Logical systems (part 2): First-order logic

Tiago de Lima

March 8, 2023
(version 0.4.2)

Contents

Chapter 1

Language

1.1 Motivation

Some arguments cannot be expressed using propositional logic:

Example 1.

John is ill.
Someone is ill.

The translation of this argument into propositional language is:

$$P_1$$
$$P_2$$

P_2 is not a logical consequence of P_1.

However, the argument is clearly valid.

The problem is that the internal structure of these sentences cannot be captured by the formulas P_1 and P_2.

It is the similarity of their internal subject-predicate structure that makes the argument valid.

To capture this, we need to enrich the propositional language with symbols capable of expressing:

- subjects, such as ‘John’ and ’Someone’; and
- predicates, such as ‘ill’.

In the **propositional logic**, we can analyse arguments containing sentences connected by ‘and’, ‘or’, etc..

In the **first-order logic**, we can analyse arguments such that the validity depends on the internal subject-predicate structure of sentences.

1.2 Quantifiers

Below, we list some examples of sentences with a similar internal structure:

Group 1:

Each of the numbers 2, 4 and 6 is even.
All natural numbers are positive.
All natural numbers are negative.
All humans are mortal.

All propositions in group 1 have the same form:

all objects (of a certain type) have the property P

or, equivalently:

for each object x, x has the property P

For the sentences of the group 1, we use the following notation:

$$\forall x[P(x)]$$

We have:

- $P(x)$ means 'x has the property P'.
- x is an **individual variable**, i.e., it ranges over the domain of individuals (or objects).

  It is also possible to use a **constant** in the place of x:

  - E.g.: $P(Socrates)$ means 'Socrates has the property P'.
- $\forall x$ means 'for each object x'.

  The symbol $\forall$ is called **universal quantifier**.

  In some (old) books, this is also noted (x) or Λx.

Note that $\forall$ is analogous to a conjunction.

- E.g.: 'Each of the numbers 2, 4 and 6 is even' = '2 is even and 4 is even and 6 is even'.
- E.g.: 'All natural numbers are positive' = '0 is positive and 1 is positive and 2 is positive . . . '. (an infinite conjunction).

Group 2:

At least one of the numbers 2, 3 and 4 is even.
There is some natural number x such that $x > 0$.
Some humans are mortal.

All propositions in group 2 have the same form:

some objects (of a certain type) have the property P

or, equivalently:

there is at least one object x such that x has the property P

For the sentences of the group 2, we use the following notation:

$$\exists x[P(x)]$$

The symbol $\exists$ is called **existential quantifier**.

In some (old) books, this is also noted Vx.

Note that $\exists$ is analogous to a disjunction:

- E.g.: 'At least one of the numbers 2, 3, and 4 is even' = 2 is even or 3 is even or 4 is even ....

The predicate of an atomic proposition can be composed using simpler predicates by using expressions, such as 'not', 'and', 'or', 'if ... then ... ' and 'if and only if'.

Examples 2.

For each natural number x, x is even if and only if x^2 is even.	$\forall x[P(x) \leftrightarrow Q(x)]$
All animals having four legs are cows.	$\forall x[P(x) \rightarrow Q(x)]$
Some natural numbers are positive and even.	$\exists x[P(x) \land Q(x)]$
All natural numbers are positive or negative.	$\forall x[P(x) \lor Q(x)]$
There is some natural number x such that x is not greater than 0.	$\exists x[\neg P(x)]$

Exercise 1.1. *Explain the difference between these two formulas:*

$$\forall x[P(x) \rightarrow Q(x)]$$
$$\exists x[P(x) \rightarrow Q(x)]$$

An atomic proposition can contain more than one quantifier.

Examples 3.

All natural numbers are equal.
(For all natural number x and all natural number y, $x = y$.) $\forall x \forall y[P(x, y)]$

There are different natural numbers.
(There is a natural number x and a natural number y such that $x \neq y$): $\exists x \exists y[P(x, y)]$

$P(a,b)$ means that 'a is in the relation P with b'.

It is also possible to use constants, such as in $P(Janet, Peter)$.
Group 3:

Every person has a mother.
(For every person x, there is some person y such that y is the mother of x.)

For every natural number, there is a greater one.
(For every natural number x, there is some natural number y such that $y > x$.)

All propositions in group 3 are of the form:

for all object x, there is some object y (possibly depending of x) such that x is in the relation P with y

For the sentences in group 3, we use the following notation:

$$\forall x \exists y [P(x,y)]$$

Group 4:

Someone is the mother of all persons.
(There is some person y such that for all person x, x is the mother of y.)

For every natural number, there is a greater one.
There is some natural number y such that for all natural number x, $x < y$.)

All propositions in group 4 are of the form:

there exists an object y such that for all objects x (including y), x is in the relation with y

For the sentences in group 4, we use the following notation:

$$\exists y \forall x [P(x,y)]$$

When the quantifiers are different, their order of appearance is important:

Example 4.

$x \backslash y$	1	2	3
1	$\times$		
2			$\times$
3		$\times$	

$$\forall x \exists y [P(x, y)]$$

$$P(1,1) \; P(2,3) \; P(3,2)$$

$x \backslash y$	1	2	3
1		$\times$	
2		$\times$	
3		$\times$	

$$\exists y \forall x [P(x, y)]$$

$$P(1,2) \; P(2,2) \; P(3,2)$$

Nonetheless $\forall x \forall y [P(x, y)]$ and $\forall y \forall x [P(x, y)]$ are equivalent.

Similarly for $\exists x \exists y [P(x, y)]$ and $\exists y \exists x [P(x, y)]$.

Group 5:

Socrates is human.
Socrates is mortal.
3 is odd.
4 is even.

All propositions in group 5 are of the form:

c has the property P

For the sentences in group 5, we use the following notation:

$$P(c)$$

The letter c is a **constant**.

Different objects must be identified with different constants.

By using the atomic propositions seen before, we can compose more complex prospositions with the connectives of propositional logic.

Examples 5.

If all natural numbers are even, then all natural numbers are odd.
$\forall x[P(x)] \to \forall x[Q(x)]$

If there is an even natural number, than there is a natural number which is equal to itself.
$\exists x[P(x)] \to \exists x[Q(x)]$

1.3 Natural language vs. first-order language

The sentences below have the same structure (noun phrase, verb phrase), but their translation into first-order language do not have the same structure

John walks.	$W(j)$
Every student walks.	$\forall x[S(x) \to W(x)]$
Some student walks.	$\exists x[S(x) \land W(x)]$
No student walks.	$\neg\exists x[S(x) \land W(x)]$
Somebody walks.	$\exists x[W(x)]$
Nobody walks.	$\neg\exists x[W(x)]$

The sentences above have been translated into a formal first-order language such that the alphabet consists of the following symbols:

symbol	intended interpretation
j	John
W	to walk
S	being a student
x	person
$\neg$, $\to$, $\land$, $\lor$, $\neg$	connectives
$\forall$, $\exists$	quantifiers

The interpretation of connectives and quantifiers has been fixed. These symbols are called **logical symbols**.

The interpretation of j, x, W and S can vary. These symbols are called **non-logical symbols**. For example, one could use the following non-intended interpretation:

symbol	non-intended interpretation
j	0
W	is even
S	is odd
x	natural number

In this case we would have:

formula	interpretation
$W(j)$	0 is even.
$\forall x[S(x) \to W(x)]$	Every odd natural number is even.
$\exists x[S(x) \land W(x)]$	Some natural number is both odd and even.
$\neg\exists x[S(x) \land W(x)]$	No natural number is both odd and even.
$\exists x[W(x)]$	Some natural number is even.
$\neg\exists x[W(x)]$	No natural number is even.

