1 Interpretations

Remark 22.1.1 It is more complicated to give meaning to the formulae of a first-order language. We must state what we are talking about, what domain is involved for the quantifiers to quantify over. We must also specify what the constant symbols denote and what properties and relations the predicate symbols refer to with respect to this domain. The domain for interpreting the quantifiers and the interpretation of the nonlogical symbols together specify a *model* (also called a first-order *structure*.

The notion of a model is relative to a first-order language, since it must interpret the nonlogical symbols of that language. Furthermore, each language will have many different models, besides any single intended interpretation. For example, a language with a single binary predicate $P(x, y)$ could be viewed as talking about

1. The natural numbers $\mathbb{N}$ with $<$,
2. The rational numbers $\mathbb{Q}$ with $>$,
3. The integers $\mathbb{Z}$ with x is *divisible by* y.

We introduce the notion of a model for a first-order language informally by providing guidelines for *translating* a first-order language into a natural language, such as English. The purpose of a translation between a first-order language and and a (tiny fragment) of a natural language is to provide a meaning for the abstract first-order language, while exposing the *logical structure* of sentences in the natural language.

Remark 22.1.2 (Properties and Relations) The simplest predicates are *one-place predicates* (a part of a language) and refer to properties (part of the world at-large) of individuals. For example, in the world of animals "x is a dog" and "x is a unicorn" are properties; in the incorporeal world, "x is a ghost" is a property. A property $\mathcal{P}(x)$ is said to *hold* for an individual i if $\mathcal{P}(i)$ is true. For example, if Phaedo is my dog, then "Phaedo is a dog" is true, but "Phaedo is a unicorn" is false.

More generally, an n-place predicate refers to an n-ary relation between individuals. For example, "x sired y" is a relation among animals, and "x haunts y" is a relation between ghosts and humans. There are also ternary relations such as "x introduced y to z" among people. More complex relations are rarely natural parts of natural languages.

All properties and relations, which are to be used in interpreting first-order languages, must be *unambiguous*. A property $\mathcal{P}(x)$ is unambiguous if it either holds or fails to hold for every individual i, that is $\mathcal{P}(i)$ is either true or it is false (although, we might not be able to decide which). For example, the property "x is short" is ambiguous, as it stands, if we take it as a property of humans. However, if we clarify it by adding "x is under 5 feet 2 inches", then the property would be unambiguous.

We do allow that properties can be *empty*. For example, if there are no ghosts, then the property "x is a ghost" applies to nothing. This also means that relations can be empty as well, such as "x haunts y".

Remark 22.1.3 (Universe of Discourse) An interpretation will pick-out a *universe of discourse* UD (also called a *domain of discourse*), which provides an interpretation of the quantifiers. For example, if my universe of discourse UD are inhabitants of the island of knights and knaves, then the statement "Everyone is a knight or knave." is true, which would not be the case in a wider universe of discourse which includes all humans.

There is one condition on the universe of discourse, it must have *at least one* member. There are logical systems which allow *empty* domains, these are systems of *free logic*. For example, if there are no ghosts, then we cannot take our universe of discourse to be the collection of ghosts.

It is very natural to have distinct domains in mind in interpreting a language. For example, a single universe of discourse may consist of disparate items such as people, plants, animals and incorporeal beings. However, the quantifiers must be interpreted as ranging over the entire universe; first-order languages do not specialize their quantifiers to only a part of a domain, as natural languages do. For example, we can makes statements in English such as "Every human..." or "Every dog..." or "Every ghoul...". A first-order language has a single type of quantifier, $\forall$ and $\exists$, which must range over the entire universe. There is a generalization of first-order logic to logics which allow for different types of individuals, called *multi-sorted logic*.

Definition 22.1.4 An *interpretation* of a first-order language $\mathcal{L}(\mathbf{CS}, \mathbf{PS})$ into a natural language consists of the following:

- A *universe of discourse* UD which is a *nonempty* set of individuals (or objects);
- Each n-place predicate is assigned an unambiguous n-place relation on the individuals of UD. It is also possible for an n-place predicate to be empty;
- Each constant symbol must pick out *exactly* one member of the UD. A member of the UD may be picked out by more than one constant or even none at all.

Example 22.1.5 Consider a first order language with the following predicates:

- One-place predicates $P(x), D(x), G(x)$,
- Two-place predicates $H(x, y)$, $O(x, y)$, $F(x, y)$

An interpretation for this language is given as follows:

- UD: People, dogs and ghosts of Amityville.
- $P(x)$: x is a person.
- $D(x)$: x is a dog.
- $G(x)$: x is a ghost.
- $H(x, y)$: x haunts y.
- $O(x, y)$: x owns y.
- $F(x, y)$: x is a friend of y.

Here are some translations from English to the first-order language.

