First-Order Logic: Syntax and Semantics

Yuting Wang

John Hopcroft Center for Computer Science
Shanghai Jiao Tong University

Nov 7, 2022

First-Order Logic

Start reading (to keep up with lecture):

- Enderton, Chapter 2

Example

Question

Premises:

- If it is raining or it is snowing then the sun is not shining.
- It is raining

Conclusion: The sun is not shining

Is the conclusion a semantic consequence of the premises?

Example

Question

Premises:

- If it is raining or it is snowing then the sun is not shining.
- It is raining

Conclusion: The sun is not shining

Is the conclusion a semantic consequence of the premises?

Answer

Yes.
Let A, B, and C represent "it is raining", "it is snowing" and "the sun is shining". We can prove

$$\{A \vee B \to \neg C, A\} \models \neg C$$

Another Example

Question

Premises:

- All men are mortal
- Socrates is a man.

Conclusion: Socrates is mortal.

Is the conclusion a semantic consequence of the premises?

Another Example

Question

Premises:

- All men are mortal
- Socrates is a man.

Conclusion: Socrates is mortal.

Is the conclusion a semantic consequence of the premises?

We need a more power logic to handle this kind of reasoning.

Syntax and Semantics of First-Order Logic

Recall that there are two parts to a logic:

Syntax and Semantics of First-Order Logic

Recall that there are two parts to a logic:

- Syntax. It provides
  - A description of the language, and
  - Other syntactic constructs (we will see later)

Syntax and Semantics of First-Order Logic

Recall that there are two parts to a logic:

- Syntax. It provides
  - A description of the language, and
  - Other syntactic constructs (we will see later)
- Semantics. It provides
  - A way of assigning *meaning* to *valid expressions*
  - In sentential logic, the meaning will be either TRUE or FALSE
  - In a first-order logic, the meaning may vary significantly

Let's Begin with the Syntax

Syntax of a First-Order Language $\mathbb{L}$

We start with the symbols of a first-order language $\mathbb{L}$.

Syntax of a First-Order Language $\mathbb{L}$

We start with the symbols of a first-order language $\mathbb{L}$.

There are two types of symbols:

- Logical Symbols, and
- Non-logical Symbols, also called Parameters

Logical Symbols

In a first-order language $\mathbb{L}$, we have the following logic symbols:

Logical Symbols

In a first-order language $\mathbb{L}$, we have the following logic symbols:

1. The two symbols (and), called parentheses;

Logical Symbols

In a first-order language $\mathbb{L}$, we have the following logic symbols:

1. The two symbols (and), called parentheses;
2. $\rightarrow$ and $\neg$. These are logical connective symbols;

Logical Symbols

In a first-order language $\mathbb{L}$, we have the following logic symbols:

1. The two symbols (and), called parentheses;
2. $\rightarrow$ and $\neg$. These are logical connective symbols;
3. $v_1, v_2, \ldots, v_n, \ldots$. An enumerable list of symbols called variables;

Logical Symbols

In a first-order language $\mathbb{L}$, we have the following logic symbols:

1. The two symbols (and), called parentheses;
2. $\rightarrow$ and $\neg$. These are logical connective symbols;
3. $v_1, v_2, \ldots, v_n, \ldots$. An enumerable list of symbols called variables;
4. $=$. The identity or equality symbol. It may or may not be present in a particular first-order language.

Logical Symbols

In a first-order language $\mathbb{L}$, we have the following logic symbols:

1. The two symbols (and), called parentheses;
2. $\rightarrow$ and $\neg$. These are logical connective symbols;
3. $v_1, v_2, \ldots, v_n, \ldots$. An enumerable list of symbols called variables;
4. $=$. The identity or equality symbol. It may or may not be present in a particular first-order language.

We will sometimes write $\doteq$ to distinguish the equality symbol from semantic equality.

Logical Symbols

In a first-order language $\mathbb{L}$, we have the following logic symbols:

1. The two symbols (and), called parentheses;
2. $\rightarrow$ and $\neg$. These are logical connective symbols;
3. $v_1, v_2, \ldots, v_n, \ldots$. An enumerable list of symbols called variables;
4. $=$. The identity or equality symbol. It may or may not be present in a particular first-order language.

We will sometimes write $\doteq$ to distinguish the equality symbol from semantic equality.

Question

Why are the $\wedge$, $\vee$, $\leftrightarrow$ connectives not present?

Parameters

In a first-order language $\mathbb{L}$, we have the following parameters:

Parameters

In a first-order language $\mathbb{L}$, we have the following parameters:

1. $\forall$. This is called the universal quantifier;

Parameters

In a first-order language $\mathbb{L}$, we have the following parameters:

1. $\forall$. This is called the universal quantifier;
2. For each $n > 0$, there is a set (possibly empty) of objects called n-ary (or n-place) predicate symbols;

Parameters

In a first-order language $\mathbb{L}$, we have the following parameters:

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

Parameters

In a first-order language $\mathbb{L}$, we have the following parameters:

1. $\forall$. This is called the universal quantifier;
2. For each $n > 0$, there is a set (possibly empty) of objects called n-ary (or n-place) 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.

Parameters

In a first-order language $\mathbb{L}$, we have the following parameters:

1. $\forall$. This is called the universal quantifier;
2. For each $n > 0$, there is a set (possibly empty) of objects called n-ary (or n-place) 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 either a logical symbol or a parameter.

Further Requirements

We require that

Further Requirements

We require that

- $\doteq$ is a 2-ary predicate symbols;

Further Requirements

We require that

- $\doteq$ is a 2-ary predicate symbols;
- There is at least one predicate symbol;

Further Requirements

We require that

- $\doteq$ is a 2-ary predicate symbols;
- There is at least one predicate symbol;
- The symbols are distinct, and no symbol is equal to a finite sequence of other symbols.

Further Requirements

We require that

- $\doteq$ is a 2-ary predicate symbols;
- There is at least one predicate symbol;
- The symbols are distinct, and no symbol is equal to a finite sequence of other symbols.

The latter means: no 2-ary function symbol is also a 3-ary function symbol, not function symbol is a predicate symbol, etc.

Example: Set Theory

Set theory may be described by the following language:

Example: Set Theory

Set theory may be described by the following language:

- Equality: Yes;

Example: Set Theory

Set theory may be described by the following language:

- Equality: Yes;
- Predicate Symbols: a 2-place predicate symbol $\dot{\in}$;

Example: Set Theory

Set theory may be described by the following language:

- Equality: Yes;
- Predicate Symbols: a 2-place predicate symbol $\dot{\in}$;
- Constant Symbols: the empty set $\dot{\emptyset}$;

Example: Set Theory

Set theory may be described by the following language:

- Equality: Yes;
- Predicate Symbols: a 2-place predicate symbol $\dot{\in}$;
- Constant Symbols: the empty set $\dot{\emptyset}$;
- Function Symbols: None.

Example: Elementary Arithmetic

Elementary arithmetic may be described by the following language:

Example: Elementary Arithmetic

Elementary arithmetic may be described by the following language:

- Equality: Yes;

Example: Elementary Arithmetic

Elementary arithmetic may be described by the following language:

- Equality: Yes;
- Predicate Symbols: a 2-place predicate symbol $\dot{<}$;

Example: Elementary Arithmetic

Elementary arithmetic may be described by the following language:

- Equality: Yes;
- Predicate Symbols: a 2-place predicate symbol $\dot{<}$;
- Constant Symbols: $\dot{0}$ and $\dot{1}$;

Example: Elementary Arithmetic

Elementary arithmetic may be described by the following language:

- Equality: Yes;
- Predicate Symbols: a 2-place predicate symbol $\dot{<}$;
- Constant Symbols: $\dot{0}$ and $\dot{1}$;
- Function Symbols: 2-ary function symbols $\dot{+}$ and $\dot{\times}$.

Non-Requirement

Remark

We have not put any requirement on the number of function, predicate or constant symbols.

Non-Requirement

Remark

We have not put any requirement on the number of function, predicate or constant symbols.

Example

For example, it is possible for a given first-order language:

Non-Requirement

Remark

We have not put any requirement on the number of function, predicate or constant symbols.

Example

For example, it is possible for a given first-order language:

- There are *three* constant symbols;

Non-Requirement

Remark

We have not put any requirement on the number of function, predicate or constant symbols.

Example

For example, it is possible for a given first-order language:

- There are *three* constant symbols;
- There are *enumerably many* function symbols;

Non-Requirement

Remark

We have not put any requirement on the number of function, predicate or constant symbols.

Example

For example, it is possible for a given first-order language:

- There are *three* constant symbols;
- There are *enumerably many* function symbols;
- There are *uncountably many* predicate symbols!

Expressions

Like in sentential logic, an expression in a language $\mathbb{L}$ is a finite sequence of symbols.

Example

$\forall\neg\rightarrow v_1 v_2 v_4$ is an expression.

Terms

Definition

Given any n-ary function symbol f, the term-building operation $\mathcal{F}_f$ is defined by:

$$\mathcal{F}_f(\sigma_1, \ldots, \sigma_n) = f\ \sigma_1 \ldots \sigma_n$$

We call σ_i the arguments to f.

Terms

Definition

Given any n-ary function symbol f, the term-building operation $\mathcal{F}_f$ is defined by:

$$\mathcal{F}_f(\sigma_1, \ldots, \sigma_n) = f\ \sigma_1 \ldots \sigma_n$$

We call σ_i the arguments to f.

Definition

A term is an expression built up from constant symbols and variables by applying some finite times of term-building operations.

Example

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.

Term Sequences

An alternative way to define terms is to use *term sequences*.

Term Sequences

An alternative way to define terms is to use *term sequences*.

Definition

A term sequence is a finite sequence $t_1, \ldots, t_n$ of expressions s.t. each t_i is either

- a variable, or a constant symbol, or
- is of the form $f\ \sigma_1 \ldots \sigma_k$ where f is a k-ary function symbol and each of $\sigma_1 \ldots \sigma_k$ occurs earlier in the sequence.

Term Sequences

An alternative way to define terms is to use *term sequences*.

Definition

A term sequence is a finite sequence $t_1, \ldots, t_n$ of expressions s.t. each t_i is either

- a variable, or a constant symbol, or
- is of the form $f\ \sigma_1 \ldots \sigma_k$ where f is a k-ary function symbol and each of $\sigma_1 \ldots \sigma_k$ occurs earlier in the sequence.

Proposition

An expression t is a term iff there is a term sequence $t_1, \ldots, t_n$ such that $t = t_n$.

Example

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

Example

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.

Atomic Formulas

Definition

An expression is an atomic formula if it is of the form $P\ t_1 \ldots t_n$ where $t_1, \ldots, t_n$ are terms, and P is a n-ary predicate symbol.

Atomic Formulas

Definition

An expression is an atomic formula if it is of the form $P\ t_1 \ldots t_n$ where $t_1, \ldots, t_n$ are terms, and P is a n-ary predicate symbol.

Example

- $\doteq v_7 v_3$ is an atomic formula (consisting of 3 symbols);

Atomic Formulas

Definition

An expression is an atomic formula if it is of the form $P\ t_1 \ldots t_n$ where $t_1, \ldots, t_n$ are terms, and P is a n-ary predicate symbol.

Example

- $\doteq v_7 v_3$ is an atomic formula (consisting of 3 symbols);
- If c_1 and c_3 are constant symbols and f is a 2-ary function symbol, then $\doteq f c_1 v_7 c_3$ is an atomic formula.

Well-Formed Formulas

Definition

The *formula-building operations* include the following:

- $\xi_{\neg}(\alpha) = (\neg\alpha)$
- $\xi_{\rightarrow}(\alpha, \beta) = (\alpha \rightarrow \beta)$
- $\mathcal{Q}_i(\gamma) = \forall v_i \gamma$

Well-Formed Formulas

Definition

The *formula-building operations* include the following:

- $\xi_{\neg}(\alpha) = (\neg\alpha)$
- $\xi_{\rightarrow}(\alpha, \beta) = (\alpha \rightarrow \beta)$
- $\mathcal{Q}_i(\gamma) = \forall v_i \gamma$

Definition

A well-formed formula (wff) is an expression built up from atomic formulas by applying some finite times of term-building operations $\xi_{\neg}$, $\xi_{\rightarrow}$ and $\mathcal{Q}_i (i = 0, 1, \ldots)$.

Example

Example

The expression $(\neg\forall v_3 \doteq v_1 v_2)$ is a wff.

Example

Example

The expression $(\neg\forall v_3 \doteq v_1 v_2)$ is a wff.