The translation of the argument:

If every student walks and John is a student, then John walks.

into propositional language would be a formula of the form:

$$(P \wedge Q) \rightarrow R$$

This is not a valid formula.

The translation of this argument into the first-order language specified above is:

$$(\forall x[S(x) \wedge W(x)] \wedge S(j)) \rightarrow W(j)$$

This formula is a valid under every possible interpretation (intended or not).

This is is guaranteed by the fixed meaning of the logical symbols in the formula.
Some more examples of valid formulas follows:

$$\forall x[S(x) \rightarrow S(x)]$$
$$\forall x[\neg(W(x) \wedge \neg W(x))]$$
$$(\forall x[S(x) \rightarrow W(x)] \wedge \forall x[S(x)]) \rightarrow \forall x[W(x)]$$

1.4 Functional propositions

Consider the sentence:

Plato is a philosopher.

In grammar, we call 'Plato' the subject and 'is a philosopher' the predicate.

In logic, we use the term 'predicate' in a more general way than in grammar.

Consider the sentence:

John loves Jane.

In grammar, we call 'John' the subject, 'loves Jane' the predicate, where 'Jane' is the object of the sentence.
In logic:

- We can call 'John' the **subject** and 'loves Jane' the **predicate** of the sentence.
- We can also call both 'John' and 'Jane' **subjects** and 'loves' the **predicate** of the sentence. (This is a predicate with two arguments.)
- We can also call 'Jane' the **subject** and 'John loves' the **predicate** of the sentence.

Instead of predicate, one can also use the term **propositional function**.

- 'a loves Jane' assigns a truth value for each value of a.
- 'John loves b' assigns a truth value for each value of b.
- 'a loves b' assigns a truth value for each value of a and b.

A predicate with one argument is also called **property**.

A predicate with two arguments is also called **binary relation**.

A predicate with three arguments is also called **ternary relation**.

1.5 Scope, bound and free variables

Scope

In the expression:

$$\forall x \exists y [P(x,y) \to \exists z [Q(y,z)]] \to \forall x [\neg P(x,a)]$$

the scope of the first occurrence of $\forall x$ is the sub-expression:

$$\exists y [P(x,y) \to \exists z [Q(y,z)]]$$

The scope of $\exists y$ is the sub-expression:

$$P(x,y) \to \exists z [Q(y,z)]$$

The scope of $\exists z$ is the sub-expression:

$$Q(y,z)$$

Similarly, in $\neg A$, $A \leftrightarrow B$, $A \to B$, $A \wedge B$ et $A \vee B$, the expression A or the pair of expressions A et B is the scope of the propositional connective in question.

Bound and free variables

Definition 6. An occurrence of a variable x is said to be **bound**, if the occurrence is in a quantifier or in the scope of a quantifier with the same x.

An occurrence of a variable x is said to be **free** if it is not bound.

Examples 7.

- In $\exists x[M(a,b)]$, the occurrences of a and b are free.
- In $\exists x[M(a,x)]$, the occurrence of a is free and both occurrences of x are bound.
- In $\forall x \exists y [M(x,y)]$, both occurrences of x and y are bound.

Definition 8. A variable a which occurs free in A is called **a free variable of** A. Likewise for bound variables.

Example 9. The free variables of $\exists x \forall y [(x < a) \wedge (y < b)]$ are a and b, and the bound variables are x and y.

1.6 Well-formed formulas

Until now, we were using particular first-order languages:

- E.g. a language to express property about students; and another to express some properties of natural numbers.

Each particular first-order language contains the following elements, which constitutes its **alphabet**:

$a_1, a_2, a_3, \ldots$	free individual variables
$x_1, x_2, x_3, \ldots$	bound individual variables
$\leftrightarrow, \rightarrow, \wedge, \vee, \neg$	connectives
$(,), [,]$	parentheses
$c_1, c_2, c_3, \ldots$	individual constants
$P_1, P_2, P_3, \ldots$	predicate symbols (each P_i is n_i-ary)

Note that an 0-ary predicate symbol is, in fact, a proposition.

Signature

The non-logical part of the alphabet is also called **signature**.

It is a pair $\mathcal{S} = (\mathcal{F}, \mathcal{R})$, where:

- $\mathcal{F}$ is a set of constants.
- $\mathcal{R}$ is a set of pairs (P, n), where P is a predicate symbol and n is its arity.

Well-formed formulas

Definition 10 (Atomic formula)**.** If P is an n-ary predicate symbol and each one of $r_1, \ldots, r_n$ is a free individual variable or an individual constant, then $P(r_1, \ldots, r_n)$ is an atomic formula.

Definition 11 (Formula)**.** The set of formulas is recursively defined as follows:

1. Each atomic formula is a formula.
2. If φ is a formula, then $\neg\varphi$ is a formula.
3. If φ and ψ are formulas, then $\varphi \leftrightarrow \psi$, $\varphi \rightarrow \psi$, $\varphi \wedge \psi$, and $\varphi \vee \psi$ are formulas.
4. If $\varphi(a)$ is a formula in which the free variable a occurs, and x is any bound variable not occurring in $\varphi(a)$, then $\forall x[\varphi(x)]$ and $\exists x[\varphi(x)]$ are formulas, where $\varphi(x)$ results from $\varphi(a)$ by replacing every occurrence of a in $\varphi(a)$ by x.
5. The only formulas are those given by 1, 2, 3 and 4 above.

Examples 12. Assume that P_1 is a unary (i.e., $n_1 = 1$) and P_2 is a binary (i.e., $n_2 = 2$) predicate.

Atomic formulas: $P_1(a_1)$, $P_1(a_7)$, $P_1(c_1)$, $P_1(c_5)$, $P_2(a_1, a_2)$, $P_2(a_1, c_2)$, $P_2(c_1, a_2)$, $P_2(c_1, c_2)$.

Non-atomic formulas: $\forall x_1[P_1(x_1)]$, $\forall x_2[P_1(x_2)]$, $\forall x_1[P_2(x_1, a_2)]$, $\forall x_2[P_2(x_2, a_2)]$.

Definition 13 (Sentence)**.** A formula A is called **closed** if it contains no free variables; otherwise it is called **open**. A closed formula is also called **sentence**.

Examples 14. Assume $n_2 = 2$.

Open formulas:

- $P_2(a_1, a_2)$
- $\exists x_x[P_2(a_1, x_2)]$

Closed formulas:

- $P_2(c_1, c_2)$
- $\exists x_2[P_2(c_1, c_2)]$
- $\forall x_1 \exists x_2[P_2(x_1, x_2)]$

Theorem 15 (Structural induction)**.** *Let Φ be a property of formulas, such that:*

1. *all atomic formulas have the property Φ,*
2. *if φ has the property Φ, then $\neg\varphi$ has the property Φ.*
3. *if φ and ψ have the property Φ, then $\varphi \leftrightarrow \psi$, $\varphi \to \psi$, $\varphi \wedge \psi$ and $\varphi \vee \psi$, also have the property Φ, and*
4. *if $\varphi(a)$ has the property Φ, x does not occur in $\varphi(a)$ and $\varphi(x)$ results from $\varphi(a)$ by replacing all occurrences of a in $\varphi(a)$ by x, then $\forall x[\varphi(x)]$ and $\exists x[\varphi(x)]$ also have the property Φ.*

Then all formulas have the property Φ.

1.7 Exercises

Exercise 1.2. Let $H(a)$ be 'a is human' and $M(a)$ be 'a is mortal'.