1. There are dog owners:
$$(\exists x)(P(x) \wedge (\exists y)(D(y) \wedge O(x, y)))$$
2. No one is haunted by ghosts:
$$\neg (\exists x)(P(x) \wedge (\exists y)(G(y) \wedge H(y, x)))$$
3. Every person has a friend:
$$(\forall x)(P(x) \rightarrow (\exists z)(P(z) \wedge F(x, z)))$$
4. Every dog owner has a friend:
$$(\forall x)((\exists y)(D(y) \wedge O(x, y)) \rightarrow (\exists z)(P(z) \wedge F(x, z)))$$

5. Every dog owner has a haunted friend:

$$(\forall x)((\exists y)(D(y) \wedge O(x,y)) \rightarrow (\exists z)(P(x) \wedge F(x,y) \wedge (\exists y)(G(y) \wedge H(y,z))))$$

The first example (1) brings up an important feature of first-order languages. The expression $(\exists y)(D(y) \wedge O(x,y))$ expresses the *property* "x is a dog owner" (There exists a y such that y is a dog and x owns y.) A formula with a free variable is understood as a property. The property of being a haunted dog owner can be expressed as

$$(\exists y)(D(y) \wedge O(x,y) \wedge (\exists z)(G(z) \wedge H(z,x))).$$

The last three examples show how we can restrict the universe of discourse to narrow the range of individuals we want to talk about. We can make a statement about every *person* we pair the predicate $P(x)$ with the conditional: $(\forall x)(P(x) \rightarrow \ldots)$ (for every individual x, if x is a person, then ...). (4,5) illustrate this combination of the universal quantifier and the conditional a more complex property: dog owners.

If there are no ghosts, then the predicate $G(x)$ refers to an empty property. In this case statement (2) is true and statement (5) is false. What about a statement such as "Every ghost haunts a dog owner.":

$$(\forall x)(G(x) \rightarrow (\exists y)(\exists z)(P(y) \wedge D(z) \wedge O(y,z) \wedge H(x,y)))?$$

This statement is true. The reason is that for any x in the UD, the conditional $G(x) \rightarrow \ldots$ is true, since the antecedent is false.

2 First-Order Models

Remark 22.2.1 The problem with determining meaning and truth by *interpreting* in a natural language like English, is that there is a lack of precision. For example, we can take the universe of discourse to be all people, but this domain is always changing. Furthermore, English has ambiguous predicates, such as "x is short" or "x is smart". These features make it hard to give a rigorous account of meaning and truth. Of course, if our first-order language is intended to be a formal model of a fragment of English, we might take these features of the natural language as a challenge. For example, if our intent was to program a computer to respond to a speaker of English, we would have to face these problems, if our computer were to respond in a reasonable manner to a human speaker.

However, our goal in studying first-order logic is to provide a language that captures *mathematical inference.* For this reason, we will consider interpretations of our first-order languages which are mathematically precise. So, a universe of discourse will consist of a set of mathematical objects. We will also need to identify what an n-place relation on such a domain, which must be appropriate for translating n-place predicates. Finally, we will give a precise definition of truth, which will be adequate to give an account of correct deductive inference as it is used in mathematics.

Definition 22.2.2 Let k be a positive integer. For any set A, the set A^k is the set of k-tuples on U:

$$A^k = \{\langle a_1, \ldots, a_k\rangle : a_1, \ldots, a_k \in A\}.$$

A *k-ary relation on* R on A is a subset of the k-tuples of A, that is $R \subseteq A^k$.

Remark 22.2.3 A *structure* (or *interpretation*) for a first-order language provides the information required for determining the truth value of all *atomic formulae*, so plays a similar role as the assignment of truth values to propositional symbols in propositional logic.

Definition 22.2.4 (structure) A *structure* $\mathcal{A}$ for a language $\mathcal{L}(\mathbf{CS}, \mathbf{PS})$ consists of the following:

1. A nonempty domain A, called the *domain* or *universe* of $\mathcal{A}$,
2. For each constant $c \in \mathbf{CS}$ an element $c^{\mathcal{A}}$ from A;

3. For each k-ary predicate $P \in \mathbf{PS}$ a k-ary relation on A:

$$P^{\mathcal{A}} \subset A^n = \{\langle a_1, \ldots, a_n \rangle : a_1, \ldots, a_n \in A\}.$$

Example 22.2.5 Consider the language with constants $\{c_0, c_1, \ldots\}$ and one binary predicate $R(x, y)$. The following is a structure $\mathcal{A}$ for this language

1. The domain is $\mathbb{N} = \{0, 1, 2, \ldots\}$
2. We interpret $c_n^{\mathcal{A}} = n$.
3. The binary predicate $R(x, y)$ is interpreted as $<$ on the natural numbers:

$$R^{\mathcal{A}} \subset \mathbb{N}^2 = \{\langle n, m \rangle : n < m\}.$$

A second interpretation $\mathcal{B}$ is given by

1. The domain is $B = \mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$
2. We interpret $c_n^{\mathcal{B}} = n$.
3. The binary predicate $R(x, y)$ is interpreted as $<$ on the integers:

$$R^{\mathcal{B}} \subset \mathbb{Z}^2 = \{\langle n, m \rangle : n < m\}.$$

Note that in the second example we do not have a name in the language for each element in the universe, although this can be easily fixed.