Example

The expression $(\neg(\forall v_3 \doteq v_1 v_2))$ is not a wff.

Well-Formed Sequences

An alternative way to define wffs is to use *well-formed sequences*.

Well-Formed Sequences

An alternative way to define wffs is to use *well-formed sequences*.

Definition

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

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

Well-Formed Sequences

An alternative way to define wffs is to use *well-formed sequences*.

Definition

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

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

Proposition

The expression α is a wff if there is a well-formed sequence $\alpha_1, \ldots, \alpha_k$ such that $\alpha = \alpha_k$.

Example

Example

The expression $(\neg\forall v_3 \doteq v_1 v_2)$ is a wff

Example

Example

The expression $(\neg\forall v_3 \doteq v_1 v_2)$ is a wff since

$$\doteq v_1 v_2, \forall v_3 \doteq v_1 v_2, (\neg\forall v_3 \doteq v_1 v_2)$$

is a well-formed sequence.

Abbreviations

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;

Abbreviations

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;

Abbreviations

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$;

Abbreviations

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \rightarrow \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \rightarrow (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$;
- $\exists x\alpha$ abbreviates $(\neg\forall x(\neg\alpha))$;

Abbreviations

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$;
- $\exists x\alpha$ abbreviates $(\neg\forall x(\neg\alpha))$;
- $u \doteq t$ abbreviates $\doteq ut$;

Abbreviations

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$;
- $\exists x\alpha$ abbreviates $(\neg\forall x(\neg\alpha))$;
- $u \doteq t$ abbreviates $\doteq ut$;
- $u \dot{\neq} t$ abbreviates $(\neg \doteq ut)$;

Abbreviations

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$;
- $\exists x \alpha$ abbreviates $(\neg\forall x(\neg\alpha))$;
- $u \doteq t$ abbreviates $\doteq ut$;
- $u \dot{\neq} t$ abbreviates $(\neg \doteq ut)$;

We also often write:

- $u \dot{+} v$ for $\dot{+} uv$;

Abbreviations

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$;
- $\exists x \alpha$ abbreviates $(\neg\forall x(\neg\alpha))$;
- $u \dot{=} t$ abbreviates $\dot{=} ut$;
- $u \dot{\neq} t$ abbreviates $(\neg \dot{=} ut)$;

We also often write:

- $u \dot{+} v$ for $\dot{+} uv$;
- $u \dot{\times} v$ for $\dot{\times} uv$;

Abbreviations

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$;
- $\exists x\alpha$ abbreviates $(\neg\forall x(\neg\alpha))$;
- $u \doteq t$ abbreviates $\doteq ut$;
- $u \dot{\neq} t$ abbreviates $(\neg \doteq ut)$;

We also often write:

- $u \dot{+} v$ for $\dot{+} uv$;
- $u \dot{\times} v$ for $\dot{\times} uv$;
- $u \dot{<} v$ for $\dot{<} uv$.

More Abbreviations

(1) We may drop the outermost parentheses;

(2) $\neg$, $\forall$, $\exists$ apply to as little as possible;

(3) $\wedge$ and $\vee$ apply to as little as possible; subject to (2);

(4) when one connective is used repeatedly, grouping is to the right.

Example: Set Theory

In set theory, we have the symbols $\dot{=}$, $\dot{\emptyset}$ and $\dot{\in}$.

Example: Set Theory

In set theory, we have the symbols $\dot{=}$, $\dot{\emptyset}$ and $\dot{\in}$. Moreover

- $\forall$ means "for every set";
- $\dot{\in}$ means "is a member of";

Example: Set Theory

In set theory, we have the symbols $\dot{=}$, $\dot{\emptyset}$ and $\dot{\in}$. Moreover

- $\forall$ means "for every set";
- $\dot{\in}$ means "is a member of";

Question

How to interpret the sentence "There is no set of which every set is a member" in first-order logic?

Example: Set Theory

In set theory, we have the symbols $\dot{=}$, $\dot{\emptyset}$ and $\dot{\in}$. Moreover

- $\forall$ means "for every set";
- $\dot{\in}$ means "is a member of";

Question

How to interpret the sentence "There is no set of which every set is a member" in first-order logic?

Answer

Example: Set Theory

In set theory, we have the symbols $\dot{=}$, $\dot{\emptyset}$ and $\dot{\in}$. Moreover

- $\forall$ means "for every set";
- $\dot{\in}$ means "is a member of";

Question

How to interpret the sentence "There is no set of which every set is a member" in first-order logic?

Answer

- $(\neg$[There is some set of which every set is a member]$)$;

Example: Set Theory

In set theory, we have the symbols $\dot{=}$, $\dot{\emptyset}$ and $\dot{\in}$. Moreover

- $\forall$ means "for every set";
- $\dot{\in}$ means "is a member of";

Question

How to interpret the sentence "There is no set of which every set is a member" in first-order logic?

Answer

- $(\neg$[There is some set of which every set is a member]);
- $(\neg\exists v_1$[Every set is a member of v_1]);

Example: Set Theory

In set theory, we have the symbols $\dot{=}$, $\dot{\emptyset}$ and $\dot{\in}$. Moreover

- $\forall$ means "for every set";
- $\dot{\in}$ means "is a member of";

Question

How to interpret the sentence "There is no set of which every set is a member" in first-order logic?

Answer

- ($\neg$[There is some set of which every set is a member]);
- ($\neg\exists v_1$[Every set is a member of v_1]);
- ($\neg\exists v_1\forall v_2$[$v_2$ is a member of v_1]);

Example: Set Theory

In set theory, we have the symbols $\dot{=}$, $\dot{\emptyset}$ and $\dot{\in}$. Moreover

- $\forall$ means "for every set";
- $\dot{\in}$ means "is a member of";

Question

How to interpret the sentence "There is no set of which every set is a member" in first-order logic?

Answer

- $(\neg$[There is some set of which every set is a member]$)$;
- $(\neg\exists v_1$[Every set is a member of v_1]$)$;
- $(\neg\exists v_1\forall v_2[v_2$ is a member of $v_1])$;
- $(\neg\exists v_1\forall v_2, v_2\dot{\in}v_1)$;

Example: Set Theory

In set theory, we have the symbols $\dot{=}$, $\dot{\emptyset}$ and $\dot{\in}$. Moreover

- $\forall$ means "for every set";
- $\dot{\in}$ means "is a member of";

Question

How to interpret the sentence "There is no set of which every set is a member" in first-order logic?

Answer

- $(\neg$[There is some set of which every set is a member]);
- $(\neg\exists v_1$[Every set is a member of v_1]);
- $(\neg\exists v_1\forall v_2[v_2$ is a member of v_1]);
- $(\neg\exists v_1\forall v_2, v_2\dot{\in}v_1)$;
- $(\neg(\neg\forall v_1(\neg\forall v_2, \dot{\in}v_2 v_1)))$;

Example: Elementary Arithmetic

In elementary arithmetic, we have the symbols $\dot{<}$, $\dot{+}$, $\dot{\times}$, $\dot{0}$ and $\dot{1}$.

Example: Elementary Arithmetic

In elementary arithmetic, we have the symbols $\dot{<}$, $\dot{+}$, $\dot{\times}$, $\dot{0}$ and $\dot{1}$.

- We represent a natural number n by

$$\underbrace{\dot{1}\dot{+}\ldots\dot{+}\dot{1}}_{n \text{ times}}\dot{+}\dot{0}$$

Example: Elementary Arithmetic

In elementary arithmetic, we have the symbols $\dot{<}$, $\dot{+}$, $\dot{\times}$, $\dot{0}$ and $\dot{1}$.

- We represent a natural number n by

$$\underbrace{\dot{1}\dot{+}\ldots\dot{+}\dot{1}}_{n \text{ times}}\dot{+}\dot{0}$$

Therefore, $2+1=3$ is interpreted as

Example: Elementary Arithmetic

In elementary arithmetic, we have the symbols $\dot{<}$, $\dot{+}$, $\dot{\times}$, $\dot{0}$ and $\dot{1}$.

- We represent a natural number n by

$$\underbrace{\dot{1}\dot{+}\ldots\dot{+}\dot{1}}_{n \text{ times}}\dot{+}\dot{0}$$

Therefore, $2+1=3$ is interpreted as

$$(\dot{1}\dot{+}\dot{1}\dot{+}\dot{0})\dot{+}(\dot{1}\dot{+}\dot{0})\dot{=}(\dot{1}\dot{+}\dot{1}\dot{+}\dot{1}\dot{+}\dot{0})$$

Example: Elementary Arithmetic

In elementary arithmetic, we have the symbols $\dot{<}$, $\dot{+}$, $\dot{\times}$, $\dot{0}$ and $\dot{1}$.

- We represent a natural number n by

$$\underbrace{\dot{1}\dot{+}\ldots\dot{+}\dot{1}}_{n \text{ times}}\dot{+}\dot{0}$$

Therefore, $2 + 1 = 3$ is interpreted as

$$(\dot{1}\dot{+}\dot{1}\dot{+}\dot{0})\dot{+}(\dot{1}\dot{+}\dot{0})\dot{=}(\dot{1}\dot{+}\dot{1}\dot{+}\dot{1}\dot{+}\dot{0})$$

- We interpret the sentence "any non-zero natural number is the successor of some natural number" as follows:

Example: Elementary Arithmetic

In elementary arithmetic, we have the symbols $\dot{<}$, $\dot{+}$, $\dot{\times}$, $\dot{0}$ and $\dot{1}$.

- We represent a natural number n by

$$\underbrace{\dot{1}\dot{+}\ldots\dot{+}\dot{1}}_{n \text{ times}}\dot{+}\dot{0}$$

  Therefore, $2 + 1 = 3$ is interpreted as

$$(\dot{1}\dot{+}\dot{1}\dot{+}\dot{0})\dot{+}(\dot{1}\dot{+}\dot{0})\dot{=}(\dot{1}\dot{+}\dot{1}\dot{+}\dot{1}\dot{+}\dot{0})$$

- We interpret the sentence "any non-zero natural number is the successor of some natural number" as follows:
  - $\forall v_1$[if v_1 is non-zero, then v_1 is the successor of some number]

Example: Elementary Arithmetic

In elementary arithmetic, we have the symbols $\dot{<}$, $\dot{+}$, $\dot{\times}$, $\dot{0}$ and $\dot{1}$.

- We represent a natural number n by

$$\underbrace{\dot{1}\dot{+}\ldots\dot{+}\dot{1}}_{n \text{ times}}\dot{+}\dot{0}$$

  Therefore, $2+1=3$ is interpreted as

$$(\dot{1}\dot{+}\dot{1}\dot{+}\dot{0})\dot{+}(\dot{1}\dot{+}\dot{0})\dot{=}(\dot{1}\dot{+}\dot{1}\dot{+}\dot{1}\dot{+}\dot{0})$$

- We interpret the sentence "any non-zero natural number is the successor of some natural number" as follows:
  - $\forall v_1$[if v_1 is non-zero, then v_1 is the successor of some number]
  - $\forall v_1(v_1 \dot{\neq} \dot{0} \to \exists v_2 v_1 \dot{=} \dot{1} \dot{+} v_2)$

Example: Elementary Arithmetic

In elementary arithmetic, we have the symbols $\dot{<}$, $\dot{+}$, $\dot{\times}$, $\dot{0}$ and $\dot{1}$.

- We represent a natural number n by

$$\underbrace{\dot{1}\dot{+}\ldots\dot{+}\dot{1}}_{n \text{ times}}\dot{+}\dot{0}$$

  Therefore, $2 + 1 = 3$ is interpreted as

$$(\dot{1}\dot{+}\dot{1}\dot{+}\dot{0})\dot{+}(\dot{1}\dot{+}\dot{0})\dot{=}(\dot{1}\dot{+}\dot{1}\dot{+}\dot{1}\dot{+}\dot{0})$$

- We interpret the sentence "any non-zero natural number is the successor of some natural number" as follows:
  - $\forall v_1$[if v_1 is non-zero, then v_1 is the successor of some number]
  - $\forall v_1(v_1 \dot{\neq} \dot{0} \to \exists v_2 v_1 \dot{=} \dot{1} \dot{+} v_2)$
  - $\forall v_1((\neg \dot{=} v_1 \dot{0}) \to (\neg \forall v_2 (\neg (\dot{=} v_1 \dot{+} \dot{1} v_2))))$

Example: Mortality of Men

Assume the language $\mathbb{L}$ contains the following symbols:

- $\dot{P}$: 1-ary predicate symbol for asserting whether a being is a man;
- $\dot{Q}$: 1-ary predicate symbol for asserting whether a being is a mortal;

Example: Mortality of Men

Assume the language $\mathbb{L}$ contains the following symbols:

- $\dot{P}$: 1-ary predicate symbol for asserting whether a being is a man;
- $\dot{Q}$: 1-ary predicate symbol for asserting whether a being is a mortal;

Question

How to interpret the sentence "All men are mortal"?

