# Mathematical Logic

(Based on lecture slides by Stan Burris)

**George Voutsadakis**[1]

[1]Mathematics and Computer Science
Lake Superior State University

LSSU Math 300

1. First-Order Languages
   - First-Order Languages without Equality
   - Interpretations and Structures
   - The Syntax of First-Order Logic
   - First-Order Syntax for the Natural Numbers
   - The Semantics of First-Order Sentences in $\mathbb{N}$
   - Other Number Systems
   - First-Order Syntax for Directed Graphs
   - The Semantics of First-Order Sentences in Directed Graphs
   - Semantics for First-Order Logic
   - Equivalent Formulas
   - Replacement and Substitution
   - Prenex Form
   - Valid Arguments
   - Skolemization

Subsection 1

## First-Order Languages without Equality

# First-Order Languages without Equality

- A **first-order language without equality** $\mathcal{L}$ consists of
  - a set $\mathcal{F}$ of **function symbols** $f, g, h, \ldots$, with associated **arities**;
  - a set $\mathcal{R}$ of **relation symbols** $r, r_1, r_2, \ldots$, with associated **arities**;
  - a set $\mathcal{C}$ of **constant symbols** $c, d, e, \ldots$;
  - a set $X$ of **variables** $x, y, z, \ldots$.
- Each relation symbol $r$ has a positive integer, called its **arity**, assigned to it; If the number is $n$, we say $r$ is $n$-**ary**. For small $n$ we use the same special names that we use for function symbols: **unary, binary, ternary, quaternary**.
- The set $\mathcal{L} = \mathcal{R} \cup \mathcal{F} \cup \mathcal{C}$ is called a **first-order language**.
- For instance, if we want to work with the integers, dealing both with their operations and their ordering, the language $\{+, \cdot, <, -, 0, 1\}$ would be a natural choice.

Subsection 2

## Interpretations and Structures

# Interpretation of Relation Symbols

- The obvious interpretation of a relation symbol is as a **relation** on a set.
- If $A$ is a set and $n$ is a positive integer, then an n-**ary relation** $r$ on $A$ is a subset of $A^n$; that is, $r$ consists of a collection of $n$-**tuples** $(a_1, \ldots, a_n)$ of elements of $A$.
- Example: The ordinary "less than" relation on the reals is the binary relation

$$r = \{(x, y) \in \mathbb{R}^2 : x < y\};$$

- Example: The adjacency relation on the vertices of a graph is the binary relation

$$r = \{(x, y) \in V^2 : x \text{ and } y \text{ are adjacent}\};$$

- Recall the notions of a reflexive, symmetric, anti-symmetric, asymmetric, transitive, equivalence binary relation on a set $A$;

# Formal Definitions of Properties of Binary Relations

- Let $A$ be a set. A binary relation $r \subseteq A^2$ is called:
  - **reflexive** if $(a, a) \in r$, for all $a \in A$;
  - **irreflexive** if $(a, a) \notin r$, for all $a \in A$;
  - **symmetric** if $(a, b) \in r$ implies $(b, a) \in r$, for all $a, b \in A$;
  - **anti-symmetric** if
    $(a, b) \in r$ and $(b, a) \in r$ imply $a = b$, for all $a, b \in A$;
  - **asymmetric** if $(a, b) \in r$ implies $(b, a) \notin r$, for all $a, b \in A$;
  - **transitive** if
    $(a, b) \in r$ and $(b, c) \in r$ imply $(a, c) \in r$, for all $a, b, c \in A$;
  - **equivalence** if it is reflexive, symmetric and transitive;
  - **partial order** if it is reflexive, anti-symmetric and transitive;
  - **strict order** if it is irreflexive and transitive (which implies asymmetric).

# Interpretations

- An **interpretation** $I$ of the first-order language $\mathcal{L}$ on a set $S$ is a mapping with domain $\mathcal{L}$ such that
    - $I(c)$ is an element of $S$ for each constant symbol $c$ in $\mathcal{C}$;
    - $I(f)$ is an $n$-ary function on $S$ for each n-ary function symbol $f$ in $\mathcal{F}$;
    - $I(r)$ is an $n$-ary relation on $S$ for each n-ary relation symbol $r$ in $\mathcal{R}$;
- An $\mathcal{L}$-**structure** $\mathbf{S}$ is a pair $\mathbf{S} = (S, I)$, where
    - $S$ is a set;
    - $I$ is an interpretation of $\mathcal{L}$ on $S$.

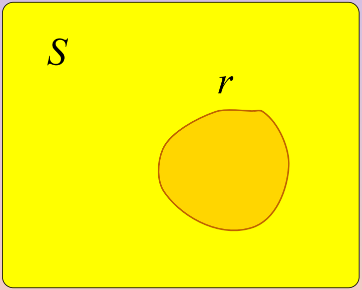
# Notation and Example

- We sometimes write
  - $c^{\mathbf{S}}$ (or just $c$) for $I(c)$;
  - $f^{\mathbf{S}}$ (or just $f$) for $I(f)$;
  - $r^{\mathbf{S}}$ (or just $r$) for $I(r)$;
  - $(S, \mathcal{F}, \mathcal{R}, \mathcal{C})$ for $(S, I)$;
- Example: The structure $\mathbb{R} = (\mathbb{R}, +, \cdot, <, 0, 1)$, the reals with addition, multiplication, less than, and two specified constants has:

$$\mathcal{F} = \{+, \cdot\}, \qquad \mathcal{R} = \{<\}, \qquad \mathcal{C} = \{0, 1\}.$$

# Unary Relation Symbols and Subsets

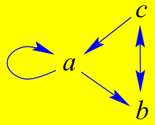
- If $r \in \mathcal{R}$ is a unary relation symbol, then in any $\mathcal{L}$-structure **S**, the relation $r^{\mathbf{S}}$ is a subset of $S$;
- We can picture this as:



# Binary Relation Symbols and Directed Graphs

- If $\mathcal{L}$ consists of a single binary relation symbol $r$, then we call an $\mathcal{L}$-structure a **directed graph**.
- A small finite directed graph can be conveniently described in three different ways:
  - By listing the ordered pairs in the relation r.
    A simple example, with $S = \{a, b, c\}$, is
    $r^{\mathbf{S}} = \{(a, a), (a, b), (b, c), (c, b), (c, a)\}$.
  - By a table: (1 indicates a pair is in the relation.)

| $r$ | $a$ | $b$ | $c$ |
|---|---|---|---|
| $a$ | 1 | 1 | 0 |
| $b$ | 0 | 0 | 1 |
| $c$ | 1 | 1 | 0 |




  - By drawing a picture:

# An Example of a First-Order Structure

- An interpretation of a language on a small set can be conveniently given by tables;
- Suppose that $\mathcal{L} = \{+, <\}$, where
  - $+$ is a binary function symbol;
  - $<$ is a binary relation symbol;
- The following tables give an interpretation $\mathbf{S} = (S, +^{\mathbf{S}}, <^{\mathbf{S}})$ of $\mathcal{L}$ on the two element set $S = \{a, b\}$:

| $+$ | $a$ | $b$ |
|---|---|---|
| $a$ | $a$ | $b$ |
| $b$ | $b$ | $a$ |

| $<$ | $a$ | $b$ |
|---|---|---|
| $a$ | 0 | 1 |
| $b$ | 0 | 0 |

Subsection 3

## The Syntax of First-Order Logic

# The Vocabulary of First-Order Logic

- **First-Order Logic** is adequate for expressing almost all reasoning performed in mathematics;
- It is the most powerful, most expressive logic that our textbook examines;
- It can be presented in many different ways;
- Our version of first-order logic will use the following symbols:
  - variables (these are individual, not propositional variables);
  - connectives ($\vee, \wedge, \rightarrow, \leftrightarrow, \neg$);
  - function symbols;
  - relation symbols;
  - constant symbols;
  - equality ($\approx$);
  - quantifiers ($\forall, \exists$).

