, by (M4), as the empty subset of $\mathbb{R}$, the first is true (every hypothesis $d \in \emptyset$ is false) and the second is false (no d is in $\emptyset$). Vacuous truth lives in the bound, not in the domain.

7.1.10 Exercises

7.1.10.1 A. Check your understanding

Exercise. A1

Let $\mathfrak{M}_{\mathbb{N}}$, $\mathfrak{M}_{\mathbb{Z}}$, $\mathfrak{M}_{\mathbb{Q}}$ be the structures with domains $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, each with $<$ the usual order.

- For each of the sentences (i) $\forall x \exists y (x < y)$, (ii) $\exists x \forall y \neg (y < x)$, (iii) $\forall x \forall y (x < y \rightarrow \exists z (x < z \wedge z < y))$ and (iv) $\forall x \exists y (y < x)$, say in which of the three structures it is true, with a witness or a failing element for each verdict.
- True or false: if a sentence is true in some structure, it is true in every structure for its symbols.
- Why must the domain of a structure be non-empty? Name the derivable sequent that would otherwise fail.
- In [Example \(A Semantic Proof, Clause by Clause\)](#), which clause produces the element d , and why may the same d serve for every e ?

Solution. (a) (i) True in all three: for each d , take $y = d + 1$, which lies in the same domain and is larger.

- This says “there is a least element”. True in $\mathfrak{M}_{\mathbb{N}}$, with $x = 0$: no natural number is below 0. False in $\mathfrak{M}_{\mathbb{Z}}$ and $\mathfrak{M}_{\mathbb{Q}}$: for any d , the element $e = d - 1$ has $e < d$, so (S6) fails for $\neg(y < x)$ under $\ell[x \mapsto d]$.
- This says “between any two elements there is a third” (density). False in $\mathfrak{M}_{\mathbb{N}}$ and $\mathfrak{M}_{\mathbb{Z}}$: at $x = 1$, $y = 2$ there is no integer strictly between. True in $\mathfrak{M}_{\mathbb{Q}}$: for $d < e$ in $\mathbb{Q}$, $z = (d + e)/2$ is rational and $d < z < e$.
- This says “no element is least”. False in $\mathfrak{M}_{\mathbb{N}}$, at $x = 0$. True in $\mathfrak{M}_{\mathbb{Z}}$ and $\mathfrak{M}_{\mathbb{Q}}$, with $y = d - 1$.

	$\mathbb{N}$	$\mathbb{Z}$	$\mathbb{Q}$
(i)	true	true	true
(ii)	true	false	false
(iii)	false	false	true
(iv)	false	true	true

(b) False. Sentence (ii) is true in $\mathfrak{M}_{\mathbb{N}}$ and false in $\mathfrak{M}_{\mathbb{Z}}$, as the table shows.

(c) The sequent is $\forall x P(x) \vdash \exists x P(x)$, derived in Proposition 5.5.1 by $\forall E$ at the variable x and $\exists I$. In an empty domain $\forall x P(x)$ would be true, with nothing to check, and $\exists x P(x)$ false, with no witness; so the rules would lead from a true premise to a false conclusion. (There would also be no assignments at all.)

(d) (S7), applied to the premise in item 2. The element d is fixed before any e is chosen, and item 2 says that d is related to **every** e , by (S6); so item 6 may use d whatever e item 3 picked.

Exercise. A2: Drill

Each part takes a line or two. Parts (a)–(d) read a structure or a clause, (e)–(i) decide a formula in a given structure, with a witness or a failing element, (j) and (k) judge, and (l) builds a structure.