Example: Mortality of Men

Assume the language $\mathbb{L}$ contains the following symbols:

- $\dot{P}$: 1-ary predicate symbol for asserting whether a being is a man;
- $\dot{Q}$: 1-ary predicate symbol for asserting whether a being is a mortal;

Question

How to interpret the sentence "All men are mortal"?

Answer

- $\forall v_1$ [if v_1 is a man then v_1 is mortal];
- $\forall v_1$ [v_1 is a man] $\to$ [v_1 is mortal];
- $\forall v_1\ (\dot{P}v_1 \to \dot{Q}v_1)$.

Summary of Syntax

We introduced the symbols of a first-order language $\mathbb{L}$, and definitions of:

- Terms
- Atomic Formulas
- Well-Formed Formulas (wffs)

What about Semantics of First-Order Logic?

Semantics: Definition of Structures

Let $\mathbb{L}$ be a first-order language.

Semantics: Definition of Structures

Let $\mathbb{L}$ be a first-order language.

Definition

A structure $\mathfrak{A}$ for $\mathbb{L}$ consists of:

- a non-empty set called the universe (or *domain*) of the structure and usually written as $|\mathfrak{A}|$;

Semantics: Definition of Structures

Let $\mathbb{L}$ be a first-order language.

Definition

A structure $\mathfrak{A}$ for $\mathbb{L}$ consists of:

- a non-empty set called the universe (or *domain*) of the structure and usually written as $|\mathfrak{A}|$;
- for each n-ary predicate symbol P of $\mathbb{L}$, other than $\doteq$, an n-ary relation $P^{\mathfrak{A}}$ on $|\mathfrak{A}|$;

Semantics: Definition of Structures

Let $\mathbb{L}$ be a first-order language.

Definition

A structure $\mathfrak{A}$ for $\mathbb{L}$ consists of:

- a non-empty set called the universe (or *domain*) of the structure and usually written as $|\mathfrak{A}|$;
- for each n-ary predicate symbol P of $\mathbb{L}$, other than $\doteq$, an n-ary relation $P^{\mathfrak{A}}$ on $|\mathfrak{A}|$;
- $\doteq^{\mathfrak{A}}$ is the identity relation on $|\mathfrak{A}|$, i.e., $\doteq^{\mathfrak{A}} = \{(a, b) \mid a, b \in |\mathfrak{A}| \text{ and } a = b\}$;

Semantics: Definition of Structures

Let $\mathbb{L}$ be a first-order language.

Definition

A structure $\mathfrak{A}$ for $\mathbb{L}$ consists of:

- a non-empty set called the universe (or *domain*) of the structure and usually written as $|\mathfrak{A}|$;
- for each n-ary predicate symbol P of $\mathbb{L}$, other than $\doteq$, an n-ary relation $P^{\mathfrak{A}}$ on $|\mathfrak{A}|$;
- $\doteq^{\mathfrak{A}}$ is the identity relation on $|\mathfrak{A}|$, i.e., $\doteq^{\mathfrak{A}} = \{(a, b) \mid a, b \in |\mathfrak{A}| \text{ and } a = b\}$;
- for each n-ary function symbol f of $\mathbb{L}$, an n-ary operation on the universe, i.e., an n-ary function $f^{\mathfrak{A}} : \underbrace{|\mathfrak{A}| \times \ldots \times |\mathfrak{A}|}_{n} \to |\mathfrak{A}|$;

Semantics: Definition of Structures

Let $\mathbb{L}$ be a first-order language.

Definition

A structure $\mathfrak{A}$ for $\mathbb{L}$ consists of:

- a non-empty set called the universe (or *domain*) of the structure and usually written as $|\mathfrak{A}|$;
- for each n-ary predicate symbol P of $\mathbb{L}$, other than $\doteq$, an n-ary relation $P^{\mathfrak{A}}$ on $|\mathfrak{A}|$;
- $\doteq^{\mathfrak{A}}$ is the identity relation on $|\mathfrak{A}|$, i.e., $\doteq^{\mathfrak{A}} = \{(a, b) \mid a, b \in |\mathfrak{A}| \text{ and } a = b\}$;
- for each n-ary function symbol f of $\mathbb{L}$, an n-ary operation on the universe, i.e., an n-ary function $f^{\mathfrak{A}} : \underbrace{|\mathfrak{A}| \times \ldots \times |\mathfrak{A}|}_{n} \to |\mathfrak{A}|$;
- for each constant symbol c of $\mathbb{L}$, $c^{\mathfrak{A}} \in |\mathfrak{A}|$.

Notation and Terminology

- $\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{C}$, $\mathfrak{M}$, $\mathfrak{N}$, $\mathfrak{Q}$, $\mathfrak{R}$ and $\mathfrak{Z}$, are the usual names we will use for structures. These are the *fraktur* (Gothic) fonts.

Notation and Terminology

- $\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{C}$, $\mathfrak{M}$, $\mathfrak{N}$, $\mathfrak{Q}$, $\mathfrak{R}$ and $\mathfrak{Z}$, are the usual names we will use for structures. These are the *fraktur* (Gothic) fonts.
- What $P^{\mathfrak{A}}$ (where $P \neq \doteq$) is changes with the structure, but $\doteq^{\mathfrak{A}}$ is always the identity relation on $|\mathfrak{A}|$.

Notation and Terminology

- $\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{C}$, $\mathfrak{M}$, $\mathfrak{N}$, $\mathfrak{Q}$, $\mathfrak{R}$ and $\mathfrak{Z}$, are the usual names we will use for structures. These are the *fraktur* (Gothic) fonts.
- What $P^{\mathfrak{A}}$ (where $P \neq \doteq$) is changes with the structure, but $\doteq^{\mathfrak{A}}$ is always the identity relation on $|\mathfrak{A}|$.
- We say P *denotes* (or stands for) $P^{\mathfrak{A}}$ in the structure $\mathfrak{A}$. Similar terminology is used for function symbols and constant symbols.

Example

Let $\mathbb{L}$ be the first-order language that has:

- $\dot{+}$ and $\dot{\times}$ (2-ary function symbols);
- $\dot{<}$ (a 2-ary predicate symbol);
- $\dot{0}$ and $\dot{1}$ (constant symbols), and
- $\dot{=}$.

Example

Let $\mathbb{L}$ be the first-order language that has:

- $\dot{+}$ and $\dot{\times}$ (2-ary function symbols);
- $\dot{<}$ (a 2-ary predicate symbol);
- $\dot{0}$ and $\dot{1}$ (constant symbols), and
- $\doteq$.

Let $\mathfrak{N}_1$ be the structure for $\mathbb{L}$ such that:

Example

Let $\mathbb{L}$ be the first-order language that has:

- $\dot{+}$ and $\dot{\times}$ (2-ary function symbols);
- $\dot{<}$ (a 2-ary predicate symbol);
- $\dot{0}$ and $\dot{1}$ (constant symbols), and
- $\dot{=}$.

Let $\mathfrak{N}_1$ be the structure for $\mathbb{L}$ such that:

- $|\mathfrak{N}_1| = \mathbb{N}$;

Example

Let $\mathbb{L}$ be the first-order language that has:

- $\dot{+}$ and $\dot{\times}$ (2-ary function symbols);
- $\dot{<}$ (a 2-ary predicate symbol);
- $\dot{0}$ and $\dot{1}$ (constant symbols), and
- $\doteq$.

Let $\mathfrak{N}_1$ be the structure for $\mathbb{L}$ such that:

- $|\mathfrak{N}_1| = \mathbb{N}$;
- $\dot{+}^{\mathfrak{N}_1} = +$(the addition function on $\mathbb{N}$);

Example

Let $\mathbb{L}$ be the first-order language that has:

- $\dot{+}$ and $\dot{\times}$ (2-ary function symbols);
- $\dot{<}$ (a 2-ary predicate symbol);
- $\dot{0}$ and $\dot{1}$ (constant symbols), and
- $\doteq$.

Let $\mathfrak{N}_1$ be the structure for $\mathbb{L}$ such that:

- $|\mathfrak{N}_1| = \mathbb{N}$;
- $\dot{+}^{\mathfrak{N}_1} = +$(the addition function on $\mathbb{N}$);
- $\dot{\times}^{\mathfrak{N}_1} = \times$(the multiplication function on $\mathbb{N}$);

Example

Let $\mathbb{L}$ be the first-order language that has:

- $\dot{+}$ and $\dot{\times}$ (2-ary function symbols);
- $\dot{<}$ (a 2-ary predicate symbol);
- $\dot{0}$ and $\dot{1}$ (constant symbols), and
- $\doteq$.

Let $\mathfrak{N}_1$ be the structure for $\mathbb{L}$ such that:

- $|\mathfrak{N}_1| = \mathbb{N}$;
- $\dot{+}^{\mathfrak{N}_1} = +$(the addition function on $\mathbb{N}$);
- $\dot{\times}^{\mathfrak{N}_1} = \times$(the multiplication function on $\mathbb{N}$);
- $\dot{<}^{\mathfrak{N}_1} = \{(a, b) \mid a, b \in \mathbb{N} \text{ and } a < b\}$;

Example

Let $\mathbb{L}$ be the first-order language that has:

- $\dot{+}$ and $\dot{\times}$ (2-ary function symbols);
- $\dot{<}$ (a 2-ary predicate symbol);
- $\dot{0}$ and $\dot{1}$ (constant symbols), and
- $\dot{=}$.

Let $\mathfrak{N}_1$ be the structure for $\mathbb{L}$ such that:

- $|\mathfrak{N}_1| = \mathbb{N}$;
- $\dot{+}^{\mathfrak{N}_1} = +$(the addition function on $\mathbb{N}$);
- $\dot{\times}^{\mathfrak{N}_1} = \times$(the multiplication function on $\mathbb{N}$);
- $\dot{<}^{\mathfrak{N}_1} = \{(a, b) \mid a, b \in \mathbb{N} \text{ and } a < b\}$;
- $\dot{0}^{\mathfrak{N}_1} = 0$;

Example

Let $\mathbb{L}$ be the first-order language that has:

- $\dot{+}$ and $\dot{\times}$ (2-ary function symbols);
- $\dot{<}$ (a 2-ary predicate symbol);
- $\dot{0}$ and $\dot{1}$ (constant symbols), and
- $\dot{=}$.

Let $\mathfrak{N}_1$ be the structure for $\mathbb{L}$ such that:

- $|\mathfrak{N}_1| = \mathbb{N}$;
- $\dot{+}^{\mathfrak{N}_1} = +$(the addition function on $\mathbb{N}$);
- $\dot{\times}^{\mathfrak{N}_1} = \times$(the multiplication function on $\mathbb{N}$);
- $\dot{<}^{\mathfrak{N}_1} = \{(a, b) \mid a, b \in \mathbb{N} \text{ and } a < b\}$;
- $\dot{0}^{\mathfrak{N}_1} = 0$;
- $\dot{1}^{\mathfrak{N}_1} = 1$.

Example

Let $\mathbb{L}$ be the first-order language that has:

- $\dot{+}$ and $\dot{\times}$ (2-ary function symbols);
- $\dot{<}$ (a 2-ary predicate symbol);
- $\dot{0}$ and $\dot{1}$ (constant symbols), and
- $\dot{=}$.

Let $\mathfrak{N}_1$ be the structure for $\mathbb{L}$ such that:

- $|\mathfrak{N}_1| = \mathbb{N}$;
- $\dot{+}^{\mathfrak{N}_1} = +$(the addition function on $\mathbb{N}$);
- $\dot{\times}^{\mathfrak{N}_1} = \times$(the multiplication function on $\mathbb{N}$);
- $\dot{<}^{\mathfrak{N}_1} = \{(a, b) \mid a, b \in \mathbb{N} \text{ and } a < b\}$;
- $\dot{0}^{\mathfrak{N}_1} = 0$;
- $\dot{1}^{\mathfrak{N}_1} = 1$.

We can describe this structure simply as $\mathfrak{N}_1 = \{\mathbb{N}, +, \times, 0, 1\}$.