# First-Order Formulas

- **Atomic Formulas** for a first-order language $\mathcal{L}$ are of two kinds:
  - $s \approx t$, where $s$ and $t$ are terms;
  - $(rt_1 \cdots t_n)$, where $r$ is an $n$-ary relation symbol and $t_1, \ldots, t_n$ are terms;
- **Formulas** for a first-order language $\mathcal{L}$ are defined inductively as follows:
  - Atomic formulas are formulas;
  - If $F$ is a formula, then so is $(\neg F)$;
  - If $F$ and $G$ are formulas, then so are

$$(F \vee G), \quad (F \wedge G), \quad (F \rightarrow G), \quad (F \leftrightarrow G);$$

  - If $F$ is a formula and $x$ is a variable, then $(\forall x F)$ and $(\exists x F)$ are formulas.

# Notational Conventions

- Drop outer parentheses;
- Adopt the previous precedence conventions for the propositional connectives (negation $\neg$ first, disjunction $\vee$ and conjunction $\wedge$ next, implication $\rightarrow$ and equivalence $\leftrightarrow$ last);
- Quantifiers bind more strongly than any of the connectives;
- Following those conventions, the expression

$$\forall y(rxy) \vee \exists y(rxy)$$

stands for the formula

$$((\forall y(rxy)) \vee (\exists y(rxy)))$$

# Subformulas of First-Order Formulas

- The **subformulas** of a formula $F$ are defined recursively as follows:
  - The only subformula of an atomic formula $F$ is $F$ itself;
  - The subformulas of $\neg F$ are $\neg F$ itself and all the subformulas of $F$;
  - The subformulas of $F \Box G$ are $F \Box G$ itself and all the subformulas of $F$ and all the subformulas of $G$; ($\Box$ is any of $\vee, \wedge, \rightarrow, \leftrightarrow$);
  - The subformulas of $\forall x F$ are $\forall x F$ itself and all the subformulas of $F$;
  - The subformulas of $\exists x F$ are $\exists x F$ itself and all the subformulas of $F$.

# Bound and Free Variables

- An occurrence of a variable $x$ in a formula $F$ is:
  - **bound** if the occurrence is in a subformula

    of the form $\forall xG$ or of the form $\exists xG$;

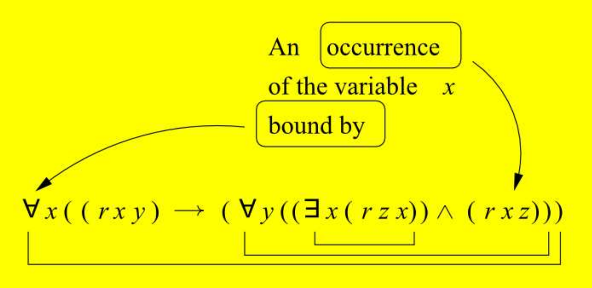
    Such a subformula is called the **scope of the quantifier** that begins the subformula.
  - Otherwise the occurrence of the variable is said to be **free**;
- Note that the same variable may occur both bound and free in the same formula; e.g.,

$$\exists x(x \approx y) \wedge \forall y(rxy)$$

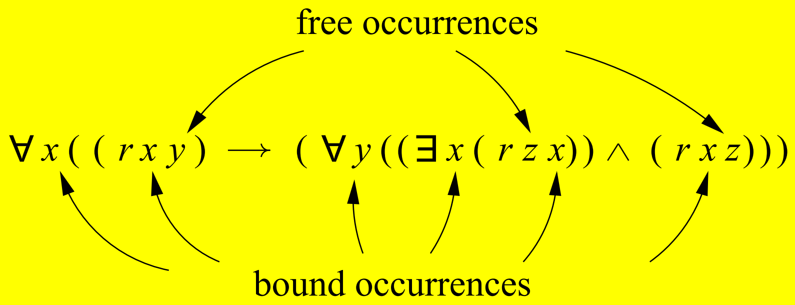
  Thus, bound and free refer to occurrences of a variable, not to the variable itself!
- A formula with no free occurrences of variables is called a **sentence**.

# Quantifiers Binding Variables

- Given a bound occurrence of $x$ in $F$, we say that $x$ is **bound by an occurrence of a quantifier** $Q$ if
  - (i) the occurrence of $Q$ quantifies the variable $x$, and
  - (ii) subject to this constraint the scope of this occurrence of $Q$ is the smallest in which the given occurrence of $x$ occurs.
- It is easier to explain scope, and quantifiers that bind variables, with a diagram; In the diagram scopes of quantifiers are underlined;



# Example with Free and Bound Occurrences of Variables



Subsection 4

# First-Order Syntax for the Natural Numbers

# The Language $\mathcal{L}_N$ for the Natural Numbers

- To discuss formally the natural number system, we consider the language
$$\mathcal{L}_N = \{+, \cdot, <, 0, 1\};$$
- The $\mathcal{L}_N$-structure $\mathbb{N} = (\mathbb{N}, +, \cdot, <, 0, 1)$ represents the natural numbers with
  - ordinary addition $+$;
  - ordinary multiplication $\cdot$;
  - ordinary strict ordering $<$;
  - constants the natural numbers 0 and 1;
- The atomic $\mathcal{L}_N$-formulas are
  - $(s \approx t)$;
  - $(s < t)$;
- For instance, the following are all atomic $\mathcal{L}_N$-formulas:
$$(0 < 0) \qquad (1 < 0) \qquad (x < 0) \qquad (x \cdot (y + z) < x \cdot z)$$
$$(x \cdot (y + 1) < x \cdot x + y \cdot z)$$

# $\mathcal{L}_N$-Formulas

- The following are $\mathcal{L}_N$-formulas:

$$((x < y) \to (x + x < y + y))$$
$$(\forall x((x \cdot (y + 1) < x \cdot x + y \cdot z) \to (\exists y(y \cdot y < x + z))))$$

- Consider the formula:

$$(\forall x(x \cdot (y + 1) < x \cdot x + y \cdot z)) \to (\exists y(y \cdot y < x + z))$$

Its subformulas are:

$$(\forall x(x \cdot (y + 1) < x \cdot x + y \cdot z)) \to (\exists y(y \cdot y < x + z))$$
$$\forall x(x \cdot (y + 1) < x \cdot x + y \cdot z) \qquad \exists y(y \cdot y < x + z)$$
$$x \cdot (y + 1) < x \cdot x + y \cdot z \qquad y \cdot y < x + z$$

- When working with the language $\mathcal{L}_N$, one uses the abbreviations
  - $2$ stands for $1 + 1$; $3$ stands for $(1 + 1) + 1$; etc.
- For instance, $3 < 5$ stands for $(1 + 1) + 1 < (((1 + 1) + 1) + 1) + 1$; it is an atomic $\mathcal{L}_N$-sentence saying that "3 is less than 5"; This sentence is true in the $\mathcal{L}_N$-structure $\mathbb{N}$.

Subsection 5

# The Semantics of First-Order Sentences in $\mathbb{N}$

# Examples of First-Order Formulas with Intuition

- $2+2<3$ is an atomic sentence; It says "four is less than three".
  False in $\mathbb{N}$.
- $\forall x \exists y (x<y)$ says that "for every number there is a larger number".
  True in $\mathbb{N}$.
- $\exists y \forall x (x<y)$ says that "there is a number that is larger than every other number".
  False in $\mathbb{N}$.
- $\forall x((0<x) \to \exists y(y \cdot y \approx x))$ says that "every positive number is a square".
  False in $\mathbb{N}$.
- $\forall x \forall y((x<y) \to \exists z((x<z) \wedge (z<y)))$ says that "if one number is less than another, then there is a number properly between the two".
  False in $\mathbb{N}$.