- (a) Let $D = \{a, b\}$ and $P^{\mathfrak{M}} = \{a\}$. Is $\forall x P(x)$ true in $\mathfrak{M}$?
- (b) In the same structure, is $\exists x \neg P(x)$ true?
- (c) In $(\mathbb{R}; 1, +, \cdot)$, what is the value of $x \cdot x + 1$ under an assignment with $\ell(x) = -2$?
- (d) By Lemma 7.1.1, the values of which variables **can** matter to the truth of $\forall y (x \leq y) \wedge z = z$ in a given structure? Do they all matter?
- (e) In $(\{0, 1, 2\}, \leq)$, is $\exists x \forall y (x \leq y)$ true?
- (f) In $(\mathbb{N}, <)$, under an assignment with $\ell(x) = 3$, is $\exists y (y < x)$ satisfied? And with $\ell(x) = 0$?
- (g) In the left structure of Figure F, is $\forall x (P(x) \rightarrow \exists y R(y, x))$ true?
- (h) Decide $\forall x \forall y (x \leq y \vee y \leq x)$ in $(\mathbb{N}, \leq)$ and in the right structure of Figure F.
- (i) Decide $\exists x \forall y (y \cdot x = y)$ in $(\mathbb{Z}, \cdot)$ and in the even integers with $\cdot$.
- (j) Is $\mathbb{N}$ with the two-place function symbol $-$ read as subtraction a structure?
- (k) In a structure with $D = \{a\}$ and $P^{\mathfrak{M}} = \emptyset$, decide $\forall x P(x) \rightarrow \exists x P(x)$ and $\forall x \neg P(x) \rightarrow \neg \exists x P(x)$.
- (l) Give a structure on $\{1, 2\}$ in which $\exists x R(x, x)$ is false and $\forall x \exists y R(x, y)$ is true.

Solution.

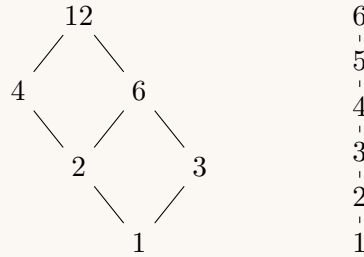
- (a) No: it fails at b .
- (b) Yes, at b .
- (c) $(-2) \cdot (-2) + 1 = 5$.
- (d) At most x and z , its free variables; not y . Not always: z never matters, since $z = z$ holds under every assignment, and whether x matters depends on the structure. In $(\mathbb{N}, \leq)$ it does (true with $\ell(x) = 0$, false with $\ell(x) = 1$); if $\leq^{\mathfrak{M}}$ is empty it does not. The lemma gives an upper bound, not the exact set.
- (e) Yes, with $x = 0$.
- (f) Yes, with $y = 0$. With $\ell(x) = 0$, no: nothing in $\mathbb{N}$ is below 0.
- (g) Yes: the only element with P is a , and the arrow $c \rightarrow a$ enters it; at b and c the hypothesis fails.

- (h) True in $\mathbb{N}$; false in the divisibility structure, at $x = 2, y = 3$.
- (i) True in $\mathbb{Z}$, with $x = 1$. False in the even integers: $y = 2$ would need $2x = 2$, so $x = 1$, which is odd.
- (j) No: (M3) fails, since $0 - 1 \notin \mathbb{N}$.
- (k) Both true: the first because its hypothesis fails at a ; the second because both sides are true.
- (l) $R^{\mathfrak{M}} = \{(1, 2), (2, 1)\}$: no loops, and each element has an arrow out.

7.1.10.2 B. Practice

Exercise. B1: One sentence, two pictures

Two structures for the one symbol $\leq$ are drawn below. On the left, $\mathfrak{M}_1$: the domain $D_1 = \{1, 2, 3, 4, 6, 12\}$, with $x \leq y$ read as “ x divides y ”, as a Hasse diagram. On the right, $\mathfrak{M}_2 = (\{1, 2, 3, 4, 5, 6\}, \leq)$, with the usual order, as a vertical chain.



Decide each sentence in each structure, with a witness or a failing pair:

- (i) $\exists x \forall y (x \leq y)$;
- (ii) $\exists x \forall y (y \leq x)$;
- (iii) $\forall x \forall y (x \leq y \vee y \leq x)$;
- (iv) $\forall x \forall y \exists z (x \leq z \wedge y \leq z)$.

Which of them tell the two structures apart?

Solution. (i) True in both: $x = 1$ in each, since 1 divides every element of D_1 and is the least element of $\mathfrak{M}_2$.

(ii) True in both: $x = 12$ in $\mathfrak{M}_1$, since every element of D_1 divides 12; $x = 6$ in $\mathfrak{M}_2$.

(iii) False in $\mathfrak{M}_1$, at $x = 2, y = 3$: neither divides the other. True in $\mathfrak{M}_2$: of two numbers, one is at most the other.

(iv) True in both: $z = 12$ in $\mathfrak{M}_1$, and $z = 6$ in $\mathfrak{M}_2$ (or z the larger of x and y).

Only (iii) separates them.