1. Translate each of the following sentences into the first-order language in an adequate way.
   (a) Every human is mortal.
   (b) Some human is mortal.
2. Explain why $\forall x[H(x) \wedge M(x)]$ is not an adequate translation for the first sentence.
3. Explain why $\exists x[H(x) \to M(x)]$ is not an adequate translation for the second sentence.

Exercise 1.3.

1. Verify that 'there is some natural number x such that if x is even, then $x \neq x$' is a true proposition of the form $\exists x[A(x) \to B(x)]$.
2. Verify that 'for all natural numbers x, if x is even, then $x \neq x$' is a false proposition of the form $\forall x[A(x) \to B(x)]$.
3. Verify that 'if there is some natural number x such that x is even, then there is some natural number x such that $x \neq x$' is a false proposition of the form $\exists x[A(x)] \to \exists x[B(x)]$.
4. Verify that 'if all natural numbers are even, then all natural numbers are odd' is a true proposition of the form $\forall x[A(x)] \to \forall x[B(x)]$.

Exercise 1.4. Translate the following formulas into sentences in natural language:

1. $\forall x \forall y [P(x, y)]$.
2. $\forall x \exists y [P(x, y)]$.
3. $\exists x \forall y [P(x, y)]$.
4. $\exists x \exists y [P(x, y)]$.

Exercise 1.5. Let $A(a, b)$ be 'a admires b'. Translate the following two sentences into the first-order language.[1]

1. Everyone has someone whom one admires.
   (*Toute personne a une personne qu'elle admire.*)
2. There is someone whom everyone admires.
   (*Il y a une personne admirée par toute personne.*)

Exercise 1.6. Let:
$D(a)$ be 'a is a Dutchman',
$C(a)$ be 'a is a kind of cheese,
$W(a)$ be 'a is a kind of wine,
$L(a, b)$ be 'a likes b,
c be 'Chip' and
d be 'Donald'.
Translate each of the following sentences into the firs-order language in an adequate way.

1. Donald likes all kinds of cheese.
2. Some Dutchmen like all kinds of cheese.
3. Donald likes some kinds of cheese.
4. All Dutchmen like at least one kind of cheese.
5. There is a kind of cheese which is liked by any Dutchman.
6. Chip doesn't like any kind of cheese.
7. All Dutchmen don't like any kind of cheese.
8. All Dutchmen like some kind of cheese and some kind of wine.
9. All Dutchmen who like some kind of cheese, also like some kind of wine.
10. If all Dutchmen like some kind of cheese, then all Dutchmen like some kind of wine.

Exercise 1.7. Let $A(a)$ be 'a has the property A', and $a = b$ be 'a equals b'. Translate each of the following sentences into the first-order language.

[1] The sentence 'Everyone admires someone' (*Toute personne admire une personne.*) is ambiguous and can have each of the two readings above.

1. There is at least one x which has the property A.
2. There is at most one x which has the property A.
3. There is exactly one x which has the property A.
4. There are at least two objects which have the property A.
5. There are at most two objects which have the property A.
6. There are exactly two objects which have the property A.

Exercise 1.8. Consider the first-order language which has the following non-logical symbols:

$\doteq$	binary predicate symbol
c_1, c_2	individual constants

1. Translate the sentences below into this language in an adequate way.
   (a) The Morning Star is the same as the Evening Star.
   (b) Every star identical to the Morning Star is the same as the Evening Star.
2. For the formulas found in 1, consider the following non-intended interpretation:

$\forall x$	for all natural numbers x
$\exists x$	there is some natural number x
$\doteq$	is equal to (=)
c_1	3
c_2	4

   Are the readings of the formulas found in 1 true or false under this interpretation?
3. Similar question as in 2 but now for the following non-intended interpretation:

$\forall x$	for all persons x
$\exists x$	there is some person x
$\doteq$	is older than
c_1	Reagan
c_2	Nixon

Exercise 1.9. Give adequate translations for the following expressions in an appropriately chosen (first-order) first-order language.[2]

1. A man walks in the park. He whistles.
2. Every farmer who owns a donkey, feeds it.

[2]The translation of these expressions, which contain anaphoric relations, are not obtained in a compositional way.

Chapter 2

Model theory

2.1 Introduction

The truth value of first-order formulas depends on the interpretation of its variables and predicates.

Examples 16.

interpretation			truth value	
x	$P(a)$	$Q(a)$	$\forall x[P(x)]$	$\exists x[Q(x)]$
$\mathbb{N}$	$x = x$	a is even	1	1
humans	a is mortal	a is immortal	1	0
$\mathbb{N}$	a is even	a is even	0	1
pets	a is a dog	a is immortal	0	0

2.2 Satisfiability, Truth and validity

Satisfiable formula

The previous table shows two interpretations under with $\forall x[P(x)]$ is true:

- 'every natural number is equal to itself'
- 'all humans are mortal'

The table also shows two interpretations under with the same formula is false:

- 'every natural number is even'
- 'all pets are dogs'

When a formula φ is true under some interpretation, we say that φ is **satisfiable**.

Valid formula

The sentence

'not all persons have red hair'

is equivalent to

'there is some person who does not have red hair'

That is, we have that

$$\neg\forall x[P(x)]$$

is equivalent to

$$\exists x[\neg P(x)]$$

This is always true, no matter how we interpret x and P. This means that the following formula is always true:

$$\neg\forall x[P(x)] \leftrightarrow \exists x[\neg P(x)]$$

When a formula φ is true under all interpretations, we say that φ is **valid**.

The notion of validity is based on the notion of truth, which has been made mathematically precise by Alfred Tarski in 1933.

Alfred Tarski (Jan, 14 1901 — Oct, 26 1983)
George Bergman, GFDL 1.2, via Wikimedia Commons

Structure

Definition 17 ($\mathcal{S}$-structure)**.** Let an alphabet with signature $\mathcal{S} = \langle \mathcal{F}, \mathcal{R} \rangle$ be given. A **$\mathcal{S}$-structure** is a tuple $\mathcal{M} = \langle \mathcal{D}, f, R \rangle$ such that:

- $\mathcal{D}$ is a non-empty set of objects called **domain**;
- f is a function such that, for each individual constant c of the alphabet, $f(c) \in \mathcal{D}$ is the interpretation of c.
- R is a function such that, for each n-ary predicate P of the alphabet, $R((P, n)) \subseteq \mathcal{D}^n$ is the interpretation of P.

Example 18. Consider the signature $\mathcal{S} = \langle \mathcal{F}, \mathcal{R} \rangle$ and the structure $\mathcal{M}_1 = \langle \mathcal{D}, f, R \rangle$, where:

- $\mathcal{F} = \{c\}$
- $\mathcal{R} = \{(P, 2)\}$

- $\mathcal{D} = \mathbb{N}$
- $f(c) = 2$
- $R((P, 2)) = \geq$

That is, the alphabet contains (in addition to the connectives, quantifiers, etc.):

- one constant c, whose interpretation is '2', and
- one predicate P of arity 2, whose interpretation is the relation '$\geq$'.

We have that, for example:

$$\forall x[P(x, c)] : \text{'all natural numbers are greater than or equal to 2'}$$

is false in $\mathcal{M}_1$.

Example 19. Consider the signature $\mathcal{S} = \langle \mathcal{F}, \mathcal{R} \rangle$, where:

- $\mathcal{F} = \{c\}$
- $\mathcal{R} = \{(P, 2)\}$

and the structure $\mathcal{M}_2 = \langle \mathcal{D}, f, R \rangle$, where:

- $\mathcal{D} = \mathbb{N}$
- $f(c) = 0$
- $R((P, 2)) = \geq$

We have that, for example:

$$\forall x[P(x, c)] : \text{'all natural numbers are greater than or equal to 0'}$$

is true in $\mathcal{M}_2$.