# Notation for Meets and Joins

- We will use the shorthand notation
$$\bigwedge_{i=1}^{n} F_i$$
to mean the same as the notation
$$F_1 \wedge \cdots \wedge F_n.$$
- Likewise, we will use the notation
$$\bigvee_{i=1}^{n} F_i$$
for
$$F_1 \vee \cdots \vee F_n.$$

# Translating English to First-Order I

- Suppose that $F(x)$ is a first-order formula with variable $x$; We can find first-order sentences to say:
  a. "There is at least one number such that $F(x)$ is true in $\mathbb{N}$".
     $\exists x F(x)$
  b. "There are at least two numbers such that $F(x)$ is true in $\mathbb{N}$".
     $\exists x \exists y(\neg(x \approx y) \wedge F(x) \wedge F(y))$
  c. "There are at least $n$ numbers ($n$ fixed) such that $F(x)$ is true in $\mathbb{N}$".
     $\exists x_1 \cdots \exists x_n((\bigwedge_{1 \leq i < j \leq n} \neg(x_i \approx x_j)) \wedge (\bigwedge_{1 \leq i \leq n} F(x_i)))$
  d. "There are infinitely many numbers that make $F(x)$ true in $\mathbb{N}$".
     $\forall x \exists y((x < y) \wedge F(y))$

# Translating English to First-Order II

- We can also find first-order sentences to say:
  - e. "There is at most one number such that $F(x)$ is true in $\mathbb{N}$".
    $\forall x \forall y((F(x) \wedge F(y)) \rightarrow (x \approx y))$
  - f. "There are at most two numbers such that $F(x)$ is true in $\mathbb{N}$".
    $\forall x \forall y \forall z((F(x) \wedge F(y) \wedge F(z)) \rightarrow ((x \approx y) \vee (x \approx z) \vee (y \approx z)))$
  - g. "There are at most $n$ numbers ($n$ fixed) such that $F(x)$ is true in $\mathbb{N}$".
    $\forall x_1 \cdots \forall x_{n+1}((\bigwedge_{1 \leq i \leq n+1} F(x_i)) \rightarrow (\bigvee_{1 \leq i < j \leq n+1}(x_i \approx x_j)))$
  - h. "There are only finitely many numbers that make $F(x)$ true in $\mathbb{N}$".
    $\exists x \forall y(F(y) \rightarrow (y < x))$

# Truth of a Formula at a Tuple of Domain Elements

- To better understand what we can express with first-order sentences we need to introduce definable relations;
- Given a first-order formula $F(x_1, \ldots, x_k)$, we say $F$ **is true at** a $k$-tuple $(a_1, \ldots, a_k)$ of natural numbers if the expression $F(a_1, \ldots, a_k)$ is a true statement about the natural numbers;
- Example: Let $F(x, y)$ be the formula $x < y$. Then $F$ is true at $(a, b)$ iff $a$ is less than $b$.
- Example: Let $F(x, y)$ be $\exists z(x \cdot z \approx y)$. Then $F$ is true at $(a, b)$ iff $a$ divides $b$, written $a \backslash b$.
- **Important Note:** Don't confuse $a \backslash b$ with $\frac{a}{b}$. The first is true or false. The second has a value.
  Check that $a \backslash 0$ for any $a$, including $a = 0$.

# Definable Relations

- For $F(x_1, \ldots, x_k)$ a formula, let $F^{\mathbb{N}}$ be the set of k-tuples $(a_1, \ldots, a_k)$ of natural numbers for which $F(a_1, \ldots, a_k)$ is true in $\mathbb{N}$;
- We call $F^{\mathbb{N}}$ the **relation on** $\mathbb{N}$ **defined by** the formula $F$;
- A $k$-ary relation $r \subseteq \mathbb{N}^k$ is **definable in** $\mathbb{N}$ if there is a formula $F(x_1, \ldots, x_k)$ such that $r = F^{\mathbb{N}}$;
- Examples:
  a. $x$ is an even number is definable in $\mathbb{N}$ by
  $$\exists y(x \approx y + y).$$
  b. $x$ divides $y$ is definable in $\mathbb{N}$ by
  $$\exists z(x \cdot z \approx y).$$

# More Examples of Definable Relations

- We continue the list of Examples:
  c. $x$ is prime is definable in $\mathbb{N}$ by
  $$(1 < x) \wedge \forall y((y \backslash x) \to ((y \approx 1) \vee (y \approx x))).$$
  d. $x \equiv y \text{ modulo } n$ is definable in $\mathbb{N}$ by
  $$\exists z((x \approx y + n \cdot z) \vee (y \approx x + n \cdot z)).$$
  e. $z$ is the remainder of dividing $x$ by $y$ is definable in $\mathbb{N}$ by
  $$z < y \wedge \exists w(x \approx w \cdot y + z).$$

# Using Abbreviations; The Metalanguage

- When we write $x \backslash y$, we understand the formula $\exists z(x \cdot z \approx y)$.
- When we write prime($x$), we understand the formula

$$(1 < x) \wedge \forall y((y \backslash x) \rightarrow ((y \approx 1) \vee (y \approx x))).$$

- Note that in prime($x$), we have used the abbreviation for $y \backslash x$. This means that to properly write prime($x$) as a first-order formula we need to replace that abbreviation; doing so gives us

$$(1 < x) \wedge \forall y((\exists z(y \cdot z \approx x)) \rightarrow ((y \approx 1) \vee (y \approx x))).$$

- Abbreviations are not a feature of first-order logic, but rather they are a tool in the language used by people to discuss first-order logic; To distinguish this language for the language of first-order logic, we sometimes call it the **metalanguage**;
- Without abbreviations, writing out the first-order sentences that we find interesting would fill up lines with tedious, hard-to-read symbolism.

# Substitution Needs Care!

- We saw that $x \backslash y$ abbreviates $\exists z(x \cdot z \approx y)$;
- Then $(u+1) \backslash (u \cdot u + 1)$ is an abbreviation for

$$\exists z((u+1) \cdot z \approx u \cdot u + 1);$$

- If we write out $z \backslash 2$ we obtain $\exists z(z \cdot z \approx 1 + 1)$.
- Unfortunately, this last formula does not define the set of elements in $\mathbb{N}$ that divide 2; It is a first-order sentence that is simply false in $\mathbb{N}$; the square root of 2 is not a natural number;

# Renaming "Dummy" Variables

- We have stumbled onto one of the subtler points of first-order logic, namely, we must be careful with substitution;
- The remedy for defining "$z$ divides 2" is to use another formula, like

$$\exists w(x \cdot w \approx y)$$

  for "$x$ divides $y$".
- We obtain such a formula by simply renaming the bound variable $z$ in the formula for $x \backslash y$;
- With this formula we can correctly express "$z$ divides 2" by $\exists w(z \cdot w \approx 2)$.
- The danger in using abbreviations in first-order logic, as showcased by this example, is that we forget the names of the bound variables in the abbreviation.