Example (Cont'd)

Let $\mathfrak{N}_2$ be the structure for the same language:

Example (Cont'd)

Let $\mathfrak{N}_2$ be the structure for the same language:

- $|\mathfrak{N}_2| = \mathbb{N}$;

Example (Cont'd)

Let $\mathfrak{N}_2$ be the structure for the same language:

- $|\mathfrak{N}_2| = \mathbb{N}$;
- $\dot{+}^{\mathfrak{N}_2} = \times$;

Example (Cont'd)

Let $\mathfrak{N}_2$ be the structure for the same language:

- $|\mathfrak{N}_2| = \mathbb{N}$;
- $\dot{+}^{\mathfrak{N}_2} = \times$;
- $\dot{\times}^{\mathfrak{N}_2} = +$;

Example (Cont'd)

Let $\mathfrak{N}_2$ be the structure for the same language:

- $|\mathfrak{N}_2| = \mathbb{N}$;
- $\dot{+}^{\mathfrak{N}_2} = \times$;
- $\dot{\times}^{\mathfrak{N}_2} = +$;
- $\dot{<}^{\mathfrak{N}_2} = \{(a, b) \mid a, b \in \mathbb{N} \text{ and } a > b\}$;

Example (Cont'd)

Let $\mathfrak{N}_2$ be the structure for the same language:

- $|\mathfrak{N}_2| = \mathbb{N}$;
- $\dot{+}^{\mathfrak{N}_2} = \times$;
- $\dot{\times}^{\mathfrak{N}_2} = +$;
- $\dot{<}^{\mathfrak{N}_2} = \{(a, b) \mid a, b \in \mathbb{N} \text{ and } a > b\}$;
- $\dot{0}^{\mathfrak{N}_2} = 1$;

Example (Cont'd)

Let $\mathfrak{N}_2$ be the structure for the same language:

- $|\mathfrak{N}_2| = \mathbb{N}$;
- $\dot{+}^{\mathfrak{N}_2} = \times$;
- $\dot{\times}^{\mathfrak{N}_2} = +$;
- $\dot{<}^{\mathfrak{N}_2} = \{(a, b) \mid a, b \in \mathbb{N} \text{ and } a > b\}$;
- $\dot{0}^{\mathfrak{N}_2} = 1$;
- $\dot{1}^{\mathfrak{N}_2} = 0$.

Example (Cont'd)

Let $\mathfrak{N}$ be the structure for the same language:

Example (Cont'd)

Let $\mathfrak{R}$ be the structure for the same language:

- $|\mathfrak{R}| = \mathbb{R}$;

Example (Cont'd)

Let $\mathfrak{R}$ be the structure for the same language:

- $|\mathfrak{R}| = \mathbb{R}$;
- $\dot{+}^{\mathfrak{R}} = +$(addition on the real numbers);

Example (Cont'd)

Let $\mathfrak{R}$ be the structure for the same language:

- $|\mathfrak{R}| = \mathbb{R}$;
- $\dot{+}^{\mathfrak{R}} = +$(addition on the real numbers);
- $\dot{\times}^{\mathfrak{R}} = \times$(multiplication on the real numbers);

Example (Cont'd)

Let $\mathfrak{R}$ be the structure for the same language:

- $|\mathfrak{R}| = \mathbb{R}$;
- $\dot{+}^{\mathfrak{R}} = +$(addition on the real numbers);
- $\dot{\times}^{\mathfrak{R}} = \times$(multiplication on the real numbers);
- $\dot{<}^{\mathfrak{R}} = <$ (on the real numbers);

Example (Cont'd)

Let $\mathfrak{R}$ be the structure for the same language:

- $|\mathfrak{R}| = \mathbb{R}$;
- $\dot{+}^{\mathfrak{R}} = +$(addition on the real numbers);
- $\dot{\times}^{\mathfrak{R}} = \times$(multiplication on the real numbers);
- $\dot{<}^{\mathfrak{R}} =<$ (on the real numbers);
- $\dot{0}^{\mathfrak{R}} = 0$;

Example (Cont'd)

Let $\mathfrak{R}$ be the structure for the same language:

- $|\mathfrak{R}| = \mathbb{R}$;
- $\dot{+}^{\mathfrak{R}} = +$(addition on the real numbers);
- $\dot{\times}^{\mathfrak{R}} = \times$(multiplication on the real numbers);
- $\dot{<}^{\mathfrak{R}} =<$ (on the real numbers);
- $\dot{0}^{\mathfrak{R}} = 0$;
- $\dot{1}^{\mathfrak{R}} = 1$.

An Example of a Very Big Language

Let $\mathbb{L}$ have

- $\doteq$, and
- for each $r \in \mathbb{R}$ a constant symbol c_r.

An Example of a Very Big Language

Let $\mathbb{L}$ have

- $\doteq$, and
- for each $r \in \mathbb{R}$ a constant symbol c_r.

This language has uncountably many constant symbols.

An Example of a Very Big Language

Let $\mathbb{L}$ have

- $\dot{=}$, and
- for each $r \in \mathbb{R}$ a constant symbol c_r.

This language has uncountably many constant symbols.

A structure for this language is $\mathfrak{R}$, where

- $|\mathfrak{R}| = \mathbb{R}$, and
- ${c_r}^{\mathfrak{R}} = r$ for every $r \in \mathbb{R}$.

Special Status of $\doteq$

We have been very careful in distinguishing between things in the language $\mathbb{L}$ and things outside of $\mathbb{L}$.

Special Status of $\doteq$

We have been very careful in distinguishing between things in the language $\mathbb{L}$ and things outside of $\mathbb{L}$.

For example, $\doteq$ is a symbol in the language, while $=$ is not.

Special Status of $\doteq$

We have been very careful in distinguishing between things in the language $\mathbb{L}$ and things outside of $\mathbb{L}$.

For example, $\doteq$ is a symbol in the language, while $=$ is not.

Question

Why does Enderton no distinguish between the two?

Example: Directed Graph

Let $\mathbb{L}$ be the first-order language that (in addition to the symbols required in every first-order language) only has a 2-ary predicate symbol $\dot{E}$.

Example: Directed Graph

Let $\mathbb{L}$ be the first-order language that (in addition to the symbols required in every first-order language) only has a 2-ary predicate symbol $\dot{E}$.

Let $\mathfrak{B}$ be the structure for $\mathbb{L}$ such that:

- $|\mathfrak{B}| = \{a, b, c, d\}$;
- $\dot{E}^{\mathfrak{B}} = \{\langle a, b\rangle, \langle b, a\rangle, \langle b, c\rangle, \langle c, c\rangle\}$.

This denotes a directed graph (See Enderton, page 82)

Example: Directed Graph

Let $\mathbb{L}$ be the first-order language that (in addition to the symbols required in every first-order language) only has a 2-ary predicate symbol $\dot{E}$.

Let $\mathfrak{B}$ be the structure for $\mathbb{L}$ such that:

- $|\mathfrak{B}| = \{a, b, c, d\}$;
- $\dot{E}^{\mathfrak{B}} = \{\langle a, b\rangle, \langle b, a\rangle, \langle b, c\rangle, \langle c, c\rangle\}$.

This denotes a directed graph (See Enderton, page 82)

Example

The wff $\exists x \forall y, \neg \dot{E} yx$ denotes

> *There is a vertex x such that for any vertex y, no edge points from y to x.*

Example: Directed Graph

Let $\mathbb{L}$ be the first-order language that (in addition to the symbols required in every first-order language) only has a 2-ary predicate symbol $\dot{E}$.

Let $\mathfrak{B}$ be the structure for $\mathbb{L}$ such that:

- $|\mathfrak{B}| = \{a, b, c, d\}$;
- $\dot{E}^{\mathfrak{B}} = \{\langle a, b\rangle, \langle b, a\rangle, \langle b, c\rangle, \langle c, c\rangle\}$.

This denotes a directed graph (See Enderton, page 82)

Example

The wff $\exists x \forall y, \neg \dot{E} yx$ denotes

> *There is a vertex x such that for any vertex y, no edge points from y to x.*

Question

How do we show define $\exists x \forall y, \neg \dot{E} yx$ is true in $\mathfrak{B}$?

Given a formula φ and a structure $\mathfrak{A}$, how do we define "φ is true in $\mathfrak{A}$",
Or equally speaking, "$\mathfrak{A}$ satisfies φ"?

Assignment of Values to Terms

Let $\mathfrak{A}$ be a structure for the language $\mathbb{L}$. Let V be the set of variables, and T be the set of terms of $\mathbb{L}$.

Definition (Assignment Functions)

An assignment for $\mathfrak{A}$ is a function $s : V \to |\mathfrak{A}|$.

Assignment of Values to Terms

Let $\mathfrak{A}$ be a structure for the language $\mathbb{L}$. Let V be the set of variables, and T be the set of terms of $\mathbb{L}$.

Definition (Assignment Functions)

An assignment for $\mathfrak{A}$ is a function $s : V \to |\mathfrak{A}|$.

Definition (Assignment to Terms)

An assignment $s : V \to |\mathfrak{A}|$ is extended to a function $\bar{s} : T \to |\mathfrak{A}|$ as follows:

- $\bar{s}(v) = s(v)$ if v is a variable;
- $\bar{s}(c) = c^{\mathfrak{A}}$ if c is a constant symbol;
- $\bar{s}(ft_1 \dots t_n) = f^{\mathfrak{A}}(\bar{s}(t_1), \dots, \bar{s}(t_n))$ if f is an n-ary function symbol and $t_1, \dots, t_n$ are terms.

Example

Look the language $\mathbb{L}$ of the earlier example. Let s be an assignment function for the structure $\mathfrak{N}_1$ such that $s(v_3) = 5$. Then

Example

Look the language $\mathbb{L}$ of the earlier example. Let s be an assignment function for the structure $\mathfrak{N}_1$ such that $s(v_3) = 5$. Then

- $\overline{s}(\dot{+}\dot{\times}\dot{0}v_3\dot{1}) =$

Example

Look the language $\mathbb{L}$ of the earlier example. Let s be an assignment function for the structure $\mathfrak{N}_1$ such that $s(v_3) = 5$. Then

- $\overline{s}(\dot{+}\dot{\times}\dot{0}v_3\dot{1}) = 1$

Example

Look the language $\mathbb{L}$ of the earlier example. Let s be an assignment function for the structure $\mathfrak{N}_1$ such that $s(v_3) = 5$. Then

- $\overline{s}(\dot{+}\dot{\times}\dot{0}v_3\dot{1}) = 1$
- $\overline{s}(\dot{+}\dot{\times}\dot{1}\dot{1}v_3) =$

Example

Look the language $\mathbb{L}$ of the earlier example. Let s be an assignment function for the structure $\mathfrak{N}_1$ such that $s(v_3) = 5$. Then

- $\bar{s}(\dot{+}\dot{\times}\dot{0}v_3\dot{1}) = 1$
- $\bar{s}(\dot{+}\dot{\times}\dot{1}\dot{1}v_3) = 6$

Changing the Assignment Function

Let:

- s be an assignment function,
- x be a variable, and
- $a \in |\mathfrak{A}|$.

$s(x|a)$ is the new assignment, where for every variable y,

$$s(x|a)(y) = \begin{cases} s(y) & \text{if } y \neq x \\ a & \text{if } y = x \end{cases}$$

Example

If $y \neq x$ then

Example

If $y \neq x$ then

- $s(x|a)(y|b)(x) =$

Example

If $y \neq x$ then

- $s(x|a)(y|b)(x) = a$

Example

If $y \neq x$ then

- $s(x|a)(y|b)(x) = a$
- $s(x|a)(y|b)(y) =$

Example

If $y \neq x$ then

- $s(x|a)(y|b)(x) = a$
- $s(x|a)(y|b)(y) = b$

Example

If $y \neq x$ then

- $s(x|a)(y|b)(x) = a$
- $s(x|a)(y|b)(y) = b$
- $s(x|a)(x|b)(x) =$

Example

If $y \neq x$ then

- $s(x|a)(y|b)(x) = a$
- $s(x|a)(y|b)(y) = b$
- $s(x|a)(x|b)(x) = b$

Satisfaction in First-Order Logic

Given a first-order language $\mathbb{L}$:

- let $\mathfrak{A}$ be a structure for $\mathbb{L}$,
- let s be an assignment for $\mathfrak{A}$, and
- let φ be a wff in $\mathbb{L}$.