A third interpretation $\mathcal{C}$ is given by

1. The domain is $B = \mathbb{R}$, the set of real numbers
2. We interpret $c_n^{\mathcal{C}} = n$.
3. The binary predicate $R(x, y)$ is interpreted as $<$ on the real numbers:

$$R^{\mathcal{B}} \subset \mathbb{R}^2 = \{\langle r, s \rangle : r < s\}.$$

In this third example we have no principled means of assigning a name to each number, because there are too many real numbers.

Remark 22.2.6 A *first-order structure* $\mathcal{A}$ for a language $\mathcal{L}$ looks alot like the interpretations we introduced at the beginning of the lecture when we translated a language into a fragment of English. So far, we have added nothing new. The tricky part is how to provide a notion of truth for a language. The fundamental difficulty is how to define the truth of a quantified formula $(\forall x)\alpha$ or $(\exists x)\alpha$.

Our approach here will be the most straightforward one. We will consider only structures $\mathcal{A}$ for languages $\mathcal{L}$ which have a name denoting each element $a \in A$. We cannot expect a language $\mathcal{L}$ built into it, so we will at the same time consider an *expansion* of the language $\mathcal{L}$ to include names for the members of the intended interpretation $\mathcal{A}$. The simplest way to do this is to allow each element of A to name itself.

Definition 22.2.7 (Universe of Discourse) Let $\mathcal{A}$ be a structure for a first-order language $\mathcal{L}(\mathbf{CS}, \mathbf{PS})$, where we further assume the domain of A is disjoint from the constant symbols $\mathbf{CS}$ of $\mathcal{L}$. The *expansion* of $\mathcal{L}$ with A denoted by $\mathcal{L}^A$, is the language $\mathcal{L}(\mathbf{CS} \cup A, \mathbf{PS})$ obtained by adding constant terms naming each element of A. (We will take each element of A as naming itself in the expanded language $\mathcal{L}^A$. If $a \in A$, we will take $a^{\mathcal{A}} = a$.)

A *formula with constants in* A (or A-formula) is like a formula of $\mathcal{L}$, except that it may have elements of A in place of constant symbols from $\mathcal{L}$. We also extend the notion of a *substitution* of elements of A for free occurrences of variables, in the same way we did with constant symbols in Lecture 20, except that now we allow substitutions of elements of A for variables as well. An A-*sentence* is an A-formula without free variables.

Example 22.2.8 Let $\mathcal{L}$ be a language which has a unary predicate P and a binary predicate R. In our previous example 22.2.5 the structures were familiar infinite structures. Very often it is more convenient to create small simple structures.

Let A be the *finite universe* $\{1, 2, 3\}$. The following provides a complete description of $P^{\mathcal{A}}$ and $R^{\mathcal{A}}$ for the structure $\mathcal{A}$:

True	False
$P(1)$	$P(2)$
$P(3)$	$R(1,1)$
$R(1,2)$	$R(2,2)$
$R(1,3)$	$R(3,3)$
$R(2,3)$	$R(2,1)$
	$R(3,1)$
	$R(3,2)$

Alternatively, we could have determined the structure $\mathcal{A}$ by specifying:

- $P^{\mathcal{A}}$ is the set $\{1, 3\}$,
- $R^{\mathcal{A}}$ is the set of ordered pairs $\{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 3\rangle\}$.

Here is a different structure $\mathcal{B}$ over the same universe $\{1, 2, 3\}$: $R^{\mathcal{B}} = R^{\mathcal{A}}$, but $P^{\mathcal{B}} = \emptyset$. There are many different structures for the same language over the same universe $\{1, 2, 3\}$.

Remark 22.2.9 We can now define when a sentence ϕ of a language $\mathcal{L}$ is true in a given structure for $\mathcal{L}$. We will write this as $\mathcal{A} \models \phi$. The definition is by Structural Recursion on sentences, since it is sentences that are ultimately true of false in a structure. (Recall that a formula with free variables is neither true nor false until we have stated what the variables stand for.) The interesting case is that for the quantifiers. Here we use the fact that each element of the structure $\mathcal{A}$ has a name in the language. (If necessary, we consider expansions $\mathcal{L}^A$ which include names for each element of A.)