Example 20. Consider the signature $\mathcal{S} = \langle \mathcal{F}, \mathcal{R} \rangle$, where:

- $\mathcal{F} = \emptyset$
- $\mathcal{R} = \{(P, 2)\}$

and the structure $\mathcal{M}_3 = \langle \mathcal{D}, f, R \rangle$, where:

- $\mathcal{D} = \mathbb{N}$
- $f = \emptyset$
- $R((P, 2)) = \geq$

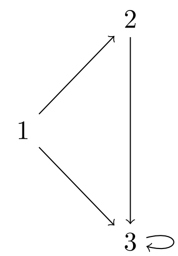
We have that, for example:

$$\exists x[P(x,x)] : \text{‘some natural number is greater than or equal to itself’}$$

is true in $\mathcal{M}_3$.

Example 21. A graph can also be seen as a $\mathcal{S}$-structure. Consider the signature $\mathcal{S} = \langle \mathcal{F}, \mathcal{R} \rangle$, where:

- $\mathcal{F} = \emptyset$
- $\mathcal{R} = \{(P, 2)\}$



and the structure $\mathcal{M}_4 = \langle \mathcal{D}, f, R \rangle$, where:

- $\mathcal{D} = \{1, 2, 3\}$
- $f = \emptyset$
- $R((P, 2)) = \{(1, 2), (1, 3), (2, 3), (3, 3)\}$

We have that, for example:

$$\forall x \exists y[P(x,y)] : \text{‘every node } x \text{ has a node } y \text{ such that there is an arrow from } x \text{ to } y\text{’}$$

is true in $\mathcal{M}_4$.

Assignment

In order to give meaning to a formulas containing free variables, e.g., $P(a)$, one has to give an interpretation to these variables.

Definition 22 ($\mathcal{M}$-assignment)**.** Let a set of variables X and a $\mathcal{S}$-structure $\mathcal{M} = \langle \mathcal{D}, f, R \rangle$ be given. A $\mathcal{M}$**-assignment** of X is an application:

$$\lambda : X \to \mathcal{D}$$

We use Λ to denote the set of all possible assignments on the domain $\mathcal{D}$.

Satisfaction of atomic formulas

Definition 23 (Interpretation)**.**

$$[\![o]\!]_{\mathcal{M},\lambda} = \begin{cases} f(o), & \text{if } o \text{ is an individual constant} \\ \lambda(o), & \text{if } o \text{ is an individual variable} \\ R((o,n)), & \text{if } o \text{ is a predicate and } n \text{ is its arity} \end{cases}$$

Definition 24 (Satisfaction of atomic formulas)**.** Let an alphabet with signature $\mathcal{S}$ be given. An $\mathcal{S}$-structure $\mathcal{M}$ and an assignment λ over its domain **satisfy an atomic formula** $P(r_1, \ldots, r_n)$ if and only if:

$$([\![r_1]\!]_{\mathcal{M},\lambda}, \ldots [\![r_n]\!]_{\mathcal{M},\lambda}) \in [\![P]\!]_{\mathcal{M},\lambda}$$

Example 25. Consider the $\mathcal{S}$-structure $\mathcal{M} = \langle \mathcal{D}, f, R \rangle$, where:

- $\mathcal{D} = \mathbb{N}$
- $f(c) = 2$
- $R((P, 2)) = \geq$

and the $\mathcal{M}_1$-assignment λ, where:

- $\lambda(a) = 3$
- $\lambda(b) = 0$

$\mathcal{M}, \lambda$ satisfies $P(a, c)$	because	$([\![a]\!]_{\mathcal{M},\lambda}, [\![c]\!]_{\mathcal{M},\lambda}) \in [\![P]\!]_{\mathcal{M},\lambda}$
	because	$(\lambda(a), f(c)) \in R((P, 2))$
	because	$(3, 2) \in \geq$
	because	$3 \geq 2$

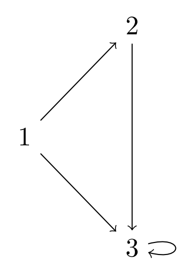
$\mathcal{M}, \lambda$ does not satisfy $P(b, c)$	because	$([\![b]\!]_{\mathcal{M},\lambda}, [\![c]\!]_{\mathcal{M},\lambda}) \notin [\![P]\!]_{\mathcal{M},\lambda}$
	because	$(\lambda(b), f(c)) \notin R((P, 2))$
	because	$(0, 2) \notin \geq$
	because	$0 \ngeq 2$

Example 26. Consider the $\mathcal{S}$-structure $\mathcal{M} = \langle \mathcal{D}, f, R \rangle$:

- $\mathcal{D} = \{1, 2, 3\}$
- $f = \emptyset$
- $R(P, 2) = \{(1, 2), (1, 3), (2, 3), (3, 3)\}$

and the $\mathcal{M}$-assignment λ, where:

- $\lambda(a) = 1$
- $\lambda(b) = 2$



$\mathcal{M}, \lambda$ satisfies $P(a, b)$	because	$([\![a]\!]_{\mathcal{M},\lambda}, [\![b]\!]_{\mathcal{M},\lambda}) \in [\![P]\!]_{\mathcal{M},\lambda}$
	because	$(\lambda(a), \lambda(b)) \in R((P, 2))$
	because	$(1, 2) \in \{(1, 2), (1, 3), (2, 3), (3, 3)\}$

$\mathcal{M}, \lambda$ does not satisfy $P(b, a)$	because	$([\![b]\!]_{\mathcal{M},\lambda}, [\![a]\!]_{\mathcal{M},\lambda}) \notin [\![P]\!]_{\mathcal{M},\lambda}$
	because	$(\lambda(b), \lambda(a)) \notin R((P, 2))$
	because	$(2, 1) \notin \{(1, 2), (1, 3), (2, 3), (3, 3)\}$

Satisfaction relation

Definition 27 (Satisfaction relation)**.** The **satisfaction relation**, $\models$, between formulas φ and pairs formed by a $\mathcal{S}$-structure and a $\mathcal{M}$-assignment λ is recursively defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, \lambda \models P(r_1, \ldots, r_n) & \text{iff} & ([\![r_1]\!]_{\mathcal{M},\lambda}, \ldots, [\![r_n]\!]_{\mathcal{M},\lambda}) \in [\![P]\!]_{\mathcal{M},\lambda} \\
\mathcal{M}, \lambda \models \neg\varphi & \text{iff} & \text{not } \mathcal{M}, \lambda \models \varphi \\
\mathcal{M}, \lambda \models \varphi_1 \wedge \varphi_2 & \text{iff} & \mathcal{M}, \lambda \models \varphi_1 \text{ and } \mathcal{M}, \lambda \models \varphi_2 \\
\mathcal{M}, \lambda \models \varphi_1 \vee \varphi_2 & \text{iff} & \mathcal{M}, \lambda \models \varphi_1 \text{ or } \mathcal{M}, \lambda \models \varphi_2 \\
\mathcal{M}, \lambda \models \varphi_1 \rightarrow \varphi_2 & \text{iff} & \text{not } \mathcal{M}, \lambda \models \varphi_1 \text{ or } \mathcal{M}, \lambda \models \varphi_2 \\
\mathcal{M}, \lambda \models \varphi_1 \leftrightarrow \varphi_2 & \text{iff} & \mathcal{M}, \lambda \models \varphi_1 \text{ if and only if } \mathcal{M}, \lambda \models \varphi_2 \\
\mathcal{M}, \lambda \models \forall x \varphi & \text{iff} & \text{for all } d \in \mathcal{D} \text{ we have } \mathcal{M}, \lambda[x := d] \models \varphi \\
\mathcal{M}, \lambda \models \exists x \varphi & \text{iff} & \text{there is } d \in \mathcal{D} \text{ such that } \mathcal{M}, \lambda[x := d] \models \varphi
\end{array}
$$

where $\lambda[x := d] = \lambda \setminus \{(x, \lambda(x))\} \cup \{(x, d)\}$.