# Substitution: Alerting Reader of the Danger

- Our solution: add a $\star$ to the abbreviation to alert the reader to the necessity for renaming the bound variables that overlap with the variables in the term to be substituted into the abbreviation;
- For example, we write $\text{prime}^\star(y+z)$ to explicitly express the need to change the formula for $\text{prime}(x)$, say to

$$(1 < x) \wedge \forall v((v \backslash x) \to ((v \approx 1) \vee (v \approx x)))$$

  so that when we substitute $y+z$ for $x$ in the formula, no new occurrence of $y$ or $z$ becomes bound.
- Thus we could express $\text{prime}(y+z)$ by

$$(1 < y+z) \wedge \forall v((v \backslash^\star (y+z)) \to ((v \approx 1) \vee (v \approx y+z))).$$

# Expressing Statements in First-Order Logic

a. The relation "divides" is transitive:
$\forall x \forall y \forall z(((x \backslash y) \wedge (y \backslash^\star z)) \to (x \backslash^\star z))$.

b. There are an infinite number of primes:
$\forall x \exists y((x < y) \wedge \text{prime}^\star(y))$.

c. **The Twin Prime Conjecture**
There are an infinite number of pairs of primes that differ by the number 2:
$\forall x \exists y((x < y) \wedge \text{prime}^\star(y) \wedge \text{prime}^\star(y+2))$.

d. **Goldbach's Conjecture**
All even numbers greater than two are the sum of two primes:
$\forall x(((2 \backslash x) \wedge (2 < x)) \to \exists y \exists z(\text{prime}^\star(y) \wedge \text{prime}(z) \wedge (x \approx y + z)))$.

Subsection 6

## Other Number Systems

# Other Number Systems: Integers, Rationals and Reals

- Our first-order language $\mathcal{L} = \{+, \cdot, <, 0, 1\}$ can just as easily be used to study other number systems, in particular,
  - the **integers** $\mathbb{Z} = (\mathbb{Z}, +, \cdot, <, 0, 1)$;
  - the **rationals** $\mathbb{Q} = (\mathbb{Q}, +, \cdot, <, 0, 1)$;
  - the **reals** $\mathbb{R} = (\mathbb{R}, +, \cdot, <, 0, 1)$;
- However, first-order sentences that are true in one can be false in another.

# Sentences Considered in Various Models

- Consider the following first-order sentences:
  (a) $\forall x \exists y (x < y)$
  "For every number, there is a (strictly) greater number".
  (b) $\forall y \exists x (x < y)$
  "For every number, there exists a (strictly) smaller number".
  (c) $\forall x \forall y ((x < y) \rightarrow \exists z ((x < z) \wedge (z < y)))$
  "For every two different numbers, there exists a number lying (properly) between the two".
- The following table evaluates the truth of (a)-(e) in the models $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ and $\mathbb{R}$:

| | $\mathbb{N}$ | $\mathbb{Z}$ | $\mathbb{Q}$ | $\mathbb{R}$ |
|---|---|---|---|---|
| $(a)$ | true | true | true | true |
| $(b)$ | false | true | true | true |
| $(c)$ | false | false | true | true |
| $(d)$ | | | | |
| $(e)$ | | | | |

# Sentences Considered in Various Models

- Two more first-order sentences:
  (d) $\forall x \exists y((0 < x) \rightarrow (x \approx y \cdot y))$
  "Every positive number has a square root".
  (e) $\exists x \forall y(x < y)$
  "There exists a number (strictly) less than all numbers".
- The following table evaluates the truth of (a)-(e) in the models $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ and $\mathbb{R}$:

| | $\mathbb{N}$ | $\mathbb{Z}$ | $\mathbb{Q}$ | $\mathbb{R}$ |
|---|---|---|---|---|
| $(a)$ | true | true | true | true |
| $(b)$ | false | true | true | true |
| $(c)$ | false | false | true | true |
| $(d)$ | false | false | false | true |
| $(e)$ | false | false | false | false |

Subsection 7

# First-Order Syntax for Directed Graphs

# The Language of Directed Graphs

- The first-order **language of (directed) graphs** is $\mathcal{L} = \{r\}$, where $r$ is a binary relation symbol;
- The only terms are the variables $x$;
- Atomic formulas look like
  - $(x \approx y)$;
  - $(rxy)$;
- **Example:** The subformulas of $\forall x((rxy) \to \exists y(ryx))$ are

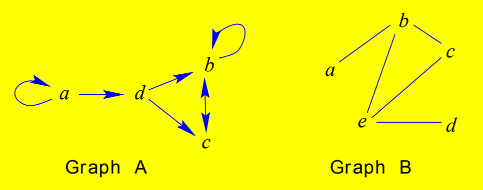
$$\forall x((rxy) \to \exists y(ryx))$$
$$(rxy) \to \exists y(ryx)$$
$$rxy$$
$$\exists y(ryx)$$
$$ryx$$

Subsection 8

## The Semantics of First-Order Sentences in Directed Graphs

# First-Order to English On Directed Graphs

- Two structures over the language $\mathcal{L} = \{r\}$ of directed graphs:




- We consider some first-order logic sentences over $\mathcal{L}$:
  a. $\forall x \neg (rxx)$
     It says: "the directed graph is **irreflexive**". False in $A$; True in $B$;
  b. $\forall x \forall y ((rxy) \rightarrow (ryx))$
     It says: "the directed graph is **symmetric**". False in $A$; True in $B$;
  c. $\forall x \forall y (rxy)$
     It says: "all possible edges are present". False in $A$; False in $B$;
  d. $\forall x \exists y (rxy)$
     It says: "for every vertex there is an outgoing edge". True in $A$; True in $B$;

# English to First-Order Logic On Directed Graphs

- Consider the following statements:
  a. The (directed) graph has at least two vertices.
     $\exists x \exists y(\neg(x \approx y))$
  b. Every vertex has an edge attached to it.
     $\forall x \exists y((rxy) \vee (ryx))$
  c. Every vertex has at most two edges directed from it to other vertices.
     $\forall x \forall y \forall z \forall w(((rxy) \wedge (rxz) \wedge (rxw)) \rightarrow ((y \approx z) \vee (y \approx w) \vee (w \approx z)))$

# Some Graph-Theoretic Definitions

- The **degree** of a vertex is the number of (undirected) edges attached to it;
- A **path of length** $n$ **from** vertex $x$ **to** vertex $y$ is a sequence of vertices $a_1, \ldots, a_{n+1}$ with each $(a_i, a_{i+1})$ being an edge, and with $x = a_1$, $y = a_{n+1}$;
- Two vertices are **adjacent** if there is an edge connecting them.

# Definable Relations and Statements about Graphs

- The following are definable relations on graphs:
  a. The degree of x is at least one.
     $\exists y(rxy)$
  b. The degree of x is at least two.
     $\exists y \exists z(\neg(y \approx z) \wedge (rxy) \wedge (rxz))$
- The following are statements about graphs:
  a. Some vertex has degree at least two.
     $\exists x \exists y \exists z(\neg(y \approx z) \wedge (rxy) \wedge (rxz))$
  b. Every vertex has degree at least two.
     $\forall x \exists y \exists z(\neg(y \approx z) \wedge (rxy) \wedge (rxz))$

Subsection 9

## Semantics for First-Order Logic

# Overview of First-Order Semantics

- Given a first-order $\mathcal{L}$-structure $\mathbf{S} = (S, I)$, the interpretation $I$ gives meaning to the symbols of the language $\mathcal{L}$;
- We associate with each term $t(x_1, \ldots, x_n)$ the $n$-ary term function $t^{\mathbf{S}} : S^n \to S$;
- We associate with each formula $F(x_1, \ldots, x_n)$ an $n$-ary relation $F^{\mathbf{S}} \subseteq S^n$;
- We continue with the formal definition after a small break!