Theorem 22.2.10 (Truth valuations) Let $\mathcal{A}$ be a structure for a language $\mathcal{L}$. We will define a truth assignment over the A-sentences of $\mathcal{L}^A$, and use this to define truth over sentences of $\mathcal{L}$.

There is a unique function $v_{\mathcal{A}}$ from the A-sentences of $\mathcal{L}^A$ to $\{\mathbf{T}, \mathbf{F}\}$ which satisfies the following conditions for every sentence ϕ and ψ in the expanded language:

$$\begin{array}{rcl}
v_{\mathcal{A}}(R(t_1,\ldots,t_n)) & = & \begin{cases} \mathbf{T} & \text{if } \langle [t_1]^{\mathcal{A}},\ldots,[t_n]^{\mathcal{A}}\rangle \in R^{\mathcal{A}}, \\ \mathbf{F} & \text{otherwise ;} \end{cases} \\
v_{\mathcal{A}}(\top) & = & \mathbf{T} \\
v_{\mathcal{A}}(\bot) & = & \mathbf{F} \\
v_{\mathcal{A}}(\neg\,\phi) & = & \begin{cases} \mathbf{T} & \text{if } v_{\mathcal{A}}(\phi) = \mathbf{F}, \\ \mathbf{F} & \text{if } v_{\mathcal{A}}(\phi) = \mathbf{T} \end{cases} \\
v_{\mathcal{A}}(\phi \wedge \psi) & = & \begin{cases} \mathbf{T} & \text{if } v_{\mathcal{A}}(\phi) = \mathbf{T} = v_{\mathcal{A}}(\psi) \\ \mathbf{F} & \text{if at least one of } v_{\mathcal{A}}(\phi), v_{\mathcal{A}}(\psi) = \mathbf{F}; \end{cases} \\
v_{\mathcal{A}}(\phi \vee \psi) & = & \begin{cases} \mathbf{T} & \text{if at least one of } v_{\mathcal{A}}(\phi), v_{\mathcal{A}}(\psi) = \mathbf{T}, \\ \mathbf{F} & \text{if } v_{\mathcal{A}}(\phi) = \mathbf{T} = v_{\mathcal{A}}(\psi); \end{cases} \\
v_{\mathcal{A}}(\phi \rightarrow \psi) & = & \begin{cases} \mathbf{T} & \text{if at least one of } v_{\mathcal{A}}(\phi) = \mathbf{F}, v_{\mathcal{A}}(\psi) = \mathbf{T}, \\ \mathbf{F} & \text{if } v_{\mathcal{A}}(\phi) = \mathbf{T}, v_{\mathcal{A}}(\psi) = \mathbf{F}; \end{cases} \\
v_{\mathcal{A}}(\phi \leftrightarrow \psi) & = & \begin{cases} \mathbf{T} & \text{if } v_{\mathcal{A}}(\phi) = v_{\mathcal{A}}(\psi), \\ \mathbf{F} & \text{if } v_{\mathcal{A}}(\phi) \neq v_{\mathcal{A}}(\psi); \end{cases} \\
v_{\mathcal{A}}((\exists x)\phi) & = & \begin{cases} \mathbf{T} & \text{if } v_{\mathcal{A}}(\phi^x_a) = \mathbf{T} \text{ for at least one } a \in A, \\ \mathbf{F} & \text{if } v_{\mathcal{A}}(\phi^x_a) = \mathbf{F} \text{ for all } a \in \mathcal{A}; \end{cases} \\
v_{\mathcal{A}}((\forall x)\phi) & = & \begin{cases} \mathbf{T} & \text{if } v_{\mathcal{A}}(\phi^x_a) = \mathbf{T} \text{ for all } a \in \mathcal{A}, \\ \mathbf{F} & \text{if } v_{\mathcal{A}}(\phi^x_a) = \mathbf{F} \text{ for at least one } a \in A; \end{cases}
\end{array}$$

We will say that a sentence ϕ is *true in the structure* $\mathcal{A}$ (or *satisfied in the structure* $\mathcal{A}$) if $v_{\mathcal{A}}(\phi) = \mathbf{T}$ and *false in the structure* $\mathcal{A}$ (or *unsatisfied in the structure* $\mathcal{A}$) if $v_{\mathcal{A}}(\phi) = \mathbf{F}$. Note that truth (or *satisfaction*, as $\models$ is often called) for longer sentences is always defined in terms of truth for shorter sentences (those with smaller depth). It is for the last two clauses specifying truth of quantified sentences that the assumption that all elements in the domain are named in the (expanded) language.

Example 22.2.11 Let $\mathcal{L}$ be a language which has a unary predicate P and a binary predicate R. In Example 22.2.8 we introduced a structure $\mathcal{A}$ over the universe $\{1, 2, 3\}$ which interpreted the predicates by

- $P^{\mathcal{A}}$ is the set $\{1, 3\}$,
- $R^{\mathcal{A}}$ is the set of ordered pairs $\{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 3\rangle\}$.