We shall talk about what it means for $\mathfrak{A}$ to satisfy φ with s, written as

$$\models_{\mathfrak{A}} \varphi[s]$$

Satisfaction in First-Order Logic

Given a first-order language $\mathbb{L}$:

- let $\mathfrak{A}$ be a structure for $\mathbb{L}$,
- let s be an assignment for $\mathfrak{A}$, and
- let φ be a wff in $\mathbb{L}$.

We shall talk about what it means for $\mathfrak{A}$ to satisfy φ with s, written as

$$\models_{\mathfrak{A}} \varphi[s]$$

Informally, it means:

> *The translation of φ determined by $\mathfrak{A}$, where a variable x is translated as $s(x)$, is true.*

Satisfaction for Atomic Formula

Definition

Let:

- $\mathfrak{A}$ be a structure for $\mathbb{L}$,
- s be an assignment for $\mathfrak{A}$, and
- $Pt_1 \ldots t_n$ be an atomic wff.

Then

Satisfaction for Atomic Formula

Definition

Let:

- $\mathfrak{A}$ be a structure for $\mathbb{L}$,
- s be an assignment for $\mathfrak{A}$, and
- $Pt_1 \ldots t_n$ be an atomic wff.

Then

- $\models_{\mathfrak{A}} Pt_1 \ldots t_n[s]$ iff $(\overline{s}(t_1), \ldots, \overline{s}(t_n)) \in P^{\mathfrak{A}}$ (when $P \neq \dot{=}$);

Satisfaction for Atomic Formula

Definition

Let:

- $\mathfrak{A}$ be a structure for $\mathbb{L}$,
- s be an assignment for $\mathfrak{A}$, and
- $Pt_1 \ldots t_n$ be an atomic wff.

Then

- $\vDash_{\mathfrak{A}} Pt_1 \ldots t_n[s]$ iff $(\overline{s}(t_1), \ldots, \overline{s}(t_n)) \in P^{\mathfrak{A}}$ (when $P \neq \doteq$);
- $\vDash_{\mathfrak{A}} \doteq t_1 t_2[s]$ iff $\overline{s}(t_1) = \overline{s}(t_2)$.

Satisfaction for Well-Formed Formula

Definition

- For an atomic formula, we have already given its definition;

Satisfaction for Well-Formed Formula

Definition

- For an atomic formula, we have already given its definition;
- Suppose $\models_{\mathfrak{A}} \alpha[s]$ and $\models_{\mathfrak{A}} \beta[s]$ have been defined. Then

Satisfaction for Well-Formed Formula

Definition

- For an atomic formula, we have already given its definition;
- Suppose $\models_{\mathfrak{A}} \alpha[s]$ and $\models_{\mathfrak{A}} \beta[s]$ have been defined. Then
  - $\models_{\mathfrak{A}} \neg\alpha[s]$ iff not $\models_{\mathfrak{A}} \alpha[s]$;

Satisfaction for Well-Formed Formula

Definition

- For an atomic formula, we have already given its definition;
- Suppose $\models_{\mathfrak{A}} \alpha[s]$ and $\models_{\mathfrak{A}} \beta[s]$ have been defined. Then
  - $\models_{\mathfrak{A}} \neg\alpha[s]$ iff not $\models_{\mathfrak{A}} \alpha[s]$;
  - $\models_{\mathfrak{A}} \alpha \to \beta[s]$ iff $\models_{\mathfrak{A}} \alpha[s] \Longrightarrow \models_{\mathfrak{A}} \beta[s]$;

Satisfaction for Well-Formed Formula

Definition

- For an atomic formula, we have already given its definition;
- Suppose $\models_{\mathfrak{A}} \alpha[s]$ and $\models_{\mathfrak{A}} \beta[s]$ have been defined. Then
  - $\models_{\mathfrak{A}} \neg\alpha[s]$ iff not $\models_{\mathfrak{A}} \alpha[s]$;
  - $\models_{\mathfrak{A}} \alpha \to \beta[s]$ iff $\models_{\mathfrak{A}} \alpha[s] \Longrightarrow \models_{\mathfrak{A}} \beta[s]$;
  - $\models_{\mathfrak{A}} \forall x\ \alpha[s]$ iff $\forall a \in |\mathfrak{A}|, \models_{\mathfrak{A}} \alpha[s(x|a)]$.

Satisfaction for Well-Formed Formula

Definition

- For an atomic formula, we have already given its definition;
- Suppose $\models_{\mathfrak{A}} \alpha[s]$ and $\models_{\mathfrak{A}} \beta[s]$ have been defined. Then
  - $\models_{\mathfrak{A}} \neg\alpha[s]$ iff not $\models_{\mathfrak{A}} \alpha[s]$;
  - $\models_{\mathfrak{A}} \alpha \to \beta[s]$ iff $\models_{\mathfrak{A}} \alpha[s] \Longrightarrow \models_{\mathfrak{A}} \beta[s]$;
  - $\models_{\mathfrak{A}} \forall x\, \alpha[s]$ iff $\forall a \in |\mathfrak{A}|, \models_{\mathfrak{A}} \alpha[s(x|a)]$.

If $\models_{\mathfrak{A}} \varphi[s]$, we say

- $\mathfrak{A}$ satisfies φ with s, or
- s satisfies φ in the structure $\mathfrak{A}$.

Example

Let $\mathfrak{N} = (\mathbb{N}, <, +, \times, 0, 1)$. This is our abbreviated way of saying:

Example

Let $\mathfrak{N} = (\mathbb{N}, <, +, \times, 0, 1)$. This is our abbreviated way of saying:

- $\mathbb{L}$ has a binary predicate symbol $\dot{<}$, 2-ary function symbols $\dot{+}$ and $\dot{\times}$, constant symbols $\dot{0}$ and $\dot{1}$, but no other predicate symbols (except for $\doteq$), functions symbols or constant symbols;

Example

Let $\mathfrak{N} = (\mathbb{N}, <, +, \times, 0, 1)$. This is our abbreviated way of saying:

- $\mathbb{L}$ has a binary predicate symbol $\dot{<}$, 2-ary function symbols $\dot{+}$ and $\dot{\times}$, constant symbols $\dot{0}$ and $\dot{1}$, but no other predicate symbols (except for $\dot{=}$), functions symbols or constant symbols;
- $\mathfrak{N}$ is the structure for $\mathbb{L}$:
  - whose universe is $\mathbb{N}$;
  - $\dot{<}^{\mathfrak{N}} = <$;
  - $\dot{+}^{\mathfrak{N}} = +$;
  - $\dot{\times}^{\mathfrak{N}} = \times$;
  - $\dot{0}^{\mathfrak{N}} = 0$ and $\dot{1}^{\mathfrak{N}} = 1$.

Example

Let $\mathfrak{N} = (\mathbb{N}, <, +, \times, 0, 1)$. This is our abbreviated way of saying:

- $\mathbb{L}$ has a binary predicate symbol $\dot{<}$, 2-ary function symbols $\dot{+}$ and $\dot{\times}$, constant symbols $\dot{0}$ and $\dot{1}$, but no other predicate symbols (except for $\doteq$), functions symbols or constant symbols;
- $\mathfrak{N}$ is the structure for $\mathbb{L}$:
  - whose universe is $\mathbb{N}$;
  - $\dot{<}^{\mathfrak{N}} =<$;
  - $\dot{+}^{\mathfrak{N}} = +$;
  - $\dot{\times}^{\mathfrak{N}} = \times$;
  - $\dot{0}^{\mathfrak{N}} = 0$ and $\dot{1}^{\mathfrak{N}} = 1$.

Similarly, let $\mathfrak{Z} = (\mathbb{Z}, <, +, \times, 0, 1)$. Note both $\mathfrak{N}$ and $\mathfrak{Z}$ are structures for the same language $\mathbb{L}$.

Example (Cont'd)

Question

Let φ be the wff

$$\forall x(\neg x \dot{<} \dot{0})$$

Which of the following judgments holds?

- For every $s : V \to \mathbb{N}$, $\models_{\mathfrak{N}} \varphi[s]$;
- For every $s : V \to \mathbb{N}$, $\models_{\mathfrak{Z}} \varphi[s]$.

More Examples

Let $\mathfrak{R} = (\mathbb{R}, <, +, \times, 0, 1)$.

Question

Let φ be the wff

$$\forall x \forall y (x \dot{<} y \to \exists z\, x \dot{<} z \land z \dot{<} y)$$

Then which of the following is true?

- For every $s : V \to \mathbb{Z}$, $\models_{\mathfrak{Z}} \varphi[s]$
- For every $s : V \to \mathbb{R}$, $\models_{\mathfrak{R}} \varphi[s]$

Soundness of Abbreviations

Recall the following abbreviations:

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \rightarrow \beta)$;

Soundness of Abbreviations

Recall the following abbreviations:

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;

Soundness of Abbreviations

Recall the following abbreviations:

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$;

Soundness of Abbreviations

Recall the following abbreviations:

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$;
- $\exists x\alpha$ abbreviates $(\neg\forall x(\neg\alpha))$;

Soundness of Abbreviations

Recall the following abbreviations:

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \rightarrow \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \rightarrow (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$;
- $\exists x\alpha$ abbreviates $(\neg\forall x(\neg\alpha))$;

For every structure $\mathfrak{A}$ and every $s : V \rightarrow |\mathfrak{A}|$, show the following:

Soundness of Abbreviations

Recall the following abbreviations:

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$;
- $\exists x\alpha$ abbreviates $(\neg\forall x(\neg\alpha))$;

For every structure $\mathfrak{A}$ and every $s : V \to |\mathfrak{A}|$, show the following:

- $\models_{\mathfrak{A}} (\alpha \vee \beta)[s]$ iff $\models_{\mathfrak{A}} \alpha[s]$ or $\models_{\mathfrak{A}} \beta[s]$;

Soundness of Abbreviations

Recall the following abbreviations:

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$;
- $\exists x\alpha$ abbreviates $(\neg\forall x(\neg\alpha))$;

For every structure $\mathfrak{A}$ and every $s : V \to |\mathfrak{A}|$, show the following:

- $\models_{\mathfrak{A}} (\alpha \vee \beta)[s]$ iff $\models_{\mathfrak{A}} \alpha[s]$ or $\models_{\mathfrak{A}} \beta[s]$;
- $\models_{\mathfrak{A}} (\alpha \wedge \beta)[s]$ iff $\models_{\mathfrak{A}} \alpha[s]$ and $\models_{\mathfrak{A}} \beta[s]$;

Soundness of Abbreviations

Recall the following abbreviations:

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$;
- $\exists x\alpha$ abbreviates $(\neg\forall x(\neg\alpha))$;

For every structure $\mathfrak{A}$ and every $s : V \to |\mathfrak{A}|$, show the following:

- $\models_{\mathfrak{A}} (\alpha \vee \beta)[s]$ iff $\models_{\mathfrak{A}} \alpha[s]$ or $\models_{\mathfrak{A}} \beta[s]$;
- $\models_{\mathfrak{A}} (\alpha \wedge \beta)[s]$ iff $\models_{\mathfrak{A}} \alpha[s]$ and $\models_{\mathfrak{A}} \beta[s]$;
- $\models_{\mathfrak{A}} (\alpha \leftrightarrow \beta)[s]$ iff $\models_{\mathfrak{A}} \alpha[s] \iff \models_{\mathfrak{A}} \beta[s]$;

Soundness of Abbreviations

Recall the following abbreviations:

- $(\alpha \vee \beta)$ abbreviates $((\neg\alpha) \to \beta)$;
- $(\alpha \wedge \beta)$ abbreviates $(\neg(\alpha \to (\neg\beta)))$;
- $(\alpha \leftrightarrow \beta)$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$;
- $\exists x\alpha$ abbreviates $(\neg\forall x(\neg\alpha))$;

For every structure $\mathfrak{A}$ and every $s : V \to |\mathfrak{A}|$, show the following:

- $\models_{\mathfrak{A}} (\alpha \vee \beta)[s]$ iff $\models_{\mathfrak{A}} \alpha[s]$ or $\models_{\mathfrak{A}} \beta[s]$;
- $\models_{\mathfrak{A}} (\alpha \wedge \beta)[s]$ iff $\models_{\mathfrak{A}} \alpha[s]$ and $\models_{\mathfrak{A}} \beta[s]$;
- $\models_{\mathfrak{A}} (\alpha \leftrightarrow \beta)[s]$ iff $\models_{\mathfrak{A}} \alpha[s] \iff \models_{\mathfrak{A}} \beta[s]$;
- $\models_{\mathfrak{A}} \exists x\alpha[s]$ iff $\exists a \in |\mathfrak{A}|, \models_{\mathfrak{A}} \alpha[s(x|a)]$

Example: Directed Graph

Let $\mathbb{L}$ be the first-order language that (in addition to the symbols required in every first-order language) only has a 2-ary predicate symbol $\dot{E}$.