(We also use $\mathcal{M}, \lambda \not\models \varphi$ to denote 'not $\mathcal{M}, \lambda \models \varphi$'.)

Example 28. Consider the $\mathcal{S}$-structure $\mathcal{M} = \langle \mathcal{D}, f, R \rangle$ and the $\mathcal{M}$-assignment λ, where:

- $\mathcal{D} = \mathbb{N}$
- $f(c) = 2$
- $R((P, 2)) = \geq$
- $\lambda(a) = 2$
- $\lambda(b) = 0$

We have:

$$\mathcal{M}, \lambda \models \forall x[P(x, b)]$$

because for all $d \in \mathcal{D}$ we have $\mathcal{M}, \lambda[x := d] \models P(x, b)$
because for all $d \in \mathbb{N}$ we have $([\![x]\!]_{\mathcal{M},\lambda[x:=d]}, [\![b]\!]_{\mathcal{M},\lambda[x:=d]}) \in [\![P]\!]_{\mathcal{M},\lambda[x:=d]}$
because for all $d \in \mathbb{N}$ we have $(\lambda[x := d](x), \lambda[x := d](b)) \in R((P, 2))$
because for all $d \in \mathbb{N}$ we have $(d, 0) \in \geq$ because for all $d \in \mathbb{N}$ we have $d \geq 0$

$$\mathcal{M}, \lambda \not\models \forall x[P(x, a)]$$

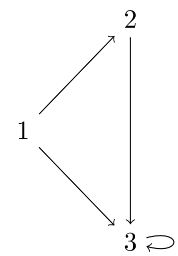
because not for all $d \in \mathcal{D}$ we have $\mathcal{M}, \lambda[x := d] \models P(x, a)$
because not for all $d \in \mathcal{D}$ we have $([\![x]\!]_{\mathcal{M},\lambda[x:=d]}, [\![a]\!]_{\mathcal{M},\lambda[x:=d]}) \in [\![P]\!]_{\mathcal{M},\lambda[x:=d]}$
because not for all $d \in \mathcal{D}$ we have $(\lambda[x := d](x), \lambda[x := d](a)) \in R((P, 2))$
because not for all $d \in \mathcal{D}$ we have $(d, 2) \in \geq$
because not for all $d \in \mathcal{D}$ we have $d \geq 2$
(e.g., $d = 0$)

Example 29. Consider the $\mathcal{S}$-structure $\mathcal{M} = \langle \mathcal{D}, f, R \rangle$:

- $\mathcal{D} = \{1, 2, 3\}$
- $f = \emptyset$
- $R((P, 2)) = \{(1, 2), (1, 3), (2, 3), (3, 3)\}$

and the $\mathcal{M}$-assignment λ, where:

- $\lambda(a) = 1$
- $\lambda(b) = 2$



$$\mathcal{M}, \lambda \models \exists x[P(x, x)]$$

because there is $d \in \mathcal{D}$ such that $\mathcal{M}, \lambda[x := d] \models P(x, x)$ because there is $d \in \mathcal{D}$ such that $([\![x]\!]_{\mathcal{M},\lambda[x:=d]}, [\![x]\!]_{\mathcal{M},\lambda[x:=d]}) \in [\![P]\!]_{\mathcal{M},\lambda[x:=d]}$ because there is $d \in \mathcal{D}$ such that $(\lambda[x := d](x), \lambda[x := d](x)) \in R((P, 2))$ because there is $d \in \mathcal{D}$ such that $(d, d) \in \{(1, 2), (1, 3), (2, 3), (3, 3)\}$ (e.g., $d = 3$)

$$\mathcal{M}, \lambda \not\models \exists x[P(x, a)]$$

because there is no $d \in \mathcal{D}$ such that $([\![x]\!]_{\mathcal{M},\lambda[x:=d]}, [\![a]\!]_{\mathcal{M},\lambda[x:=d]}) \in [\![P]\!]_{\mathcal{M},\lambda[x:=d]}$ because there is no $d \in \mathcal{D}$ such that $(\lambda[x := d](x), \lambda[x := d](a)) \in R((P, 2))$ because there is no $d \in \mathcal{D}$ such that $(d, 1) \in \{(1, 2), (1, 3), (2, 3), (3, 3)\}$

Satisfiability

Definition 30 (Satisfiability)**.** A formula φ is **satisfiable** if and only if there is a $\mathcal{S}$-structure $\mathcal{M}$ and a $\mathcal{M}$-assignment λ such that $\mathcal{M}, \lambda \models \varphi$.

Examples 31.

- $\forall x[P(x)]$ is satisfiable (but not valid).
- $\forall x[P(x)] \wedge \neg\exists x[P(x)]$ is not satisfiable.

Truth

Definition 32 (Truth)**.** A formula φ is **true** in a $\mathcal{S}$-structure $\mathcal{M}$ if and only if, for all $\mathcal{M}$-assignments λ, we have $\mathcal{M}, \lambda \models \varphi$.

We use $\mathcal{M} \models \varphi$ (without the assignment) to denote that φ is true in $\mathcal{M}$.

When a formula φ is true in a structure $\mathcal{M}$ we also say that $\mathcal{M}$ is a **model** of φ.

Examples 33.

- The structure $\mathcal{M}_1 = \langle \mathbb{N}, \{(c, 0)\}, \{((P, 2), \geq)\}\rangle$ is a model of $P(a, c)$, since, for all assignments λ we have $\mathcal{M}_1, \lambda \models P(a, c)$.
- The structure $\mathcal{M}_2 = \langle \mathbb{N}, \{(c, 2)\}, \{((P, 2), \geq)\}\rangle$ is not a model of $P(a, c)$, since, there is an assignment λ such that $\mathcal{M}_2, \lambda \not\models P(a, c)$ (e.g., $\lambda(a) = 0$).
- The structure $\mathcal{M}_3 = \langle \mathbb{N}, \emptyset, \{(P, 2), <)\}\rangle$ is not a model of $\exists x[P(x, a)]$, since there is an assignment λ such that $\mathcal{M}_3, \lambda \not\models \exists x[P(x, a)]$ (e.g., $\lambda(a) = 0$).

Validity

Definition 34 (Validity)**.** A formula φ is **valid** if and only if, for all $\mathcal{S}$-structure $\mathcal{M}$, we have $\mathcal{M} \models \varphi$.

We use $\models \varphi$ (without the structure) to denote that φ is valid.

Examples 35.