Since $P(1)$ is true in $\mathcal{A}$, it follows that $(\exists x)P(x)$ is also true in $\mathcal{A}$. On the other hand, $(\forall x)P(x)$ is false in $\mathcal{A}$, since $P(2)$ is false in $\mathcal{A}$. Also $(\forall x)(P(x) \rightarrow (\exists y)R(x, y))$ is false in $\mathcal{A}$, since the element 3 provides a counterexample: $P(3)$ is true in $\mathcal{A}$, but $(\exists y)R(3, y)$ is false because each of $R(3, 1), R(3, 2), R(3, 3)$ are false in $\mathcal{A}$. It is a little more involved to verify that

$$(\forall x)(\forall y)(\forall z)((R(x, y) \wedge R(y, z)) \rightarrow R(x, z))$$

There are 27 possible combinations of elements from $\{1, 2, 3\}$ that we must check. For example,

$$\begin{array}{ll}
(R(1,1) \wedge R(1,1)) \rightarrow R(1,1) & \qquad (R(1,2) \wedge R(2,1)) \rightarrow R(1,1) \\
(R(2,1) \wedge R(1,2)) \rightarrow R(2,2) & \qquad (R(1,2) \wedge R(2,3)) \rightarrow R(1,3)
\end{array}$$

You can verify that the last instance is the only one case where the antecedent is true. Verifying a universal statement is true (or an existential statement is false) is often a combinatorial nightmare, so that it is typically easier to try to show that no counterexample exists.

The structure $\mathcal{B}$ from Example 22.2.8 was over the same universe $\{1, 2, 3\}$, but interpreted $P^{\mathcal{B}} = \emptyset$. So, $(\exists x)P(x)$ is false in $\mathcal{B}$ and $(\forall x)\neg P(x)$ is true in $\mathcal{B}$. Also, $(\forall x)(P(x) \rightarrow (\exists y)R(x, y))$ is true in $\mathcal{B}$, since each of $P(1), P(2), P(3)$ are false, so that each of the following are true:

$$(P(1) \rightarrow (\exists y)R(1, y)) \quad (P(2) \rightarrow (\exists y)R(2, y)) \quad (P(2) \rightarrow (\exists y)R(2, y))$$

Definition 22.2.12 (satisfaction) All structures are structures for a fixed language $\mathcal{L}$.

A set of sentences Σ is *satisfied in a structure* $\mathcal{A}$, if each sentence ϕ of Σ is true in $\mathcal{A}$. A set of sentences Σ is *satisfiable* if there is some structure $\mathcal{A}$ which satisfies Σ. A set Σ is *unsatisfiable* if there is no structure $\mathcal{A}$ which satisfies each sentence of Σ.

Definition 22.2.13 (validity, logical consequence) All structures are structures for a fixed language $\mathcal{L}$.

A sentence ϕ is *valid*, denoted by $\models \phi$, if it is true in every structure $\mathcal{A}$.

Given a set of sentences Σ, we say that a sentence α is a *logical consequence* of Σ, denoted by $\Sigma \models \alpha$, if every structure $\mathcal{A}$ which satisfies Σ also satisfies α.

Note that if $\Sigma = \emptyset$, then $\emptyset \models \alpha$ is equivalent to α is valid. The reason is that every structure $\mathcal{A}$ satisfies each of the sentences in $\emptyset$. We will write $\models \alpha$ instead of $\emptyset \models \alpha$.

Lemma 22.2.14 The following statements are proved in exactly the same manner as the corresponding statements in propositional logic from Lecture 4.

1. α is valid if and only if $\neg\,\alpha$ is not satisfiable.
2. α is satisfiable if and only if $\neg\,\alpha$ is not valid.
3. $\Sigma \models \alpha$ if and only if $\Sigma \cup \{\neg\,\alpha\}$ is unsatisfiable.

Example 22.2.15 Let R be a binary relation symbol. Consider the following sentences:

$$\begin{array}{ll}
(a) & (\forall x)R(x, x) \\
(b) & (\forall x)\neg\, R(x, x) \\
(c) & (\forall x)(\forall y)(R(x, y) \rightarrow R(y, x)) \\
(d) & (\forall x)(\forall y)(R(x, y) \rightarrow \neg\, R(y, x)) \\
(e) & (\forall x)(\forall y)(\forall z)((R(x, y) \wedge R(y, z)) \rightarrow R(x, z))
\end{array}$$

The set $\{(a), (b)\}$ is unsatisfiable, because of our insistence that the universe of a structure is nonempty. If $\mathcal{A}$ is any structure (for this language), then there is an element $a \in A$, so that either $R(a, a)$ is true in $\mathcal{A}$ and (b) is false, or $R(a, a)$ is false in $\mathcal{A}$ and (a) is false as well. The set $\{(c), (d)\}$ is satisfiable: let the structure $\mathcal{A}$ have domain $A = \{1\}$ and interpret $R^{\mathcal{A}} = \emptyset$. Since $R(1, 1)$ is false in $\mathcal{A}$, so each of (b) and (c) are true in $\mathcal{A}$. Also, $\{(a), (d)\}$ is unsatisfiable: if $\mathcal{A}$ is any structure and $a \in A$, then $R(a, a)$ is true in $\mathcal{A}$, means $R(a, a) \rightarrow R(a, a)$ is true as well, so provides a counterexample to (d).