Let $\mathfrak{B}$ be the structure for $\mathbb{L}$ such that:

- $|\mathfrak{B}| = \{a, b, c, d\}$;
- $\dot{E}^{\mathfrak{B}} = \{\langle a, b\rangle, \langle b, a\rangle, \langle b, c\rangle, \langle c, c\rangle\}$.

Example: Directed Graph

Let $\mathbb{L}$ be the first-order language that (in addition to the symbols required in every first-order language) only has a 2-ary predicate symbol $\dot{E}$.

Let $\mathfrak{B}$ be the structure for $\mathbb{L}$ such that:

- $|\mathfrak{B}| = \{a, b, c, d\}$;
- $\dot{E}^{\mathfrak{B}} = \{\langle a, b\rangle, \langle b, a\rangle, \langle b, c\rangle, \langle c, c\rangle\}$.

Question

Let $\sigma = \exists x \forall y, \neg \dot{E} yx$. For every assignment $s : V \to |\mathfrak{B}|$, does $\models_{\mathfrak{B}} \sigma[s]$ hold?

More Examples

Let

- φ_1 be $\forall x(\neg x \dot{<} y)$, and
- φ_2 be $\forall x(\neg x \dot{<} \dot{0})$.

Then

More Examples

Let

- φ_1 be $\forall x(\neg x \dot{<} y)$, and
- φ_2 be $\forall x(\neg x \dot{<} \dot{0})$.

Then

(1) $\models_{\mathfrak{N}} \varphi_1[s]$ iff

More Examples

Let

- φ_1 be $\forall x(\neg x \dot{<} y)$, and
- φ_2 be $\forall x(\neg x \dot{<} \dot{0})$.

Then

(1) $\models_{\mathfrak{N}} \varphi_1[s]$ iff $\forall a \in \mathbb{N}, s(y) \leq a$;

More Examples

Let

- φ_1 be $\forall x(\neg x \dot{<} y)$, and
- φ_2 be $\forall x(\neg x \dot{<} \dot{0})$.

Then

(1) $\models_{\mathfrak{N}} \varphi_1[s]$ iff $\forall a \in \mathbb{N}, s(y) \leq a$;

(2) $\models_{\mathfrak{N}} \varphi_2[s]$ iff

More Examples

Let

- φ_1 be $\forall x(\neg x \dot{<} y)$, and
- φ_2 be $\forall x(\neg x \dot{<} \dot{0})$.

Then

(1) $\models_{\mathfrak{N}} \varphi_1[s]$ iff $\forall a \in \mathbb{N}, s(y) \leq a$;

(2) $\models_{\mathfrak{N}} \varphi_2[s]$ iff $\forall a \in \mathbb{N}, 0 \leq a$;

More Examples

Let

- φ_1 be $\forall x(\neg x \dot{<} y)$, and
- φ_2 be $\forall x(\neg x \dot{<} \dot{0})$.

Then

(1) $\models_{\mathfrak{N}} \varphi_1[s]$ iff $\forall a \in \mathbb{N}, s(y) \leq a$;

(2) $\models_{\mathfrak{N}} \varphi_2[s]$ iff $\forall a \in \mathbb{N}, 0 \leq a$;

Note that

- (1) is true iff $s(y) = 0$, so whether it is true or not depend on s, whereas
- (2) is true for *all* s.

More Examples

Let

- φ_1 be $\forall x(\neg x \dot{<} y)$, and
- φ_2 be $\forall x(\neg x \dot{<} \dot{0})$.

Then

(1) $\vDash_{\mathfrak{N}} \varphi_1[s]$ iff $\forall a \in \mathbb{N}, s(y) \leq a$;

(2) $\vDash_{\mathfrak{N}} \varphi_2[s]$ iff $\forall a \in \mathbb{N}, 0 \leq a$;

Note that

- (1) is true iff $s(y) = 0$, so whether it is true or not depend on s, whereas
- (2) is true for *all* s.

What is the difference between φ_1 and φ_2 account for this?

Let's go back to talk about an important syntactic concept: *Free Occurrences of Variables*

Free Occurrence of Variables

Definition

- The variable x occurs free in an atomic wff φ iff it occurs in φ;

Free Occurrence of Variables

Definition

- The variable x occurs free in an atomic wff φ iff it occurs in φ;
- x occurs free in $\neg\alpha$ iff x occurs free in α;

Free Occurrence of Variables

Definition

- The variable x occurs free in an atomic wff φ iff it occurs in φ;
- x occurs free in $\neg\alpha$ iff x occurs free in α;
- x occurs free in $\alpha \to \beta$ iff x occurs free in α or in β;

Free Occurrence of Variables

Definition

- The variable x occurs free in an atomic wff φ iff it occurs in φ;
- x occurs free in $\neg\alpha$ iff x occurs free in α;
- x occurs free in $\alpha \to \beta$ iff x occurs free in α or in β;
- x occurs free in $\forall y\ \alpha$ iff x occurs free in α and $x \neq y$.

Sentences

Definition (Sentences)

φ is a sentence iff no variable occurs free in φ.

Sentences

Definition (Sentences)

φ is a sentence iff no variable occurs free in φ.

Remark

We often use σ or τ to stand for sentences.

Examples

Question

Which variables occur free in the following?

Examples

Question

Which variables occur free in the following?

- $\dot{0} \mathbin{\dot{<}} \dot{1}$

Examples

Question

Which variables occur free in the following?

- $\dot{0}\mathrel{\dot{<}}\dot{1}$

  None, so this is a sentence.

Examples

Question

Which variables occur free in the following?

- $\dot{0} \dot{<} \dot{1}$

  None, so this is a sentence.

- $\forall x(\neg x \dot{<} y)$

Examples

Question

Which variables occur free in the following?

- $\dot{0}\dot{<}\dot{1}$

  None, so this is a sentence.

- $\forall x(\neg x\dot{<}y)$

  y occurs free, but x *does not*.

Examples

Question

Which variables occur free in the following?

- $\dot{0}\dot{<}\dot{1}$

  None, so this is a sentence.

- $\forall x(\neg x\dot{<}y)$

  y occurs free, but x *does not*.

- $\forall x(\neg x\dot{<}\dot{0})$

Examples

Question

Which variables occur free in the following?

- $\dot{0} \dot{<} \dot{1}$
  None, so this is a sentence.
- $\forall x (\neg x \dot{<} y)$
  y occurs free, but x *does not*.
- $\forall x (\neg x \dot{<} \dot{0})$
  No variable occurs free, so this is a sentence.

Examples

Question

Which variables occur free in the following?

- $\dot{0} \dot{<} \dot{1}$
  None, so this is a sentence.
- $\forall x(\neg x \dot{<} y)$
  y occurs free, but x *does not*.
- $\forall x(\neg x \dot{<} \dot{0})$
  No variable occurs free, so this is a sentence.
- $\forall x \forall y(x \dot{<} y \to \exists z\, x \dot{<} z \wedge z \dot{<} y)$

Examples

Question

Which variables occur free in the following?

- $\dot{0} \dot{<} \dot{1}$
  None, so this is a sentence.
- $\forall x(\neg x \dot{<} y)$
  y occurs free, but x *does not*.
- $\forall x(\neg x \dot{<} \dot{0})$
  No variable occurs free, so this is a sentence.
- $\forall x \forall y(x \dot{<} y \to \exists z\, x \dot{<} z \wedge z \dot{<} y)$
  None, so this is a sentence.

How do free occurrences of variables affect satisfiability?

Satisfaction Depends Only on Variables that Occur Free

Theorem

Let $\mathfrak{A}$ be a structure for $\mathbb{L}$, s_1 and s_2 be two assignment for $\mathfrak{A}$ and φ be a wff of $\mathbb{L}$.
If $s_1(x) = s_2(x)$ for every x that occurs free in φ, then

$$\models_{\mathfrak{A}} \varphi[s_1] \iff \models_{\mathfrak{A}} \varphi[s_2]$$

Satisfaction Depends Only on Variables that Occur Free

Theorem

Let $\mathfrak{A}$ be a structure for $\mathbb{L}$, s_1 and s_2 be two assignment for $\mathfrak{A}$ and φ be a wff of $\mathbb{L}$.
If $s_1(x) = s_2(x)$ for every x that occurs free in φ, then

$$\models_{\mathfrak{A}} \varphi[s_1] \Longleftrightarrow \models_{\mathfrak{A}} \varphi[s_2]$$

Lemma

Let $\mathfrak{A}$ be a structure for $\mathbb{L}$, s_1 and s_2 be two assignment for $\mathfrak{A}$ and t be a term of $\mathbb{L}$.
If $s_1(x) = s_2(x)$ for every x that occurs in t, then

$$\overline{s_1}(t) = \overline{s_2}(t)$$

Abbreviations

Definition

Let φ be a wff such that all variables occurring free in φ are included among $v_1, \ldots, v_k$. Given $a_1, \ldots, a_k \in |\mathfrak{A}|$,

$$\models_{\mathfrak{A}} \varphi[\![a_1, \ldots, a_k]\!]$$

means $\models_{\mathfrak{A}} \varphi[s]$ for some $s : V \to |\mathfrak{A}|$ such that $s(v_i) = a_i (1 \leq i \leq k)$.

Abbreviations

Definition

Let φ be a wff such that all variables occurring free in φ are included among $v_1, \ldots, v_k$. Given $a_1, \ldots, a_k \in |\mathfrak{A}|$,

$$\models_{\mathfrak{A}} \varphi[\![a_1, \ldots, a_k]\!]$$

means $\models_{\mathfrak{A}} \varphi[s]$ for some $s : V \to |\mathfrak{A}|$ such that $s(v_i) = a_i (1 \leq i \leq k)$.

Example

Let $\mathfrak{N} = (\mathbb{N}, <, +, \times, 0, 1)$. We have

Abbreviations

Definition

Let φ be a wff such that all variables occurring free in φ are included among $v_1, \ldots, v_k$. Given $a_1, \ldots, a_k \in |\mathfrak{A}|$,

$$\models_{\mathfrak{A}} \varphi[\![a_1, \ldots, a_k]\!]$$

means $\models_{\mathfrak{A}} \varphi[s]$ for some $s : V \to |\mathfrak{A}|$ such that $s(v_i) = a_i (1 \leq i \leq k)$.

Example

Let $\mathfrak{N} = (\mathbb{N}, <, +, \times, 0, 1)$. We have

- $\models_{\mathfrak{N}} \forall v_2, (\neg v_2 \dot{<} v_1)[\![0]\!]$;

Abbreviations

Definition

Let φ be a wff such that all variables occurring free in φ are included among $v_1, \ldots, v_k$. Given $a_1, \ldots, a_k \in |\mathfrak{A}|$,

$$\models_{\mathfrak{A}} \varphi[\![a_1, \ldots, a_k]\!]$$

means $\models_{\mathfrak{A}} \varphi[s]$ for some $s : V \to |\mathfrak{A}|$ such that $s(v_i) = a_i (1 \leq i \leq k)$.

Example

Let $\mathfrak{N} = (\mathbb{N}, <, +, \times, 0, 1)$. We have

- $\models_{\mathfrak{N}} \forall v_2, (\neg v_2 \dot{<} v_1)[\![0]\!]$;
- $\not\models_{\mathfrak{N}} \forall v_2, (\neg v_2 \dot{<} v_1)[\![2]\!]$.

Satisfaction for Sentences

Corollary

If σ is a sentence then either:

(1) $\models_{\mathfrak{A}} \sigma[s]$ for every assignment s, or

(2) $\not\models_{\mathfrak{A}} \sigma[s]$ for every assignment s.

In case (1), we say σ is true in $\mathfrak{A}$, and in case (2), we say σ is false in $\mathfrak{A}$.

Satisfaction for Sentences

Corollary

If σ is a sentence then either:

(1) $\models_{\mathfrak{A}} \sigma[s]$ for every assignment s, or

(2) $\not\models_{\mathfrak{A}} \sigma[s]$ for every assignment s.

In case (1), we say σ is true in $\mathfrak{A}$, and in case (2), we say σ is false in $\mathfrak{A}$.

Thus if σ is a sentence then whether or note $\models_{\mathfrak{A}} \sigma[s]$ does not depend on s. So we can just write $\models_{\mathfrak{A}} \sigma$ or $\not\models_{\mathfrak{A}} \sigma$.