# Alfred Tarski

- Alfred Tarski, born in Warsaw, Kingdom of Poland (1901-1983)

# Tarski's Definition of Truth

- The notion of a formula $F$ **being true** or **holding in** a structure $\mathbf{S} = (S, I)$ **under an assignment** $\vec{a}$ of values from $S$ to its variables $\vec{x}$ is defined by induction on the structure of $F$:
  - $F(\vec{x})$ is atomic:
    - $F$ is the formula $t_1(\vec{x}) \approx t_2(\vec{x})$: $F(\vec{a})$ **holds** iff $t_1^{\mathbf{S}}(\vec{a}) = t_2^{\mathbf{S}}(\vec{a})$.
    - $F$ is the formula $r(t_1(\vec{x}), \ldots, t_n(\vec{x}))$: $F(\vec{a})$ **holds** iff $r^{\mathbf{S}}(t_1^{\mathbf{S}}(\vec{a}), \ldots, t_n^{\mathbf{S}}(\vec{a}))$ holds.
  - $F = \neg G$: Then $F(\vec{a})$ **holds** iff $G(\vec{a})$ does not hold.
  - $F = G \vee H$: Then $F(\vec{a})$ **holds** iff $G(\vec{a})$ holds or $H(\vec{a})$ holds.
  - $F = G \wedge H$: Then $F(\vec{a})$ **holds** iff $G(\vec{a})$ holds and $H(\vec{a})$ holds.
  - $F = G \rightarrow H$: Then $F(\vec{a})$ **holds** iff $G(\vec{a})$ does not hold or $H(\vec{a})$ holds.
  - $F = G \leftrightarrow H$: Then $F(\vec{a})$ **holds** iff both or neither of $G(\vec{a})$ and $H(\vec{a})$ holds.
  - $F(\vec{x})$ is $\forall y G(y, \vec{x})$: Then $F(\vec{a})$ **holds** iff $G(b, \vec{a})$ holds for every $b \in S$.
  - $F(\vec{x})$ is $\exists y G(y, \vec{x})$: Then $F(\vec{a})$ **holds** iff $G(b, \vec{a})$ holds for some $b \in S$.

# Illustrating the Definition of Truth I

- Consider the language $\mathcal{L} = \{f, r\}$, where
  - $f$ is a unary function symbol;
  - $r$ is a binary relation symbol;
- Consider the $\mathcal{L}$-structure $\mathbf{S} = (S, f^{\mathbf{S}}, r^{\mathbf{S}})$, with

$S = \{a, b\}$,

| $x$ | $fx$ |
|---|---|
| $a$ | $b$ |
| $b$ | $a$ |

| $r$ | $a$ | $b$ |
|---|---|---|
| $a$ | 0 | 1 |
| $b$ | 1 | 0 |

- Consider the $\mathcal{L}$-formula

$$F(x) = \forall y \exists z((rfxfy) \wedge (rfyfz)).$$

- In the next slide, we evaluate $F(x)$ at both $x = a$ and $x = b$ in $\mathbf{S}$; i.e., we fully determine $F^{\mathbf{S}}$ (set of all $x \in S$ for which $F(x)$ holds).

# Evaluation of $F(x) = \forall y \exists z((rfxfy) \wedge (rfyfz))$

| $x$ | $y$ | $z$ | $fx$ | $fy$ | $fz$ | $rfxfy$ | $rfyfz$ | $(rfxfy) \wedge (rfyfz)$ |
|---|---|---|---|---|---|---|---|---|
| $a$ | $a$ | $a$ | $b$ | $b$ | $b$ | 0 | 0 | 0 |
| $a$ | $a$ | $b$ | $b$ | $b$ | $a$ | 0 | 1 | 0 |
| $a$ | $b$ | $a$ | $b$ | $a$ | $b$ | 1 | 1 | 1 |
| $a$ | $b$ | $b$ | $b$ | $a$ | $a$ | 1 | 0 | 0 |
| $b$ | $a$ | $a$ | $a$ | $b$ | $b$ | 1 | 0 | 0 |
| $b$ | $a$ | $b$ | $a$ | $b$ | $a$ | 1 | 1 | 1 |
| $b$ | $b$ | $a$ | $a$ | $a$ | $b$ | 0 | 1 | 0 |
| $b$ | $b$ | $b$ | $a$ | $a$ | $a$ | 0 | 0 | 0 |

| $x$ | $y$ | $\exists z((rfxfy) \wedge (rfyfz))$ |
|---|---|---|
| $a$ | $a$ | 0 |
| $a$ | $b$ | 1 |
| $b$ | $a$ | 1 |
| $b$ | $b$ | 0 |

| $x$ | $\forall y \exists z((rfxfy) \wedge (rfyfz))$ |
|---|---|
| $a$ | 0 |
| $b$ | 0 |

Therefore $F^{\mathbf{S}} = \emptyset$.

# Illustrating the Definition of Truth II

- Consider the same language $\mathcal{L} = \{f, r\}$;
- Consider the same $\mathcal{L}$-structure $\mathbf{S} = (S, f^{\mathbf{S}}, r^{\mathbf{S}})$, with

$S = \{a, b\}$,

| $x$ | $fx$ |
|---|---|
| $a$ | $b$ |
| $b$ | $a$ |

,

| $r$ | $a$ | $b$ |
|---|---|---|
| $a$ | 0 | 1 |
| $b$ | 1 | 0 |

- Consider the $\mathcal{L}$-formula

$$F(x, y) = \exists z((rxfz) \wedge (fy \approx z)) \to (fy \approx fx).$$

- In the next slide, we evaluate $F(x, y)$ at all pairs $(a, b) \in S^2$; i.e., we fully determine $F^{\mathbf{S}}$ (set of all $(x, y) \in S^2$ for which $F(x, y)$ holds).

# Evaluation of $F(x, y) = \exists z((rxfz) \wedge (fy \approx z)) \to (fy \approx fx)$

| $x$ | $y$ | $z$ | $fx$ | $fy$ | $fz$ | $rxfz$ | $fy \approx z$ | $rxfz \wedge fy \approx z$ |
|---|---|---|---|---|---|---|---|---|
| $a$ | $a$ | $a$ | $b$ | $b$ | $b$ | 1 | 0 | 0 |
| $a$ | $a$ | $b$ | $b$ | $b$ | $a$ | 0 | 1 | 0 |
| $a$ | $b$ | $a$ | $b$ | $a$ | $b$ | 1 | 1 | 1 |
| $a$ | $b$ | $b$ | $b$ | $a$ | $a$ | 0 | 0 | 0 |
| $b$ | $a$ | $a$ | $a$ | $b$ | $b$ | 0 | 0 | 0 |
| $b$ | $a$ | $b$ | $a$ | $b$ | $a$ | 1 | 1 | 1 |
| $b$ | $b$ | $a$ | $a$ | $a$ | $b$ | 0 | 1 | 0 |
| $b$ | $b$ | $b$ | $a$ | $a$ | $a$ | 1 | 0 | 0 |

| $x$ | $y$ | $\exists z(rxfz \wedge fy \approx z)$ | $fy \approx fx$ | $\exists z((rxfz) \wedge (fy \approx z)) \to (fy \approx fx)$ |
|---|---|---|---|---|
| $a$ | $a$ | 0 | 1 | 1 |
| $a$ | $b$ | 1 | 0 | 0 |
| $b$ | $a$ | 1 | 0 | 0 |
| $b$ | $b$ | 0 | 1 | 1 |

Therefore $F^{\mathbf{S}} = \{(a, a), (b, b)\}$.

# Definition of Truth for Sentences

- Let $\mathcal{L}$ be a language, $F$ be an $\mathcal{L}$-sentence and **S** an $\mathcal{L}$-structure;
- Then $F$ is **true in S** provided one of the following holds:
  - $F$ is $rt_1 \ldots t_n$ and $r^{\mathbf{S}}(t_1^{\mathbf{S}}, \ldots, t_n^{\mathbf{S}})$ holds;
  - $F$ is $t_1 \approx t_2$ and $t_1^{\mathbf{S}} = t_2^{\mathbf{S}}$;
  - $F$ is $\neg G$ and $G$ is not true in **S**;
  - $F$ is $G \vee H$ and at least one of $G, H$ is true in **S**;
  - $F$ is $G \wedge H$ and both of $G, H$ are true in **S**;
  - $F$ is $G \rightarrow H$ and $G$ is not true in **S** or $H$ is true in **S**;
  - $F$ is $G \leftrightarrow H$ and both or neither of $G, H$ is true in **S**;
  - $F$ is $\forall x G(x)$ and $G^{\mathbf{S}}(a)$ is true for every $a \in S$;
  - $F$ is $\exists x G(x)$ and $G^{\mathbf{S}}(a)$ is true for some $a \in S$.
- If $F$ is not true in **S**, then we say $F$ is **false in S**.