The structure $\mathcal{A}$ whose universe is $A = \{1, 2, 3\}$ and which interprets $R^{\mathcal{A}} = \{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 3\rangle\}$ satisfies $\{(b), (d), (e)\}$. On the other hand, $\{(b), (c), (e)\}$ is satisfiable in any structure $\mathcal{A}$ in which $R^{\mathcal{A}} = \emptyset$, but in not otherwise. Suppose $R(a, b)$ holds for some a, b in the domain, and that (c) and (e) are true. Then $R(b, a)$ is true by (c), and so $R(a, a)$ and $R(b, b)$ must be true by (e).

Finally, $\{(c), (e)\} \not\models (a)$. Any structure $\mathcal{A}$ for which $R^{\mathcal{A}} = \emptyset$ will satisfy (c) and (e), but not (a). A (slightly) less trivial result is the structure $\mathcal{A}$ with domain $\{1, 2\}$ and $R^{\mathcal{A}} = \{\langle 1, 1\rangle\}$. On the other hand $\{(c), (e), (\forall x)(\exists y)R(x, y)\} \models (a)$: Let $\mathcal{A}$ be any structure satisfying $\{(c), (e), (\forall x)(\exists y)R(x, y)\}$, and let $a \in A$ be any element of the domain. Then $R(a, b)$ is true in $\mathcal{A}$ for some $b \in A$ because $(\forall x)(\exists y)R(x, y)$ is true in $\mathcal{A}$. It now follows from (c) and (e), as above, that $R(a, a)$ true in $\mathcal{A}$ as well.

Remark 22.2.16 An interpretation provides a "meaning" for the sentences it interprets, although the simple example we have given so far hardly provide an "intelligible" reading of first-order sentences. For this, we turn to structures which are more familiar to us.

Example 22.2.17 Let $\mathcal{L}$ be a language which has one binary predicate symbol R.

(1) One structure for $\mathcal{L}$ consists of the domain $\mathbb{N}$ where $R^{\mathcal{A}}$ is the usual less than relation $<$. The sentence $(\forall x)(\exists y)R(x, y)$ says that for every natural number there is a large one, which is true in this structure.

(2) Another structure for $\mathcal{L}$ consists of the rational numbers $\mathbb{Q}$ where $R^{\mathcal{A}}$ is again $<$. The sentence

$$(\forall x)(\forall y)\big(R(x, y) \rightarrow (\exists z)(R(x, z) \wedge R(z, y))\big)$$

says that the rationals are *dense*: between any two distinct rational numbers there is a third that is between them. This sentence is true in this structure, but it is not valid, since it is not true in the structure of (1).

In each of (1) and (2), *proving* that the sentences given are true requires deeper insight into the properties of the structures themselves. This is one reason why questions of truth in a structure, logical consequence and validity are so hard to establish, when based on the definitions given in this lecture. Unlike the case of propositional logic, it can be quite difficult to even provide interpretations in which some selected sentences are true and others false. There are no exhaustive search procedures, as we had with truth tables; the search for counterexamples in first-order logic is open-ended. It is even possible that a sentence requires an infinite domain for it to be true. For example, Homework 4 asks you to prove that the three sentences

$$(\forall x)\neg R(x, x)$$
$$(\forall x)(\forall y)(\forall z)((R(x, y) \wedge R(x, z)) \rightarrow R(x, z))$$
$$(\forall x)(\exists y)R(x, y)$$

are *only jointly satisfied* in a domain with infinitely many elements.

3 Reduction and Expansion of a Structure

Definition 22.3.1 Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages such that $\mathcal{L}_1 \subseteq \mathcal{L}_2$. For any structure $\mathcal{A}$ for $\mathcal{L}_2$ the *reduction of $\mathcal{A}$ to $\mathcal{L}_1$* is the structure obtained when we forget the interpretation given to the symbols in $\mathcal{L}_2 - \mathcal{L}_1$. For any structure $\mathcal{A}$ for $\mathcal{L}_1$ an *expansion of $\mathcal{A}$ to $\mathcal{L}_2$* is a structure $\mathcal{B}$ for $\mathcal{L}_2$ with the same domain as A and the same interpretation for each symbol in $\mathcal{L}_1$.