Earlier Example

Let σ be the sentence $\forall x \forall y (x \dot{<} y \to \exists z\, x \dot{<} z \land z \dot{<} y)$.
Then σ is true in $\mathfrak{R}$ but false in $\mathfrak{Z}$.

Sentences that Distinguish Between Structures

Let:

- $\mathfrak{N} = (\mathbb{N}, <)$;
- $\mathfrak{Z} = (\mathbb{Z}, <)$;
- $\mathfrak{Q} = (\mathbb{Q}, <)$;
- $\mathfrak{R} = (\mathbb{R}, <)$.

Sentences that Distinguish Between Structures

Let:

- $\mathfrak{N} = (\mathbb{N}, <)$;
- $\mathfrak{Z} = (\mathbb{Z}, <)$;
- $\mathfrak{Q} = (\mathbb{Q}, <)$;
- $\mathfrak{R} = (\mathbb{R}, <)$.

Question

For each pair of these structures, can you find a sentence in this language that is true in one and false in the other?

Elementary Equivalence

Definition

Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures for the same language $\mathbb{L}$. $\mathfrak{A}$ and $\mathfrak{B}$ are elementarily equivalent (written $\mathfrak{A} \equiv \mathfrak{B}$) if for every sentence σ of $\mathbb{L}$

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

Elementary Equivalence

Definition

Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures for the same language $\mathbb{L}$. $\mathfrak{A}$ and $\mathfrak{B}$ are elementarily equivalent (written $\mathfrak{A} \equiv \mathfrak{B}$) if for every sentence σ of $\mathbb{L}$

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

Remark

We have just seen that:

- $\mathfrak{N} \not\equiv \mathfrak{Z}$;
- $\mathfrak{Z} \not\equiv \mathfrak{Q}$;
- $\mathfrak{Z} \not\equiv \mathfrak{R}$.

Comparing $\mathfrak{Q}$ and $\mathfrak{R}$

Question

Is is true that $\mathfrak{Q}$ and $\mathfrak{R}$ are elementarily equivalent?

Comparing $\mathfrak{Q}$ and $\mathfrak{R}$

Question

Is is true that $\mathfrak{Q}$ and $\mathfrak{R}$ are elementarily equivalent?

Answer

Perhaps the answer is not so easy!

Models of Sentences

Definition

- $\mathfrak{A}$ is a model of the sentence σ if $\models_{\mathfrak{A}} \sigma$, i.e., if σ is true in $\mathfrak{A}$;

Models of Sentences

Definition

- $\mathfrak{A}$ is a model of the sentence σ if $\models_{\mathfrak{A}} \sigma$, i.e., if σ is true in $\mathfrak{A}$;
- $\mathfrak{A}$ is a model of a set Σ of sentences if $\mathfrak{A}$ is a model of every member of Σ, i.e., every sentence in Σ is true in $\mathfrak{A}$.

Models of Sentences

Definition

- $\mathfrak{A}$ is a model of the sentence σ if $\models_{\mathfrak{A}} \sigma$, i.e., if σ is true in $\mathfrak{A}$;
- $\mathfrak{A}$ is a model of a set Σ of sentences if $\mathfrak{A}$ is a model of every member of Σ, i.e., every sentence in Σ is true in $\mathfrak{A}$.

Question

Let $\mathfrak{R} = (\mathbb{R}, <, +, \times, 0, 1)$ and $\mathfrak{Q} = (\mathbb{Q}, <, +, \times, 0, 1)$. Is there a sentence that is true in $\mathfrak{R}$, but not in $\mathfrak{Q}$?

Models of Sentences

Definition

- $\mathfrak{A}$ is a model of the sentence σ if $\models_{\mathfrak{A}} \sigma$, i.e., if σ is true in $\mathfrak{A}$;
- $\mathfrak{A}$ is a model of a set Σ of sentences if $\mathfrak{A}$ is a model of every member of Σ, i.e., every sentence in Σ is true in $\mathfrak{A}$.

Question

Let $\mathfrak{R} = (\mathbb{R}, <, +, \times, 0, 1)$ and $\mathfrak{Q} = (\mathbb{Q}, <, +, \times, 0, 1)$. Is there a sentence that is true in $\mathfrak{R}$, but not in $\mathfrak{Q}$?

Answer

Yes. Let σ be $\exists x\; x \dot{\times} x \doteq \dot{1} \dot{+} \dot{1}$.

Example

Let $\mathbb{L}$ be a first-order language with 2-ary predicate symbols $\dot{P}$ and $\doteq$. Given a structure for $\mathbb{L}$, we have:

Example

Let $\mathbb{L}$ be a first-order language with 2-ary predicate symbols $\dot{P}$ and $\doteq$. Given a structure for $\mathbb{L}$, we have:

- $\mathfrak{A}$ is a model of $\forall x \forall y\ x \doteq y$ iff

Example

Let $\mathbb{L}$ be a first-order language with 2-ary predicate symbols $\dot{P}$ and $\doteq$.
Given a structure for $\mathbb{L}$, we have:

- $\mathfrak{A}$ is a model of $\forall x \forall y\ x \doteq y$ iff $|\mathfrak{A}|$ contains exactly one element;

Example

Let $\mathbb{L}$ be a first-order language with 2-ary predicate symbols $\dot{P}$ and $\doteq$.
Given a structure for $\mathbb{L}$, we have:

- $\mathfrak{A}$ is a model of $\forall x \forall y\ x \doteq y$ iff $|\mathfrak{A}|$ contains exactly one element;
- $\mathfrak{A}$ is a model of $\forall x \forall y\ \dot{P}xy$ iff

Example

Let $\mathbb{L}$ be a first-order language with 2-ary predicate symbols $\dot{P}$ and $\doteq$.
Given a structure for $\mathbb{L}$, we have:

- $\mathfrak{A}$ is a model of $\forall x \forall y \; x \doteq y$ iff $|\mathfrak{A}|$ contains exactly one element;
- $\mathfrak{A}$ is a model of $\forall x \forall y \; \dot{P}xy$ iff $\dot{P}^{\mathfrak{A}} = |\mathfrak{A}| \times |\mathfrak{A}|$;

Example

Let $\mathbb{L}$ be a first-order language with 2-ary predicate symbols $\dot{P}$ and $\doteq$. Given a structure for $\mathbb{L}$, we have:

- $\mathfrak{A}$ is a model of $\forall x \forall y \; x \doteq y$ iff $|\mathfrak{A}|$ contains exactly one element;
- $\mathfrak{A}$ is a model of $\forall x \forall y \; \dot{P}xy$ iff $\dot{P}^{\mathfrak{A}} = |\mathfrak{A}| \times |\mathfrak{A}|$;
- $\mathfrak{A}$ is a model of $\forall x \forall y \; \neg \dot{P}xy$ iff

Example

Let $\mathbb{L}$ be a first-order language with 2-ary predicate symbols $\dot{P}$ and $\dot{=}$. Given a structure for $\mathbb{L}$, we have:

- $\mathfrak{A}$ is a model of $\forall x \forall y\ x \dot{=} y$ iff $|\mathfrak{A}|$ contains exactly one element;
- $\mathfrak{A}$ is a model of $\forall x \forall y\ \dot{P}xy$ iff $\dot{P}^{\mathfrak{A}} = |\mathfrak{A}| \times |\mathfrak{A}|$;
- $\mathfrak{A}$ is a model of $\forall x \forall y\ \neg \dot{P}xy$ iff $\dot{P}^{\mathfrak{A}} = \emptyset$;

Example

Let $\mathbb{L}$ be a first-order language with 2-ary predicate symbols $\dot{P}$ and $\doteq$. Given a structure for $\mathbb{L}$, we have:

- $\mathfrak{A}$ is a model of $\forall x \forall y \; x \doteq y$ iff $|\mathfrak{A}|$ contains exactly one element;
- $\mathfrak{A}$ is a model of $\forall x \forall y \; \dot{P}xy$ iff $\dot{P}^{\mathfrak{A}} = |\mathfrak{A}| \times |\mathfrak{A}|$;
- $\mathfrak{A}$ is a model of $\forall x \forall y \; \neg \dot{P}xy$ iff $\dot{P}^{\mathfrak{A}} = \emptyset$;
- $\mathfrak{A}$ is a model of $\forall x \exists y \; \dot{P}xy$ iff

Example

Let $\mathbb{L}$ be a first-order language with 2-ary predicate symbols $\dot{P}$ and $\doteq$. Given a structure for $\mathbb{L}$, we have:

- $\mathfrak{A}$ is a model of $\forall x \forall y\ x \doteq y$ iff $|\mathfrak{A}|$ contains exactly one element;
- $\mathfrak{A}$ is a model of $\forall x \forall y\ \dot{P}xy$ iff $\dot{P}^{\mathfrak{A}} = |\mathfrak{A}| \times |\mathfrak{A}|$;
- $\mathfrak{A}$ is a model of $\forall x \forall y\ \neg \dot{P}xy$ iff $\dot{P}^{\mathfrak{A}} = \emptyset$;
- $\mathfrak{A}$ is a model of $\forall x \exists y\ \dot{P}xy$ iff the domain of $\dot{P}^{\mathfrak{A}}$ is $|\mathfrak{A}|$.

We notice that a sentence may denote a class of structures (i.e., its models).

We notice that a sentence may denote a class of structures (i.e., its models).

We shall devle into this point in detail. However, let us first revisit some basic concepts in set theory.

Reflexivity, Symmetry and Transitivity

Definition

Let R be a binary relation.

Reflexivity, Symmetry and Transitivity

Definition

Let R be a binary relation.

- R is symmetric if for every a and b,

$$(a, b) \in R \implies (b, a) \in R$$

Reflexivity, Symmetry and Transitivity

Definition

Let R be a binary relation.

- R is symmetric if for every a and b,

$$(a, b) \in R \Longrightarrow (b, a) \in R$$

- R is transitive if for every a, b, and c,

$$(a, b) \in R \Longrightarrow (b, c) \in R \Longrightarrow (a, c) \in R$$

Reflexivity, Symmetry and Transitivity

Definition

Let R be a binary relation.

- R is symmetric if for every a and b,

$$(a, b) \in R \implies (b, a) \in R$$

- R is transitive if for every a, b, and c,

$$(a, b) \in R \implies (b, c) \in R \implies (a, c) \in R$$

- R is reflexive on the set A if for all $a \in A$,

$$(a, a) \in R$$

Reflexivity, Symmetry and Transitivity

Definition

Let R be a binary relation.

- R is symmetric if for every a and b,

$$(a, b) \in R \implies (b, a) \in R$$

- R is transitive if for every a, b, and c,

$$(a, b) \in R \implies (b, c) \in R \implies (a, c) \in R$$

- R is reflexive on the set A if for all $a \in A$,

$$(a, a) \in R$$

- R satisfies trichotomy on A if for all $a, b, c \in A$, exactly one of the following is true:

$$(a, b) \in R, \qquad (b, a) \in R, \qquad a = b$$

Linear Ordering

Definition

A binary relation R is a linear ordering on A if R is transitive and satisfies trichotomy on A.

Linear Ordering

Definition

A binary relation R is a linear ordering on A if R is transitive and satisfies trichotomy on A.

Definition

Let $\mathbb{L}$ be the language with a binary relation symbol $\dot{R}$ and $\dot{=}$ (and no other symbols). Let $\mathfrak{A} = (A, R)$, i.e., $(A = |\mathfrak{A}|$ and $R = \dot{R}^{\mathfrak{A}})$.

Linear Ordering

Definition

A binary relation R is a linear ordering on A if R is transitive and satisfies trichotomy on A.

Definition

Let $\mathbb{L}$ be the language with a binary relation symbol $\dot{R}$ and $\doteq$ (and no other symbols). Let $\mathfrak{A} = (A, R)$, i.e., $(A = |\mathfrak{A}|$ and $R = \dot{R}^{\mathfrak{A}})$.

- $\mathfrak{A}$ is *transitive* if R is transitive;

Linear Ordering

Definition

A binary relation R is a linear ordering on A if R is transitive and satisfies trichotomy on A.

Definition

Let $\mathbb{L}$ be the language with a binary relation symbol $\dot{R}$ and $\doteq$ (and no other symbols). Let $\mathfrak{A} = (A, R)$, i.e., ($A = |\mathfrak{A}|$ and $R = \dot{R}^{\mathfrak{A}}$).

- $\mathfrak{A}$ is *transitive* if R is transitive;
- $\mathfrak{A}$ is a *linearly ordered* structure if R is a linear ordering on A.