- $\models \forall x[P(x,c) \vee \neg P(x,c)]$
- $\models \forall x[P(x,a) \to P(x,a)]$
- $\models (\forall x[P(x) \to Q(x)] \wedge P(c)) \to Q(c)$
- $\models \neg\forall x[P(x)] \leftrightarrow \exists x[\neg P(x)]$

2.3 Theorems

Theorem 36. $\models \neg\forall x[P(x)] \leftrightarrow \exists x[\neg P(x)]$

Proof. Let $\mathcal{M} = \langle \mathcal{D}, f, R \rangle$ be any $\mathcal{S}$-structure and λ be any $\mathcal{M}$-assignment. $\mathcal{M}, \lambda \models \neg\forall x[P(x)]$ if and only if not $\mathcal{M}, \lambda \models \forall x[P(x)]$, if and only if not for all $d \in \mathcal{D}$ we have $\mathcal{M}, \lambda[x := d] \models P(x)$, if and only if there is $d \in \mathcal{D}$ such that not $\mathcal{M}, \lambda[x := d] \models P(x)$, if and only if there is $d \in \mathcal{D}$ such that $\mathcal{M}, \lambda[x := d] \models \neg P(x)$, if and only if $\mathcal{M}, \lambda \models \exists x[\neg P(x)]$. This means that, $\mathcal{M}, \lambda \models \neg\forall x[P(x)]$ iff $\mathcal{M}, \lambda \models \exists x[\neg P(x)]$. The later is the case if and only if $\mathcal{M}, \lambda \models \neg\forall x[P(x)] \leftrightarrow \exists x[\neg P(x)]$. Therefore, $\models \neg\forall x[P(x)] \leftrightarrow \neg\exists x[\neg P(x)]$ (because we assumed arbitrary $\mathcal{M}$ and λ). $\square$

Theorem 37. $\models \neg\exists x[P(x)] \leftrightarrow \forall x[\neg P(x)]$

Exercise 2.1. *Prove the theorem above.*

Theorem 38.

1. $\models \neg\forall x[\neg P(x)] \leftrightarrow \exists x[P(x)]$

2. $\models \neg\exists x[\neg P(x)] \leftrightarrow \forall x[P(x)]$

Exercise 2.2. *Prove the theorem above.*

Theorem 39. *1.* $\models (\forall x[P(x)] \vee \forall x[Q(x)]) \to \forall x[P(x) \vee Q(x)]$

2. not $\models \forall x[P(x) \vee Q(x)] \to (\forall x[P(x)] \vee \forall x[Q(x)])$

Exercise 2.3. *Prove the theorem above.*

2.4 Exercises

Exercise 2.4. Check that the following formulas are valid

1. $\forall x[P(x) \to Q(x)] \to (\forall x[P(x)] \to \forall x[Q(x)])$
2. $(\exists x[P(x) \to \exists x[Q(x)]) \to \exists x[P(x) \to Q(x)]$

Exercise 2.5. *Give counterexamples to the following formulas:*

1. $(\forall x[P(x)] \to \forall x[Q(x)]) \to \forall x[P(x) \to Q(x)]$
2. $\exists x[P(x) \to Q(x)] \to (\exists x[P(x)] \to \exists x[Q(x)]$

Exercise 2.6. *Give counterexamples to the following formulas:*

1. $\exists x[\neg P(x)] \to \neg\exists x[P(x)]$
2. $\neg\forall x[P(x)] \to \forall x[\neg P(x)]$

Exercise 2.7. *Check that the following formulas are valid.*

1. $\neg\exists x[P(x)] \to \exists x[\neg P(x)]$
2. $\forall x[\neg P(x)] \to \neg\forall x[P(x)]$

Exercise 2.8. *Check that the following formulas are valid.*

1. $\forall x\exists y[P(x) \to P(y)]$
2. $\forall y\exists x[P(x) \to P(y)]$
3. $\exists x\forall y[P(x) \to P(y)]$
4. $\exists y\forall x[P(x) \to P(y)]$

Exercise 2.9. *Which of the following formulas are valid? Give either an intuitive proof or a counterexample.*

1. $\forall x\exists y[P(x,y)] \to \exists x\forall y[P(x,y)]$
2. $\exists x\forall y[P(x,y)] \to \forall x\exists y[P(x,y)]$
3. $\forall x\exists y[P(x,y)] \to \exists x\forall y[P(y,x)]$
4. $\exists x\forall y[P(x,y)] \to \forall y\exists x[P(x,y)]$

Exercise 2.10. *Which of the following formulas are valid? Give either an intuitive proof or a counterexample.*

1. $(\forall x[P(x)] \to \exists x[Q(x)]) \leftrightarrow \exists x[P(x) \to Q(x)]$
2. $(\exists x[P(x)] \to \forall x[Q(x)]) \leftrightarrow \forall x[P(x) \to Q(x)]$

Exercise 2.11. *Show that* $\models \neg\exists\forall x[S(y,x) \leftrightarrow \neg S(x,x)]$
(Think of 'y shaves x' and recall Russell's barber's paradox.)

Exercise 2.12. *Which of the following formulas are valid? Give either an intuitive proof or a counterexample.*

1. $\forall x[P(x) \to Q(x)] \to (\exists x[P(x)] \to \exists x[Q(x)]$
2. $\forall x[P(x) \to Q(x)] \leftarrow (\exists x[P(x)] \to \exists x[Q(x)]$
3. $\exists x[P(x) \to Q(x)] \to (\forall x[P(x)] \to \forall x[Q(x)]$
4. $\exists x[P(x) \to Q(x)] \leftarrow (\forall x[P(x)] \to \forall x[Q(x)]$

Exercise 2.13. *Prove or refute:*

$$\models \forall x \exists y[P(x) \to Q(y)] \to \exists y \forall x[P(x) \to Q(y)]$$

Exercise 2.14. *Which of the following alternatives (i), (ii), (iii) applies to the following formuas?*

(i) *not satisfiable*

(ii) *satisfiable, but not valid*

(iii) *valid, and hence satisfiable*

1. $\exists x[P(x)] \to \forall x[P(x)]$
2. $\exists x[P(x)] \to \exists x[\neg P(x)]$
3. $\exists x[P(x)] \land \forall x[\neg P(x)]$
4. $\forall x[P(x)] \land \neg\exists x[P(x)]$
5. $\neg\forall x[P(x)] \to \forall x[\neg P(x)]$
6. $\forall x[\neg P(x)] \to \neg\forall x[P(x)]$
7. $\forall x \exists y[P(x, y)] \land \exists x \forall y[\neg P(x, y)]$
8. $\forall x \exists y[P(x, y)] \to \exists y \forall x[P(x, y)]$
9. $(\exists x[P(x)] \land \exists x[Q(x)]) \to \exists x[P(x) \land Q(x)]$
10. $\forall x[P(x) \lor Q(x)] \to (\exists x[P(x)] \lor \forall x[Q(x)])$

Chapter 3

Model theory II

3.1 Valid consequence

Valid consequence

Definition 40 (Valid consequence)**.** ψ is a **valid consequence** of $\varphi_1, \ldots, \varphi_m$ if and only if, for every $\mathcal{S}$-structure $\mathcal{M}$ and all $\mathcal{M}$-assignment λ, if for all i, $1 \leq i \leq m$, $\mathcal{M}, \lambda \models \varphi_i$, then $\mathcal{M}, \lambda \models \psi$.

Notation: $\psi_1, \ldots, \psi_n \models \varphi$
We also say:

- ψ is a **semantic consequence** of $\varphi_1, \ldots, \varphi_m$.
- ψ is a **logical consequence** of $\varphi_1, \ldots, \varphi_m$.
- $\varphi_1, \ldots, \varphi_m$ **entails** ψ.

Example 41.

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

In particular, for $\mathcal{M} = \langle \mathcal{D}, f, R \rangle$, where:

- $\mathcal{D}$: set of all people
- $f(c) =$ Caspar
- $R((P, 1)) = \{a \mid a \text{ is a man}\}$ (set of men)
- $R((Q, 1)) = \{a \mid a \text{ is mortal}\}$ (set of mortal people)

it means:

All men are mortal.
Caspar is a men.
Therefore, Caspar is mortal.