Marking scheme. (10 points) - 8: 2 for each of (i)–(iv) in both structures, 1 for the verdicts and 1 for a named witness or failing pair for every verdict. - 2: naming (iii) as the only separating sentence. Fatal errors (0 for the part): a verdict on $\mathfrak{M}_1$ that reads $\leq$ as the usual order of numbers.

Exercise. B2: A model on three elements

Give an example of a structure with exactly three elements in which both

$$\forall x \exists y R(x, y) \quad \text{and} \quad \forall x \neg R(x, x)$$

are true. Check both sentences. Which properties must every such structure have?

Solution. Take $D = \{0, 1, 2\}$ and $R^{\mathfrak{M}} = \{(0, 1), (1, 2), (2, 0)\}$: each element is related to the next one around a circle. First sentence: for $d = 0, 1, 2$, the witnesses $e = 1, 2, 0$ give pairs in $R^{\mathfrak{M}}$. Second sentence: none of $(0, 0), (1, 1), (2, 2)$ is in $R^{\mathfrak{M}}$, so $\neg R(x, x)$ is satisfied under $\ell[x \mapsto d]$ for each d .

The properties every answer must have: the domain has three elements; every element d has **some** e with $(d, e) \in R^{\mathfrak{M}}$; and **no** pair (d, d) is in $R^{\mathfrak{M}}$, so that e is always different from d . Any $R^{\mathfrak{M}}$ meeting these works, for instance $\{(0, 1), (1, 0), (2, 0)\}$ or the set of all pairs (d, e) with $d \neq e$.

Exercise. B3: Is it a structure?

For each description, say whether it is a structure for the symbols named. If not, name the clause of Definition 7.1.1 that fails, and repair it with as small a change as you can.

- (i) $D = \mathbb{Q}$, with a one-place function symbol f read as $x \mapsto 1/x$ for $x \neq 0$ and $f(0) = 0$.
- (ii) $D = \mathbb{N}$, with f read as $x \mapsto x - 1$.
- (iii) $D = \emptyset$, with P read as the empty set.
- (iv) $D = \mathbb{Z}$, with a constant c naming $\frac{1}{2}$.
- (v) $D = \mathbb{R}$, with a two-place predicate symbol R read as $\{(x, y) : y = \sqrt{x}\}$.
- (vi) $D = \mathbb{R}$, with a one-place function symbol g read as $x \mapsto \sqrt{x}$.

Solution. (i) Yes: every element has a value in $\mathbb{Q}$.

(ii) No: (M3) fails, since $f(0) = -1 \notin \mathbb{N}$. Repair: $f(0) := 0$.

(iii) No: (M1) fails. Repair: any non-empty domain.

(iv) No: (M2) fails, since $\frac{1}{2} \notin \mathbb{Z}$. Repair: $D = \mathbb{Q}$, or let c name an integer.

(v) Yes: (M4) asks only for a set of pairs, and nothing requires a relation to hold for every first element.

(vi) No: (M3) fails, since $\sqrt{-1}$ is not real. Repair: $D = [0, \infty)$, or fix $g(x) := 0$ for $x < 0$.

Descriptions (v) and (vi) carry the same information, and only the function symbol must be total.

Marking scheme. (10 points) - 6: 1 for each verdict. - 2: the clause named at each “no” (all four right). - 2: repairs that are minimal and correct. Fatal errors (0 for the part): calling (v) a non-structure because $\sqrt{x}$ is undefined for $x < 0$.

7.1.10.3 C. Going deeper

Exercise. C1: Proofs by meaning

- (a) Prove $\forall x (P(x) \wedge Q(x)) \models \forall x P(x) \wedge \forall x Q(x)$ clause by clause, as in [Example \(A Semantic Proof, Clause by Clause\)](#).
- (b) Prove $\forall x (P(x) \vee Q(x)) \models \forall x P(x) \vee \exists x Q(x)$, condensed.
- (c) Prove $\exists x (P(x) \rightarrow Q(c)) \models \forall x P(x) \rightarrow Q(c)$, condensed, and say where your proof uses the existential premise.

Solution. (a) Let $\mathfrak{M}, \ell \models \forall x (P(x) \wedge Q(x))$. By (S5) it suffices to show each conjunct. For $\forall x P(x)$, let $d \in D$ (S6). By (S6) for the premise, $\mathfrak{M}, \ell[x \mapsto d] \models P(x) \wedge Q(x)$; by (S5), $\mathfrak{M}, \ell[x \mapsto d] \models P(x)$. Since d was arbitrary, $\mathfrak{M}, \ell \models \forall x P(x)$ (S6). Likewise $\mathfrak{M}, \ell \models \forall x Q(x)$, with the other half of (S5).