# Notational Conventions for Truth

- Given a first-order language $\mathcal{L}$, let $F$ be an $\mathcal{L}$-sentence, $\mathcal{S}$ a set of $\mathcal{L}$-sentences, and $\mathbf{S}$ a structure for this language;
- $\mathbf{S} \models F$ means $F$ is true in $\mathbf{S}$;
- $F$ is **valid** if it is true in all $\mathcal{L}$-structures;
- $\mathbf{S} \models \mathcal{S}$ means every sentence $F$ in $\mathcal{S}$ is true in $\mathbf{S}$;
- $\text{Sat}(\mathcal{S})$ means $\mathcal{S}$ is satisfiable;
- $\mathcal{S} \models F$ means every model of $\mathcal{S}$ is a model of $F$; If this is the case, we say $F$ is a **consequence of** $\mathcal{S}$.

# The Propositional Skeleton of a Formula

- The **propositional skeleton**, $\text{Skel}(F)$, of a formula $F$ is defined as follows:
  - Delete all quantifiers and terms;
  - Replace $\approx$ with $1$;
  - Replace the relation symbols $r$ with propositional variables $R$;
- Example: The formula

$$F = \forall x \forall y(\neg(x < y) \leftrightarrow \exists z((x < z) \vee (fz \approx y)))$$

has

$$\text{Skel}(F) = \neg P \leftrightarrow P \vee 1.$$

# The Propositional Skeleton Criterion

**Theorem**

The first-order formula $F$ has a one-element model iff $\text{Skel}(F)$ is satisfiable.

- If $\text{Skel}(F)$ is satisfiable, then choose an evaluation $\mathbf{e}$ that makes it true in a model $\mathbf{S}$ with universe $S = \{a\}$, as follows:
  - Let $f^{\mathbf{S}}(a, \ldots, a) = a$ for $f \in \mathcal{F}$;
  - Let $r^{\mathbf{S}}(a, \ldots, a)$ hold iff $\mathbf{e}(R) = 1$ for $r \in \mathcal{R}$;
- Example: $F = \forall x \forall y(\neg(x < y) \leftrightarrow \exists z((x < z) \vee (fz \approx y)))$;
  - We obtained $\text{Skel}(F) = \neg P \leftrightarrow P \vee 1$;
  - This is satisfiable if $P$ is evaluated as 0;
  - $F$ has the one-element model $\mathbf{S} = (\{a\}, f, <)$, where

$fa = a$,

| $<$ | $a$ |
|---|---|
| $a$ | 0 |

Subsection 10

# Equivalent Formulas

# Equivalent Sentences

- The sentences $F$ and $G$ are **equivalent**, written $F \sim G$, if they are true in the same $\mathcal{L}$-structures $\mathbf{S}$, that is, for all structures $\mathbf{S}$, we have

$$\mathbf{S} \models F \quad \text{iff} \quad \mathbf{S} \models G.$$

- For example, the sentences

$$\forall x(\neg(x \approx 0) \rightarrow \exists y(x \cdot y \approx 1)) \quad \text{and} \quad \forall x \exists y(\neg(x \approx 0) \rightarrow (x \cdot y \approx 1))$$

are equivalent.

**Theorem**

The sentences $F$ and $G$ are equivalent iff $F \leftrightarrow G$ is a valid sentence.

# Equivalent Formulas

- Two formulas $F(x_1, \ldots, x_n)$ and $G(x_1, \ldots, x_n)$ are **equivalent**, written $F(x_1, \ldots, x_n) \sim G(x_1, \ldots, x_n)$, iff $F$ and $G$ define the same relation on any $\mathcal{L}$-structure **S**, that is, $F^{\mathbf{S}} = G^{\mathbf{S}}$;
- For example, the following formulas are equivalent

$$\neg(x \approx 0) \to \exists y(x \cdot y \approx 1) \quad \text{and} \quad \exists y(\neg(x \approx 0) \to (x \cdot y \approx 1)).$$

**Proposition**

The formulas $F(\vec{x})$ and $G(\vec{x})$ are equivalent iff $\forall \vec{x}(F(\vec{x}) \leftrightarrow G(\vec{x}))$ is a valid sentence.

**Proposition**

The relation $\sim$ is an equivalence relation on sentences as well as on formulas.

- This is immediate from the definition of $\sim$ and the fact that ordinary equality $(=)$ is an equivalence relation.

# Fundamental Equivalences

- The following are some fundamental Equivalences of Formulas:
  1. $\neg\exists xF \sim \forall x(\neg F)$;
  2. $\neg\forall xF \sim \exists x(\neg F)$;
  3. $(\forall xF) \vee G \sim \forall x(F \vee G)$ if $x$ is not free in $G$;
  4. $(\exists xF) \vee G \sim \exists x(F \vee G)$ if $x$ is not free in $G$;
  5. $(\forall xF) \wedge G \sim \forall x(F \wedge G)$ if $x$ is not free in $G$;
  6. $(\exists xF) \wedge G \sim \exists x(F \wedge G)$ if $x$ is not free in $G$;
  7. $(\forall xF) \rightarrow G \sim \exists x(F \rightarrow G)$ if $x$ is not free in $G$;
  8. $(\exists xF) \rightarrow G \sim \forall x(F \rightarrow G)$ if $x$ is not free in $G$;
  9. $F \rightarrow (\forall xG) \sim \forall x(F \rightarrow G)$ if $x$ is not free in $F$;
  10. $F \rightarrow (\exists xG) \sim \exists x(F \rightarrow G)$ if $x$ is not free in $F$;
  11. $\forall x(F \wedge G) \sim (\forall xF) \wedge (\forall xG)$
  12. $\exists x(F \vee G) \sim (\exists xF) \vee (\exists xG)$

# Important Remarks on Freeness

- If $x$ occurs free in $G$ then we cannot conclude

$$(\forall x F) \vee G \sim \forall x(F \vee G);$$

- for example,

$$(\forall x(x < 0)) \vee (0 < x) \quad \text{and} \quad \forall x((x < 0) \vee (0 < x))$$

  are not equivalent; This can be seen by considering the natural numbers $\mathbb{N}$: in $\mathbb{N}$, the first is true of positive numbers $x$ (Note that $x$ occurs free in this formula);
  whereas the second is false (Note that there are no free occurrences of $x$ in this formula);

# Some Other Remarks

- For the implication we have:

$$\begin{aligned}(\forall xF) \to G &\sim \neg(\forall xF) \vee G\\ &\sim \exists x(\neg F) \vee G\\ &\sim \exists x(\neg F \vee G)\\ &\sim \exists x(F \to G).\end{aligned}$$

- To see that

$$\forall x(F \vee G) \sim (\forall xF) \vee (\forall xG)$$

need not be true consider the following example:

$$\forall x((0 \approx x) \vee (0 < x)) \quad \text{and} \quad (\forall x(0 \approx x)) \vee (\forall x(0 < x)).$$

- And to see that

$$\exists x(F \wedge G) \sim (\exists xF) \wedge (\exists xG)$$

need not be true consider the example:

$$\exists x((0 \approx x) \wedge (0 < x)) \quad \text{and} \quad (\exists x(0 \approx x)) \wedge (\exists x(0 < x)).$$

Subsection 11

# Replacement and Substitution

# Substitution of Formulas for Propositional Variables

- Equivalent propositional formulas lead to equivalent first-order formulas as follows:

## Proposition

If $F(P_1, \ldots, P_n)$ and $G(P_1, \ldots, P_n)$ are equivalent propositional formulas, then for any sequence $H_1, \ldots, H_n$ of first-order formulas we have $F(H_1, \ldots, H_n) \sim G(H_1, \ldots, H_n)$.