To see that it holds for all $\mathcal{S}$-structures.
Assume arbitrary $\mathcal{S}$-structure $\mathcal{M}$ and $\mathcal{M}$-assignment λ:
Suppose $\mathcal{M}, \lambda \models \forall x[P(x) \to Q(x)]$ and $\mathcal{M}, \lambda \models P(c)$:
We have that, for all $d \in \mathcal{D}$, $\mathcal{M}, \lambda[x := d] \models P(x) \to Q(x)$.
In particular, we have that $\mathcal{M}, \lambda[x := f(c)] \models P(x) \to Q(x)$,
iff $\mathcal{M}, \lambda[x := f(c)] \not\models P(x)$ or $\mathcal{M}, \lambda[x := f(c)] \models Q(x)$.
Suppose $\mathcal{M}, \lambda[x := f(c)] \not\models P(x)$:
We have that $f(c) \notin R((P, 1))$,
iff $\mathcal{M}, \lambda \not\models P(c)$.
The latter contradicts the second assumption.
Therefore, $\mathcal{M}, \lambda[x := f(c)] \models Q(x)$,
iff $f(c) \in R((Q, 1))$,
iff $\mathcal{M}, \lambda \models Q(c)$.
Then, if $\mathcal{M}, \lambda \models \forall x[P(x) \to Q(x)]$ and $\mathcal{M}, \lambda \models P(c)$ then $\mathcal{M}, \lambda \models Q(c)$.
Therefore, for all $(\mathcal{M}, \lambda)$, if $\mathcal{M}, \lambda \models \forall x[P(x) \to Q(x)]$ and $\mathcal{M}, \lambda \models P(c)$, then $\mathcal{M}, \lambda \models Q(c)$. (because $(\mathcal{M}, \lambda)$ has been taken arbitrarily), iff $\forall x[P(x) \to Q(x)], P(c) \models Q(c)$. □

Examples 42.

1. $\forall x[P(x) \to Q(x)], \exists x[R(x) \land \neg Q(x)] \models \exists x[R(x) \land \neg P(x)]$

   This is one of Aristotle's syllogisms:

   > All logicians are philosophers. There are men who are not philosophers. Hence there are men who are not philosophers.

2. $\forall x[P(x) \to \neg Q(x)], \exists x[R(x) \land Q(x)] \models \exists x[R(x) \land \neg P(x)]$

   Another Aristotle's syllogism.

3. $\forall x[P(x) \to Q(x)] \models \exists x[P(x) \to Q(x)]$

   It holds because $\mathcal{D}$ is not empty.

4. $\forall x[P(x) \to Q(x)] \not\models \exists x[P(x) \land Q(x)]$

   Consider the structure $\mathcal{M} = \langle \mathcal{D}, \emptyset, R \rangle$, where:

   - $\mathcal{D} = \mathbb{N}$
   - $R((P, 1)) = \{a \mid a \neq a\}$
   - $R((Q, 1)) = \{a \mid a \mod 2 = 0\}$ (a is even)

Semantic deduction theorem

Theorem 43 (Semantic deduction)**.**

1. $\varphi \models \psi$ *if and only if* $\models \varphi \to \psi$

2. $\varphi_1, \ldots, \varphi_{m-1}, \varphi_m \models \psi$ *if and only if* $\varphi_1, \ldots, \varphi_{m-1} \models \varphi_m \to \psi$

Proof. Exercise. □

Valid consequence II

Definition 44 (Valid consequence II)**.** ψ is a **valid consequence** of $\varphi_1, \ldots, \varphi_m$ **with all free variables general** if and only if for every $\mathcal{S}$-structure $\mathcal{M}$, if, for all i, $1 \leq i \leq m$, $\mathcal{M} \models \varphi_i$, then $\mathcal{M} \models \psi$.

Notation: $\psi_1, \ldots \psi_m \models^2 \varphi$

Theorem 45. *If $\psi \models \varphi$ then $\psi \models^2 \varphi$ but, in general, not conversely.*

Proof. First part: Exercise.

Second part: Let $\psi = P(a)$ and $\varphi = \forall x[P(x)]$.
First, we show that $P(a) \models^2 \forall x[P(x)]$.
Assume an arbitrary $\mathcal{S}$-structure $\mathcal{M}$:
Suppose $\mathcal{M} \models P(a)$:
We have that, for all $\mathcal{M}$-assignment λ, $\mathcal{M}, \lambda \models P(a)$.
Then, for all $\mathcal{M}$-assignment λ, $\lambda(a) \in R((P, 1))$.
Assume an arbitrary λ and an arbitrary $d \in \mathcal{D}$ s.t. $\lambda(a) = d$:
We have that $d \in R((P, 1))$,
iff $\mathcal{M}, \lambda[x := d] \models P(x)$.
Therefore for all λ and all $d \in \mathcal{D}$, $\mathcal{M}, \lambda[x := d] \models P(x)$.
iff, for all λ, $\mathcal{M}, \lambda \models \forall x[P(x)]$.
iff, $\mathcal{M} \models \forall x[P(x)]$.
Therefore, if $\mathcal{M} \models P(a)$ then $\mathcal{M} \models \forall x[P(x)]$.
Therefore, for all $\mathcal{M}$, if $\mathcal{M} \models P(a)$ then $\mathcal{M} \models \forall x[P(x)]$ (because $\mathcal{M}$ is taken arbitrarily),
iff $P(a) \models^2 \forall x[P(x)]$.

Now, we show that $P(a) \not\models \forall x[P(x)]$. Let $\mathcal{M} = \langle \mathcal{D}, f, R \rangle$ and λ, were:

- $\mathcal{D} = \{d_1, d_2\}$
- $f = \emptyset$
- $R((P, 1)) = \{d_1\}$
- $\lambda(a) = d_1$
- $\lambda(b) = d_2$.

We have that $\mathcal{M}, \lambda \models P(a)$, but $\mathcal{M}, \lambda \not\models P(b)$. Then, $\mathcal{M}, \lambda \not\models \forall x[P(x)]$. Therefore, $P(a) \not\models \forall x[P(x)]$. □

An example of $\models$ vs. $\models^2$ in "real life"

Consider the equation:

$$a^2 = 3a$$

We can infer from it that $a = 3$, i.e.:

$$(a = 3) \text{ is a valid consequence of } (a^2 = 3a)$$

However, we cannot infer from it that:

$$b^2 = 3b$$

because, when we assign, e.g., 2 to the free variable b, the latter equation is false, i.e.:

$$(b^2 = 3b) \text{ is } \textbf{not} \text{ a valid consequence of } (a^2 = 3a)$$

Here, we are using the first form of consequence (i.e., $\models$).

Now, consider the equation:

$$a + b = b + a$$

We can infer from it that $c + d = d + c$, i.e.:

$$c + d = d + c \text{ is a valid consequence of } a + b = b + a$$

because, the equation is true whatever assignments we make for the variables involved.

Here, we are using the second form of consequence (i.e., $\models^2$).

We can also express the difference between $\models$ and $\models^2$ in natural language, as follows.