The importance of reductions is the following Lemma (compare to Lemma 4.1.4 for Boolean assignments).

Definition 22.3.2 Let α be a formula in a language $\mathcal{L}$. The *support of* α is the subset of $\mathcal{L}$ which includes all symbols which actually occur in α.

Lemma 22.3.3 Let α be a sentence in a language $\mathcal{L}$. For any structure $\mathcal{A}$ for $\mathcal{L}$ with restriction $\mathcal{A}^*$ to the support of α,

$$v_{\mathcal{A}}(\alpha) = v_{\mathcal{A}^*}(\alpha).$$

Proof. The proof is by structural induction in the extended sense Theorem 21.3.4. Let $\mathcal{A}$ be any structure for $\mathcal{L}$. Let $R(t_1, \ldots, t_n)$ be an atomic formula in $\mathcal{L}^A$ and $\mathcal{A}^*$ be the restriction to the symbols in $R(t_1, \ldots, t_n)$. So $[t_i]^{\mathcal{A}} = [t_i]^{\mathcal{A}^*}$ for each i and $R^{\mathcal{A}} = R^{\mathcal{A}*}$,

$$v_{\mathcal{A}}(R(t_1, \ldots, t_n)) = \mathbf{T} \text{ iff } \langle [t_1]^{\mathcal{A}} \ldots, [t_n]^{\mathcal{A}} \in R^{\mathcal{A}} \rangle \text{ iff } \langle [t_1]^{\mathcal{A}^*} \ldots, [t_n]^{\mathcal{A}^*} \rangle \in R^{\mathcal{A}^*} \text{ iff } v_{\mathcal{A}^*}(R(t_1, \ldots, t_n)).$$

The case of the propositional connectives is exactly as in the proof of Lemma 4.1.4 for propositions. For example, if $v_{\mathcal{A}}(\alpha) = v_{\mathcal{A}^*}(\alpha)$, then

$$v_{\mathcal{A}}(\neg\,\alpha) = \mathbf{T} \text{ iff } v_{\mathcal{A}}(\alpha) = \mathbf{F} \text{ iff } v_{\mathcal{A}^*}(\alpha) = \mathbf{F} \text{ iff } v_{\mathcal{A}^*}(\neg\,\alpha) = \mathbf{T}.$$

For the quantifiers we suppose that $v_{\mathcal{A}}(\alpha^x_a) = v_{\mathcal{A}^*}(\alpha^x_a)$ for each a in the domain A (which is the same as the domain of $\mathcal{A}^*$). Then,

$$\begin{array}{rll} v_{\mathcal{A}}((\forall x)\alpha) = \mathbf{T} & \text{iff} & v_{\mathcal{A}}(\alpha^x_a) = \mathbf{T} \qquad \text{for each } a \in A \\ & \text{iff} & v_{\mathcal{A}^*}(\alpha^x_a) = \mathbf{T} \qquad \text{for each } a \in A \\ & \text{iff} & v_{\mathcal{A}^*}((\forall x)\alpha) = \mathbf{T} \end{array}$$

The case of the existential quantifier follows the same pattern. □

The importance of expansions is when we extend a language by adding new constant symbols. This is discussed in more detail in the next lecture.

Remark 22.3.4 We often are interested in structures which have the same universe A. For example, our semantic tableaux will produce structures with the same universe. A celebrated result of Löwenheim is that if α is any satisfiable sentence, then α is satisfiable in some countable domain. Skolem extended this result to any countable set of sentences which is satisfiable is satisfiable in a countable domain. These results have had a profound impact on the foundations of mathematics. For example, if Σ is the set of axioms for Zermelo-Frankel set theory (which has a countable set of axioms sufficient for proving all the theorems in mathematics), then if Σ is satisfiable (all mathematicians believe it is satisfiable, just take the universe of mathematics!) it is satisfiable in a *countable domain.* Of course, there are uncountably many recognized mathematical objects. For example, the set of real numbers alone is uncountable.

Definition 22.3.5 Let A be a nonempty set. A sentence α is *valid* in a domain A if it is valid in all structures with domain $\mathcal{A}$ which provide an interpretation for the support of α. A sentence α is *satisfiable* if there is at least one interpretation with domain A which provides an interpretation of the support of α in which α is true.

So, α is valid if and only if it is valid in every domain; and α is satisfiable if and only if it is satisfiable in some domain.

The next lemma states some basic facts about satisfiability and validity in a domain. We say that two sets A and B have *same cardinality* (or size) if there is a 1-1 and onto map $f : A \to B$.

Lemma 22.3.6 Let A be a nonempty set and α any sentence.