- Example: De Morgan's Law gives the equivalence of the two propositional formulas $\neg(P \wedge Q) \sim \neg P \vee \neg Q$. By the Proposition above, then, the following first-order formulas are also equivalent:

$$\neg((\exists x(x \cdot x \approx 1)) \wedge (\forall x \forall y(x \cdot y \approx y \cdot x))) \sim \neg(\exists x(x \cdot x \approx 1)) \vee \neg(\forall x \forall y(x \cdot y \approx y \cdot x))$$

# Compatibility of Equivalence with Connectives

- Applying logical connectives preserves equivalence;
- This property of equivalence is called **compatibility with the logical connectives**;

## Compatibility Lemma

Suppose $F_1 \sim G_1$ and $F_2 \sim G_2$. Then

1. $\neg F_1 \sim \neg G_1$;
2. $F_1 \vee F_2 \sim G_1 \vee G_2$;
3. $F_1 \wedge F_2 \sim G_1 \wedge G_2$;
4. $F_1 \rightarrow F_2 \sim G_1 \rightarrow G_2$;
5. $F_1 \leftrightarrow F_2 \sim G_1 \leftrightarrow G_2$;
6. $\forall x F_1 \sim \forall x G_1$;
7. $\exists x F_1 \sim \exists x G_1$.

# Replacement in First-Order Logic

- The replacement theorem says that, in a first-order formula the replacement of a subformula by an equivalent formula results in a equivalent formula; More formally:

**Replacement Theorem**

If $F \sim G$ then $H(\cdots F \cdots) \sim H(\cdots G \cdots)$.

- Example: We have that

$$\neg((\exists x(x \cdot x \approx 1)) \wedge (\forall x \forall y(x \cdot y \approx y \cdot x))) \sim \neg(\exists x(x \cdot x \approx 1)) \vee \neg(\forall x \forall y(x \cdot y \approx y \cdot x))$$

Therefore, by the Replacement Theorem

$$(\forall x \exists y(x < y)) \rightarrow \neg((\exists x(x \cdot x \approx 1)) \wedge (\forall x \forall y(x \cdot y \approx y \cdot x))) \sim (\forall x \exists y(x < y)) \rightarrow \neg(\exists x(x \cdot x \approx 1)) \vee \neg(\forall x \forall y(x \cdot y \approx y \cdot x))$$

# Substitution of Terms for Variables

- Substitution of terms for variables in first-order logic often requires the need to rename variables;
- We need to be careful with renaming variables to avoid binding any newly introduced occurrences of variables;
- Given a first-order formula $F$, define a **conjugate of** $F$ to be any formula $\bar{F}$ obtained by renaming the occurrences of bound variables of $F$ so that no free occurrences of variables in $F$ become bound; When renaming, we must keep bound occurrences of distinct variables distinct;

## Equivalence of Conjugates

If $\bar{F}$ is a conjugate of $F$, then $\bar{F} \sim F$.

- Example: $\exists y(x \cdot y \approx 1) \sim \exists w(x \cdot w \approx 1)$.

# The Substitution Theorem

**Substitution Theorem**

If $F(x_1,\ldots,x_n) \sim G(x_1,\ldots,x_n)$ and $t_1,\ldots,t_n$ are terms, then $F^\star(t_1,\ldots,t_n) \sim G^\star(t_1,\ldots,t_n)$.

- For instance, since $\neg\exists y(x \cdot y \approx 1) \sim \forall y(\neg(x \cdot y \approx 1))$, substitution of $(y+w)$ for $x$ and $u$ for $y$ yields

$$\neg\exists u((y+w) \cdot u \approx 1) \sim \forall u(\neg((y+w) \cdot u \approx 1)).$$

Subsection 12

# Prenex Form

# Prenex Form

- A formula $F$ is in **prenex form** if it looks like
$$Q_1x_1 \cdots Q_nx_nG,$$
where
  - the $Q_i$ are quantifiers;
  - $G$ has no occurrences of quantifiers;
- A formula with no occurrences of quantifiers is called an **open formula**;
- The formula
$$\exists x((rxy) \rightarrow \forall u(ruy))$$
is not in prenex form, but it is equivalent to the prenex form formula
$$\exists x\forall u((rxy) \rightarrow (ruy)).$$

**Prenex Form Theorem**

Every formula is equivalent to a formula in prenex form.

# Obtaining an Equivalent Formula in Prenex Form

- The following steps put $F$ in prenex form:
  1. Rename the quantified variables so that distinct occurrences of quantifiers bind distinct variables, and no free variable is equal to a bound variable;
     Example: Change
     $$\forall z((rzy) \to \neg\forall y((rxy) \wedge \exists y(ryx)))$$
     to
     $$\forall z((rzy) \to \neg\forall u((rxu) \wedge \exists w(rwx)))$$
  2. Eliminate all occurrences of $\to$ and $\leftrightarrow$ using
     - $G \to H \sim \neg G \vee H$;
     - $G \leftrightarrow H \sim (\neg G \vee H) \wedge (\neg H \vee G)$;

     Example (Cont'd): The equivalent form is
     $$\forall z(\neg(rzy) \vee \neg\forall u((rxu) \wedge \exists w(rwx)));$$
  3. Pull the quantifiers to the front;

# Obtaining an Equivalent Formula in Prenex Form (Cont'd)

- This can be accomplished by using the equivalences:
  - $\neg(F \vee G) \sim (\neg F \wedge \neg G)$
  - $\neg(F \wedge G) \sim (\neg F \vee \neg G)$
  - $G \vee (\forall x H) \sim \forall x(G \vee H)$
  - $G \vee (\exists x H) \sim \exists x(G \vee H)$
  - $G \wedge (\forall x H) \sim \forall x(G \wedge H)$
  - $G \wedge (\exists x H) \sim \exists x(G \wedge H)$
  - $(\forall x G) \vee H \sim \forall x(G \vee H)$
  - $(\exists x G) \vee H \sim \exists x(G \vee H)$
  - $(\forall x G) \wedge H \sim \forall x(G \wedge H)$
  - $(\exists x G) \wedge H \sim \exists x(G \wedge H)$
  - $\neg \exists x G \sim \forall x \neg G$
  - $\neg \forall x G \sim \exists x \neg G$

# Example Continued

- Applying some of the equivalences of the previous slide, we get

$$\forall z(\neg(rzy) \vee \neg\forall u((rxu) \wedge \exists w(rwx)))$$
$$\downarrow$$
$$\forall z(\neg(rzy) \vee \exists u(\neg((rxu) \wedge \exists w(rwx))))$$
$$\downarrow$$
$$\forall z\exists u(\neg(rzy) \vee \neg((rxu) \wedge \exists w(rwx)))$$
$$\downarrow$$
$$\forall z\exists u(\neg(rzy) \vee (\neg(rxu) \vee \neg(\exists w(rwx))))$$
$$\downarrow$$
$$\forall z\exists u(\neg(rzy) \vee (\neg(rxu) \vee \forall w(\neg(rwx))))$$
$$\downarrow$$
$$\forall z\exists u(\neg(rzy) \vee \forall w(\neg(rxu) \vee (\neg(rwx))))$$
$$\downarrow$$
$$\forall z\exists u\forall w(\neg(rzy) \vee (\neg(rxu) \vee (\neg(rwx))))$$

Subsection 13

## Valid Arguments

# Valid or Correct Arguments

- We will be working with sentences in a fixed first-order language $\mathcal{L}$;
- An argument $F_1, \ldots, F_n \therefore F$ is **valid** (or **correct**) in first-order logic provided every structure $\mathbf{S}$ that makes $F_1, \ldots, F_n$ true also makes $F$ true, i.e., for every $\mathcal{L}$-structure $\mathbf{S}$,

$$\mathbf{S} \models \{F_1, \ldots, F_n\} \quad \text{implies} \quad \mathbf{S} \models F.$$

## Proposition

An argument $F_1, \ldots, F_n \therefore F$ in first-order logic is valid iff

$$F_1 \wedge \cdots \wedge F_n \rightarrow F$$

is a valid sentence; Moreover, this holds iff $\{F_1, \ldots, F_n, \neg F\}$ is not satisfiable.