(b) If $P^{\mathfrak{M}} = D$, the left disjunct holds. Otherwise some $d \in D$ is not in $P^{\mathfrak{M}}$. The premise at d gives $d \in P^{\mathfrak{M}}$ or $d \in Q^{\mathfrak{M}}$, so $d \in Q^{\mathfrak{M}}$, and $\exists x Q(x)$ holds. Either way the disjunction holds.

(c) Let d be the premise's witness: if $d \in P^{\mathfrak{M}}$, then $c^{\mathfrak{M}} \in Q^{\mathfrak{M}}$. Suppose $\forall x P(x)$ holds; then $d \in P^{\mathfrak{M}}$, so $Q(c)$ holds. The witness d is where the existential premise is used: the hypothesis $\forall x P(x)$ is applied at it.

Marking scheme. (10 points) - 4: (a), with every step's clause named, and (S6) used both to fix d and on the premise. - 3: (b), split on whether $P^{\mathfrak{M}} = D$ (or on a failing element), with the premise used at that element. - 3: (c), taking the premise's witness d and applying the hypothesis $\forall x P(x)$ at d (in either order). Fatal errors (0 for the part): in (a), a proof in which d is a particular element (for example a constant's value, or an element chosen to lie in $P^{\mathfrak{M}}$) rather than an arbitrary $d \in D$, so that the closing (S6) step for $\forall x P(x)$ is unjustified.

Exercise. C2: Challenge: only the symbols that occur matter

Let $\mathfrak{M}$ and $\mathfrak{N}$ be structures with the same domain D that give the same meaning to every symbol occurring in a formula φ . Prove that for every assignment ℓ , $\mathfrak{M}, \ell \models \varphi$ if and only if $\mathfrak{N}, \ell \models \varphi$. (This is the claim after [Definition 7.1.4](#) that extra meanings change nothing.)

Solution (Hint 1). Terms first, as in the proof of [Lemma 7.1.1](#).

Solution (Hint 2). Prove, by structural induction on φ , the property: for **every** assignment $\ell, \dots$; the quantifier case needs it at $\ell[x \mapsto d]$.

Solution. Terms. For every term t whose symbols $\mathfrak{M}$ and $\mathfrak{N}$ interpret alike, $t^{\mathfrak{M}, \ell} = t^{\mathfrak{N}, \ell}$ for every ℓ , by structural induction on t . For a variable x , both values are $\ell(x)$. For a constant c , they are $c^{\mathfrak{M}} = c^{\mathfrak{N}}$. For $f(t_1, \dots, t_n)$, the hypothesis gives equal values of the t_i , and $f^{\mathfrak{M}} = f^{\mathfrak{N}}$ applied to equal tuples gives equal values.

Formulas. Let $A(\varphi)$ be the property: *for every assignment ℓ , $\mathfrak{M}, \ell \models \varphi$ if and only if $\mathfrak{N}, \ell \models \varphi$* , for formulas φ whose symbols $\mathfrak{M}$ and $\mathfrak{N}$ interpret alike. Atomic formulas: the tuples of values are equal by the claim for terms, and $R^{\mathfrak{M}} = R^{\mathfrak{N}}$, so (S1) gives the same answer; equations likewise by (S2); $\perp$ by (S3). Connectives: (S4) and (S5) combine equal answers into equal answers. For $\forall x \varphi$: the two structures have the same domain, so d ranges over the same elements in both; for each d , $A(\varphi)$ at the assignment $\ell[x \mapsto d]$ gives equal answers, and (S6) gives $A(\forall x \varphi)$. For $\exists x \varphi$, the same with (S7). Every clause of Definition 5.2.2 has been covered, so $A(\varphi)$ holds for every formula.

Marking scheme. (10 points) - 2: the claim for terms, by induction. - 2: the atomic cases, with (S1) and (S2). - 2: the connective cases. - 3: the quantifier case, using the property at the changed assignments and the shared domain. - 1: naming where “same domain” is used. Fatal errors (0 for the step): an induction hypothesis stated for one fixed ℓ only.