1. Validity and satisfiability of a sentence in A depends only on the cardinality of A. That is, if B is a set with the same cardinality as A, α is valid in A if and only if it is valid in B, and α is satisfiable in A if and only if it is satisfiable in B.
2. If α is satisfiable in A and $B \supseteq A$, then α is satisfiable in B.
3. If α is valid in A and $C \subseteq A$, then α is valid in C.

Proof. The proof of (2) and (3) were given as advanced problems on Homework 4.

(1). Let $f : A \to B$ be a 1-1 and onto map. The $f^{-1} : B \to A$ exists and is also 1-1 and onto. Fix a language $\mathcal{L}$. We will show that for any structure $\mathcal{A}$ for $\mathcal{L}$ with domain A, there is a structure $\mathcal{B}$ for $\mathcal{L}$ with domain B such that $v_{\mathcal{A}}(\alpha) = v_{\mathcal{B}}(\alpha)$ for any α in $\mathcal{L}^{\mathcal{A}}$. Let $\mathcal{A}$ be a structure for $\mathcal{L}$. Interpret each a in A by $f(a)$ in B, each constant $c^{\mathcal{A}}$ by $f(c^{\mathcal{A}})$ and for each n-place predicate of $\mathcal{L}$:

$$R^{\mathcal{B}} = \{\langle f(a_1), \ldots, f(a_n)\rangle : \langle a_1, \ldots, a_n\rangle \in R^{\mathcal{A}}\}$$

The proof that $v_{\mathcal{A}}(\alpha) = v_{\mathcal{B}}(\alpha)$ for each sentence of $\mathcal{L}^A$ is by structural induction in the extended sense Theorem 21.3.4.

First note that if t is any term in $\mathcal{L}$, then $f([t]^{\mathcal{A}}) = [t]^{\mathcal{B}}$ by definition of $\mathcal{B}$. (That is, if $\mathcal{A}$ interprets t by a, then $\mathcal{B}$ interprets t by b.) For any n-place predicate,

$$\langle a_1, \ldots, a_n \rangle \in R^{\mathcal{A}} \text{ iff } \langle f(a_1), \ldots, f(a_n) \rangle \in R^{\mathcal{B}},$$

so for any atomic sentence $R(t_1, \ldots, t_n)$ in $\mathcal{L}^A$,

$$\langle [t_1]^{\mathcal{A}}, \ldots, [t_n]^{\mathcal{A}} \rangle \in R^{\mathcal{A}} \text{ iff } \langle [t_1]^{\mathcal{B}}, \ldots, [t_n]^{\mathcal{B}} \rangle \in R^{\mathcal{B}},$$

So, the case of atomic sentences is proved.

The case of propositional connectives is straightforward. I will do the case of negation. Suppose $v_{\mathcal{A}}(\alpha) = v_{\mathcal{B}}(\alpha)$, then

$$v_{\mathcal{A}}(\neg\,\alpha) = \mathbf{T} \text{ iff } v_{\mathcal{A}}(\alpha) = \mathbf{F} \text{ iff } v_{\mathcal{B}}(\alpha) = \mathbf{F} \text{ iff } v_{\mathcal{B}}(\neg\,\alpha) = \mathbf{T}$$

For the quantifiers we assume $v_{\mathcal{A}}(\alpha^x_a) = v_{\mathcal{B}}(\alpha^x_a)$. Since $a^{\mathcal{B}} = f(a)$, we have $v_{\mathcal{B}}(\alpha^x_a) = v_{\mathcal{B}}(\alpha^x_{f(a)}$. Then

$$\begin{array}{lll} v_{\mathcal{A}}((\forall x)\alpha) = \mathbf{T} & \text{iff} & v_{\mathcal{A}}(\alpha^x_a) = \mathbf{T} \quad \text{for each } a \in A \\ & \text{iff} & v_{\mathcal{B}}(\alpha^x_{f(a)}) = \mathbf{T} \quad \text{for each } a \in A \\ & \text{iff} & v_{\mathcal{B}}(\alpha^x_b) = \mathbf{T} \quad \text{for each } b \in B \\ & \text{iff} & v_{\mathcal{B}}((\forall x)\alpha) = \mathbf{T}. \end{array}$$

The second line is by the inductive hypothesis, the third line is because f maps A onto B, so for each $b \in B$ there is some $a \in A$ with $f(a) = b$. The case of the existential quantifier is similar.

Although we did not use the fact that f is 1-1 in this argument, we also need to show that for each structure $\mathcal{B}$ for $\mathcal{L}$ with domain B there is a structure $\mathcal{A}$ with domain A with $v_{\mathcal{B}}(\alpha) = v_{\mathcal{A}}(\alpha)$ for each α in $\mathcal{L}$. The proof switches the roles of $\mathcal{A}$ and $\mathcal{B}$ using $f^{-1} : B \to A$, which maps B onto A. The existence of f^{-1} does depend on the fact that f is 1-1. □