Linear Ordering

Definition

A binary relation R is a linear ordering on A if R is transitive and satisfies trichotomy on A.

Definition

Let $\mathbb{L}$ be the language with a binary relation symbol $\dot{R}$ and $\doteq$ (and no other symbols). Let $\mathfrak{A} = (A, R)$, i.e., ($A = |\mathfrak{A}|$ and $R = \dot{R}^{\mathfrak{A}}$).

- $\mathfrak{A}$ is *transitive* if R is transitive;
- $\mathfrak{A}$ is a *linearly ordered* structure if R is a linear ordering on A.

See the discussion on Page 93 of Enderton's.

Examples

Each of the following is a linearly ordered structure:

- $(\mathbb{N}, <)$;
- $(\mathbb{Z}, <)$;
- $(\mathbb{R}, <)$.

Examples

Each of the following is a linearly ordered structure:

- $(\mathbb{N}, <)$;
- $(\mathbb{Z}, <)$;
- $(\mathbb{R}, <)$.

Also, each of the following is linearly ordered structure:

- $(\mathbb{N}, >)$;
- $(\mathbb{Z}, >)$;
- $(\mathbb{R}, >)$.

Examples

Each of the following is a linearly ordered structure:

- $(\mathbb{N}, <)$;
- $(\mathbb{Z}, <)$;
- $(\mathbb{R}, <)$.

Also, each of the following is linearly ordered structure:

- $(\mathbb{N}, >)$;
- $(\mathbb{Z}, >)$;
- $(\mathbb{R}, >)$.

Question

Is $(\mathbb{N}, \leq)$ a linearly ordered structure?

What Can Sentences Say About Structures

Question

Let $\mathfrak{A} = (A, R)$.

What Can Sentences Say About Structures

Question

Let $\mathfrak{A} = (A, R)$.

- $\mathfrak{A}$ is transitive iff $\models_{\mathfrak{A}} \sigma$, where $\sigma = ?$;

What Can Sentences Say About Structures

Question

Let $\mathfrak{A} = (A, R)$.

- $\mathfrak{A}$ is transitive iff $\models_{\mathfrak{A}} \sigma$, where $\sigma = ?$;
- $\mathfrak{A}$ is linearly ordered iff $\models_{\mathfrak{A}} \sigma$, where $\sigma = ?$;

What Can Sentences Say About Structures

Question

Let $\mathfrak{A} = (A, R)$.

- $\mathfrak{A}$ is transitive iff $\models_{\mathfrak{A}} \sigma$, where $\sigma = ?$;
- $\mathfrak{A}$ is linearly ordered iff $\models_{\mathfrak{A}} \sigma$, where $\sigma = ?$;
- $dom(R) = A$ iff $\models_{\mathfrak{A}} \sigma$, where $\sigma = ?$;

What Can Sentences Say About Structures

Question

Let $\mathfrak{A} = (A, R)$.

- $\mathfrak{A}$ is transitive iff $\models_{\mathfrak{A}} \sigma$, where $\sigma =?$;
- $\mathfrak{A}$ is linearly ordered iff $\models_{\mathfrak{A}} \sigma$, where $\sigma =?$;
- $dom(R) = A$ iff $\models_{\mathfrak{A}} \sigma$, where $\sigma =?$;
- $rng(R) = A$ iff $\models_{\mathfrak{A}} \sigma$, where $\sigma =?$;

What Can Sentences Say About Structures

Question

Let $\mathfrak{A} = (A, R)$.

- $\mathfrak{A}$ is transitive iff $\models_{\mathfrak{A}} \sigma$, where $\sigma = ?$;
- $\mathfrak{A}$ is linearly ordered iff $\models_{\mathfrak{A}} \sigma$, where $\sigma = ?$;
- $dom(R) = A$ iff $\models_{\mathfrak{A}} \sigma$, where $\sigma = ?$;
- $rng(R) = A$ iff $\models_{\mathfrak{A}} \sigma$, where $\sigma = ?$;
- R is a function iff $\models_{\mathfrak{A}} \sigma$, where $\sigma = ?$.

What Can Sentences Say About Structures

Question

Let $\mathfrak{A} = (A, R)$.

- $\mathfrak{A}$ is transitive iff $\models_{\mathfrak{A}} \sigma$, where $\sigma =?$;
- $\mathfrak{A}$ is linearly ordered iff $\models_{\mathfrak{A}} \sigma$, where $\sigma =?$;
- $dom(R) = A$ iff $\models_{\mathfrak{A}} \sigma$, where $\sigma =?$;
- $rng(R) = A$ iff $\models_{\mathfrak{A}} \sigma$, where $\sigma =?$;
- R is a function iff $\models_{\mathfrak{A}} \sigma$, where $\sigma =?$.

See the discussion on Page 93 of Enderton's for some of the answers.

Models of a Single Sentence

Definition

A set of structures $\mathcal{K}$ is an elementary class (EC) if there is a sentence σ such that

$$\mathcal{K} = \{\mathfrak{A} \mid \mathfrak{A} \text{ is a model of } \sigma\},$$

i.e.,

$$\mathcal{K} = \{\mathfrak{A} \mid \models_{\mathfrak{A}} \sigma\}.$$

Example

Question

Let $\mathbb{L}$ be the language with a binary predicate symbol $\dot{E}$ and $\dot{=}$, but no other symbols. A structure $\mathfrak{G} = (G, E)$ for $\mathbb{L}$ (where $G = |\mathfrak{G}|$ and $E = \dot{E}^{\mathfrak{G}}$ is a *graph* if

- E is symmetric, and
- for every $a \in G$, $(a, a) \notin E$

Is the set of graphs an elementary class?

Example

Question

Let $\mathbb{L}$ be the language with a binary predicate symbol $\dot{E}$ and $\doteq$, but no other symbols. A structure $\mathfrak{G} = (G, E)$ for $\mathbb{L}$ (where $G = |\mathfrak{G}|$ and $E = \dot{E}^{\mathfrak{G}}$ is a *graph* if

- E is symmetric, and
- for every $a \in G$, $(a, a) \notin E$

Is the set of graphs an elementary class?

Answer

Yes. For every structure $\mathfrak{G}$, it is a graph iff $\models_{\mathfrak{G}} \sigma$ where σ is the conjunction of

Example

Question

Let $\mathbb{L}$ be the language with a binary predicate symbol $\dot{E}$ and $\doteq$, but no other symbols. A structure $\mathfrak{G} = (G, E)$ for $\mathbb{L}$ (where $G = |\mathfrak{G}|$ and $E = \dot{E}^{\mathfrak{G}}$ is a *graph* if

- E is symmetric, and
- for every $a \in G$, $(a, a) \notin E$

Is the set of graphs an elementary class?

Answer

Yes. For every structure $\mathfrak{G}$, it is a graph iff $\models_{\mathfrak{G}} \sigma$ where σ is the conjunction of

- $\forall x \forall y (\dot{E}xy \rightarrow \dot{E}yx)$, and

Example

Question

Let $\mathbb{L}$ be the language with a binary predicate symbol $\dot{E}$ and $\doteq$, but no other symbols. A structure $\mathfrak{G} = (G, E)$ for $\mathbb{L}$ (where $G = |\mathfrak{G}|$ and $E = \dot{E}^{\mathfrak{G}}$ is a *graph* if

- E is symmetric, and
- for every $a \in G$, $(a, a) \notin E$

Is the set of graphs an elementary class?

Answer

Yes. For every structure $\mathfrak{G}$, it is a graph iff $\models_{\mathfrak{G}} \sigma$ where σ is the conjunction of

- $\forall x \forall y (\dot{E}xy \to \dot{E}yx)$, and
- $\forall x (\neg \dot{E}xx)$.

Models of a Set of Sentences

Definition

A set of structures $\mathcal{K}$ is an elementary class in the wider sense (EC_Δ) if there is a set Σ of sentences such that

$$\mathcal{K} = \{\mathfrak{A} \mid \mathfrak{A} \text{ is a model of } \Sigma\},$$

i.e.,

$$\mathcal{K} = \{\mathfrak{A} \mid \models_{\mathfrak{A}} \sigma \text{ for every } \sigma \in \Sigma\}.$$

Size of Structures

Question

- $\mathfrak{A}$ has at least two elements iff $\models_{\mathfrak{A}} \lambda_2$ where $\lambda_2 = ?$;

Size of Structures

Question

- $\mathfrak{A}$ has at least two elements iff $\models_{\mathfrak{A}} \lambda_2$ where $\lambda_2 =?$;
- $\mathfrak{A}$ has at least three elements iff $\models_{\mathfrak{A}} \lambda_3$ where $\lambda_3 =?$;

Size of Structures

Question

- $\mathfrak{A}$ has at least two elements iff $\models_{\mathfrak{A}} \lambda_2$ where $\lambda_2 = ?$;
- $\mathfrak{A}$ has at least three elements iff $\models_{\mathfrak{A}} \lambda_3$ where $\lambda_3 = ?$;
- In general, for each positive integer n there is a sentence λ_n such that
  $\mathfrak{A}$ has at least n elements iff $\models_{\mathfrak{A}} \lambda_n$;

Size of Structures

Question

- $\mathfrak{A}$ has at least two elements iff $\models_{\mathfrak{A}} \lambda_2$ where $\lambda_2 = ?$;
- $\mathfrak{A}$ has at least three elements iff $\models_{\mathfrak{A}} \lambda_3$ where $\lambda_3 = ?$;
- In general, for each positive integer n there is a sentence λ_n such that
  $\mathfrak{A}$ has at least n elements iff $\models_{\mathfrak{A}} \lambda_n$;
- $\mathfrak{A}$ has exactly n elements iff $\models_{\mathfrak{A}} \sigma_n$ where $\sigma_n = ?$.

Size of Structures

Question

- $\mathfrak{A}$ has at least two elements iff $\models_{\mathfrak{A}} \lambda_2$ where $\lambda_2 = ?$;
- $\mathfrak{A}$ has at least three elements iff $\models_{\mathfrak{A}} \lambda_3$ where $\lambda_3 = ?$;
- In general, for each positive integer n there is a sentence λ_n such that
  $\mathfrak{A}$ has at least n elements iff $\models_{\mathfrak{A}} \lambda_n$;
- $\mathfrak{A}$ has exactly n elements iff $\models_{\mathfrak{A}} \sigma_n$ where $\sigma_n = ?$.

See the discussion on Page 93 of Enderton's for some of the answers.

Easy and Hard Questions

Question

Is there a set Σ of sentences such that for every $\mathfrak{A}$, $\mathfrak{A}$ is a model of Σ iff $|\mathfrak{A}|$ is infinite?

Easy and Hard Questions

Question

Is there a set Σ of sentences such that for every $\mathfrak{A}$, $\mathfrak{A}$ is a model of Σ iff $|\mathfrak{A}|$ is infinite?

Answer

Yes. Let $\Sigma = \{\lambda_2, \lambda_3, \ldots\}$.

Easy and Hard Questions

Question

Is there a set Σ of sentences such that for every $\mathfrak{A}$, $\mathfrak{A}$ is a model of Σ iff $|\mathfrak{A}|$ is infinite?

Answer

Yes. Let $\Sigma = \{\lambda_2, \lambda_3, \ldots\}$.

Now, ask the following question:

Question

Is there a single sentence σ such that $\mathfrak{A}$ is a model of σ iff $|\mathfrak{A}|$ is *infinite*?

Some Hard and Very Hard Questions

- Is there a set Σ of sentences such that for every $\mathfrak{A}$, $\mathfrak{A}$ is a model of Σ iff $|\mathfrak{A}|$ is *finite*?

Some Hard and Very Hard Questions

- Is there a set Σ of sentences such that for every $\mathfrak{A}$, $\mathfrak{A}$ is a model of Σ iff $|\mathfrak{A}|$ is *finite*?
- Is there a set Σ of sentences such that for every $\mathfrak{A}$, $\mathfrak{A}$ is a model of Σ iff $|\mathfrak{A}|$ is *enumerable*?

Some Hard and Very Hard Questions

- Is there a set Σ of sentences such that for every $\mathfrak{A}$, $\mathfrak{A}$ is a model of Σ iff $|\mathfrak{A}|$ is *finite*?
- Is there a set Σ of sentences such that for every $\mathfrak{A}$, $\mathfrak{A}$ is a model of Σ iff $|\mathfrak{A}|$ is *enumerable*?
- Is there a set Σ of sentences such that for every $\mathfrak{A}$, $\mathfrak{A}$ is a model of Σ iff $|\mathfrak{A}|$ is *uncountable*?