# Some Examples Involving Equations

- In first-order logic equations are treated as universally quantified sentences:
$$\forall\vec{x}(s(\vec{x}) \approx t(\vec{x}));$$
- The following argument is valid
$$\begin{array}{l} \forall x \forall y \forall u \forall v(x \cdot y \approx u \cdot v) \\ \therefore\ \forall x \forall y \forall z((x \cdot y) \cdot z \approx x \cdot (y \cdot z)) \end{array}$$
- In fact, if a structure **S** satisfies the premiss then all multiplications give the same value. Thus, the multiplication must be associative.
- The argument $\begin{array}{l} \exists y \forall x(rxy) \\ \therefore\ \forall x \exists y(rxy) \end{array}$ is valid;
- To see this, suppose **S** is a structure satisfying the premiss. Then, for some $a \in S$, $\forall x(rxa)$ holds. Thus, $\forall x \exists y(rxy)$ also holds.
- We have demonstrated the validity of the above arguments by appealing to our reasoning skills in mathematics.

# Proving Non-Validity of Arguments

- To show that an argument $F_1, \ldots, F_n \therefore F$ is not valid it suffices to find a single structure **S** such that
  - each of the premisses $F_1, \ldots, F_n$ is true in **S**, but
  - the conclusion $F$ is false in **S**.

  Such a structure **S** is called a **counterexample** to the argument.
- Example: The argument $\begin{array}{l} \forall x \exists y (rxy) \\ \therefore\ \exists y \forall x (rxy) \end{array}$ is not valid;

  A simple two-element graph gives a counterexample:


(Let us verify this!)

Subsection 14

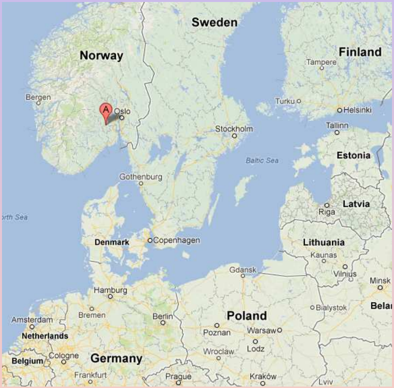
## Skolemization

# Leopold Löwenheim

- Leopold Löwenheim, born in Krefeld, Germany (1878-1957)

# Thoralf Albert Skolem

- Thoralf Albert Skolem, born in Sandsvær, Norway (1887-1963)



# Skolemization: The Intuition

- Skolem, following the work of Löwenheim (1915), developed a technique to convert a first-order sentence $F$ into a sentence $F'$ in prenex form, with only universal quantifiers, such that

$$F \text{ is satisfiable iff } F' \text{ is satisfiable.}$$

- Universally quantified sentences are apparently much easier to understand.
- This has provided one of the powerful techniques in automated theorem proving.

# Skolemization: The Main Lemma

## Skolemization Lemma

1. Given the sentence $\exists y G(y)$, augment the language with a new constant $c$ and form the sentence $G(c)$. Then

$$\text{Sat}(\exists y G(y)) \quad \text{iff} \quad \text{Sat}(G(c));$$

2. Given the sentence $\forall x_1 \cdots \forall x_n \exists y G(\vec{x}, y)$, augment the language with a new $n$-ary function symbol $f$ and form the sentence $\forall x_1 \cdots \forall x_n G^\star(\vec{x}, f(\vec{x}))$. Then

$$\text{Sat}(\forall x_1 \cdots \forall x_n \exists y G(\vec{x}, y)) \quad \text{iff} \quad \text{Sat}(\forall x_1 \cdots \forall x_n G^\star(\vec{x}, f(\vec{x}))).$$

# Universal Formulas

- A first-order formula $F$ is **universal** if it is in prenex form and all quantifiers are universal, that is, $F$ is of the form $\forall\vec{x}G$, where $G$ is quantifier-free;
- $G$ is called the **matrix** of F;
- Example:

$$\forall x\forall y\forall z\underbrace{((x\leq y)\wedge(y\leq z)\rightarrow(x\leq z))}_{\text{matrix}}.$$

# Producing an Equivalent Universal Sentence

## Universal Equivalent of a Sentence

Given a first-order sentence $F$, there is an effective procedure for finding a universal sentence $F'$ (usually in an extended language) such that

$$\text{Sat}(F) \quad \text{iff} \quad \text{Sat}(F').$$

Furthermore, we can choose $F'$ such that every model of $F$ can be expanded to a model of $F'$, and every model of $F'$ can be reducted to a model of $F$.

- To produce $F'$, given $F$,
  - first, we put $F$ in prenex form;
  - then, we just apply the Skolemization Lemma repeatedly until there are no existential quantifiers.
- This process is called **skolemizing**;
- The newly introduced constants and functions are called **skolem constants** and **skolem functions**.

# Example of Skolemization

- We skolemize the sentence

$$F = \forall x \forall y((x < y) \to \exists z((x < z) \land (z < y)))$$

- First put it in prenex form

$$F \sim \forall x \forall y \exists z((x < y) \to (x < z) \land (z < y))$$

- Applying the Skolemization Lemma, we introduce a new binary function symbol, say $f$, and arrive at the universal sentence

$$F' = \forall x \forall y((x < y) \to (x < f(x, y)) \land (f(x, y) < y))$$

- The structure $\mathbb{Q} = (\mathbb{Q}, <)$, consisting of the rational numbers with the usual $<$, satisfies $F$; If we choose $f(a, b) = \frac{a+b}{2}$, for $a, b \in \mathbb{Q}$, we see that the expansion $(\mathbb{Q}, <, f)$ of $\mathbb{Q}$ satisfies $F'$.

# Equivalent Universal Set of Sentences

## Universal Equivalent of Sets of Sentences

Given a set of first-order sentences $\mathcal{S}$, there is a set $\mathcal{S}'$ of universal sentences (usually in an extended language) such that

$$\text{Sat}(\mathcal{S}) \quad \text{iff} \quad \text{Sat}(\mathcal{S}').$$

Furthermore, every model of $\mathcal{S}$ can be expanded to a model of $\mathcal{S}'$, and every model of $\mathcal{S}'$ can be reducted to a model of $\mathcal{S}$.

- To obtain $\mathcal{S}'$, given $\mathcal{S}$, we skolemize each sentence in S, as before, making sure that distinct sentences do not have any common skolem constants or functions.

# An Example of Skolemization of a Set of Sentences

- Example: We skolemize the set of sentences

$$\{\exists x \forall y \exists z (x < y + z), \exists x \forall y \exists z (\neg (x < y + z))\};$$

we obtain a set of universal sentences

$$\{\forall y (a < y + fy), \forall y (\neg (b < y + gy))\}.$$