The first argument can be written:

$$\text{for all } a, \text{ if } a^2 = 3a \text{ then } a = 3$$

whereas the second can be written:

$$\text{for all } a \text{ and all } b, \text{ if } a + b = b + a \text{ then for all } c \text{ and all } d, c + d = d + c$$

3.2 Results about validity and consequence

Theorem 46. *Let r be a free individual variable or an individual constant. Let $\varphi(r)$ result from $\varphi(x)$ by substituting r for all occurrences of x in $\varphi(x)$. Then:*

1. $\models \forall x[\varphi(x)] \to \varphi(r)$, *where r is free for x in $\varphi(x)$*[1]

2. $\models \varphi(r) \to \exists x[\varphi(x)]$

Example 47. Let $\mathcal{M} = \langle \mathcal{D}, f, R \rangle$, where:

- $\mathcal{D}$ is the set of all persons
- $P(a)$ means a is mortal
- $f(c)$ is 'John Lennon'

we have, respectively:

1. If all persons are mortal, then John Lennon is mortal.
2. If John Lennon is mortal, then some person is mortal.

[1] If any free occurrence of x in φ is replaced by an occurrence of r, then the resulting occurrence of r in $\varphi(r)$ is also free.

Proof. We show item 1.

Assume arbitrary $\mathcal{M}$ and λ:

Suppose $\mathcal{M}, \lambda \models \forall x[\varphi(x)]$:

iff for all $d \in \mathcal{D}$, $\mathcal{M}, \lambda[x := d] \models \varphi(x)$,

iff for all $d \in \mathcal{D}$, $\mathcal{M}, \lambda[a := d] \models \varphi(a)$, where a is a free variable not occurring in $\varphi(x)$.

Then, in particular, $\mathcal{M}, \lambda[a := \lambda(r)] \models \varphi(a)$,

iff $\mathcal{M}, \lambda \models \varphi(r)$.

Therefore, if $\mathcal{M}, \lambda \models \forall x[\varphi(x)]$ then $\mathcal{M}, \lambda \models \varphi(r)$,

iff $\mathcal{M}, \lambda \models \forall x[\varphi(x)] \rightarrow \varphi(r)$.

Therefore, for all $\mathcal{M}, \lambda$, $\mathcal{M}, \lambda \models \forall x[\varphi(x)] \rightarrow \varphi(r)$,

iff $\models \forall x[\varphi(x)] \rightarrow \varphi(r)$.

The proof of item 2 is similar. □

Note that if y is a bound variable, then:

$$\not\models \forall x[\varphi(x)] \rightarrow \varphi(y)$$

For instance, let $\varphi(x) = \exists y[P(x, y)]$ and $\mathcal{M} = \langle \mathcal{D}, f, R \rangle$, where:

- $\mathcal{D} = \mathbb{N}$
- $R((P, 2)) = \{x, y \mid x < y\}$

We have:

$$\not\models \forall x \exists y[P(x, y)] \rightarrow \exists y[P(y, y)]$$

In words: 'For all x, there is y such that $x < y$ implies that there is y such that $y < y$' is false.

Theorem 48. *Let $\varphi(a)$ be a formula and let ψ be a formula not containing the free variable a.*

1. *If $\mathcal{M} \models \varphi(a)$, then $\mathcal{M} \models \forall x[\varphi(x)]$, but $\varphi(a) \not\models \forall x[\varphi(x)]$.*
2. *If $\mathcal{M} \models \psi \rightarrow \varphi(a)$, then $\mathcal{M} \models \psi \rightarrow \forall x[\varphi(x)]$, but $\psi \rightarrow \varphi(a) \not\models \psi \rightarrow \forall x[\varphi(x)]$.*
3. *If $\mathcal{M} \models \varphi(a) \rightarrow \psi$, then $\mathcal{M} \models \exists x[\varphi(x)] \rightarrow \psi$, but $\varphi(a) \rightarrow \psi \not\models \exists x[\varphi(x)] \rightarrow \psi$.*

Proof.

1. Suppose $\mathcal{M} \models \varphi(a)$:

   iff for all λ, $\mathcal{M}, \lambda \models \varphi(a)$,

   Assume arbitrary λ :

   For all $d \in \mathcal{D}$, $\mathcal{M}, \lambda[a := d] \models \varphi(a)$,

   iff for all $d \in \mathcal{D}$, $\mathcal{M}, \lambda[x := d] \models \varphi(x)$,

   iff $\mathcal{M}, \lambda \models \forall x[\varphi(x)]$,

   Therefore, for all λ, $\mathcal{M}, \lambda \models \forall x[\varphi(x)]$,

   iff $\mathcal{M} \models \forall x[\varphi(x)]$.

   Therefore, if $\mathcal{M} \models \varphi(a)$ then $\mathcal{M} \models \forall x[\varphi(x)]$.

On the other hand, let $\mathcal{M} = \langle \mathbb{N}, \emptyset, R \rangle$, where $R((P,1)) = \{a \mid a \text{ is even}\}$.

We have $\mathcal{M}, \lambda[a := 2] \models P(a)$,

but $\mathcal{M}, \lambda[x := 3] \not\models P(x)$.

Therefore, $\mathcal{M}, \lambda \not\models \forall x[P(x)]$.

2. Suppose $\mathcal{M} \models \psi \to \varphi(a)$, where ψ does not contain a:

   Then $\mathcal{M} \models \forall x[\psi \to \varphi(x)]$ (by item 1 of the Theo.).

   Then $\mathcal{M} \models \psi \to \forall x[\varphi(x)]$.

   On the other hand, let $\psi = \chi \vee \neg\chi$.

   Then $\psi \to \varphi(a)$ is equivalent to $\varphi(a)$

   and $\psi \to \forall x[\varphi(x)]$ is equivalent to $\forall x[\varphi(x)]$.

   Therefore, $\psi \to \varphi(a) \models \psi \to \forall x[\varphi(x)]$ is equivalent to $\varphi(a) \models \forall x[\varphi(x)]$,

   The latter is false, according to item 1 of the Theorem.

3. Suppose $\mathcal{M} \models \varphi(a) \to \psi$ and ψ does not contain a:

   We have, for all λ, $\mathcal{M}, \lambda \models \varphi(a) \to \psi$,

   Assume an arbitrary λ:

   We have, $\mathcal{M}, \lambda \models \varphi(a) \to \psi$.

   Then, for every $d \in \mathcal{D}$, $\mathcal{M}, \lambda[a := d] \models \varphi(a) \to \psi$ (by the initial assumption),

   Suppose $\mathcal{M}, \lambda \models \exists x[\varphi(x)]$:

   We have, there is $d' \in \mathcal{D}$ s.t. $\mathcal{M}, \lambda[x := d'] \models \varphi(x)$.

   Then, there is $d' \in \mathcal{D}$ s.t. $\mathcal{M}, \lambda[a := d'] \models \varphi(a)$.

   Then, there is $d' \in \mathcal{D}$ s.t. $\mathcal{M}, \lambda[a := d'] \models \psi$ (by assumption).

   Then $\mathcal{M}, \lambda \models \psi$ (because ψ does not contain a).

   Therefore, $\mathcal{M}, \lambda \models \exists x[\varphi(x)] \to \psi$.

   Therefore $\mathcal{M} \models \exists x[\varphi(x)] \to \psi$.

   On the other hand, let $\mathcal{M} = \langle \{d_1, d_2\}, \emptyset, R \rangle$, where $R((P,1)) = \{d_1\}$,

   and $\lambda(a) = d_2$.

   Let $\psi = \chi \wedge \neg\chi$.

   We have $\mathcal{M}, \lambda \models P(a) \to \psi$,

   but $\mathcal{M}, \lambda \not\models \exists x[P(x)] \to \psi$. □

In the preceding theorem, the condition that ψ does not contain the free variable a is necessary.

For instance, we have $\models \varphi(a) \to \varphi(a)$, but $\not\models \varphi(a) \to \forall x[\varphi(x)]$.

- E.g.: 'If Antoine has his birthday in in March, then everyone has ones birthday in March' is false.

Also, we have $\models \varphi(a) \to \varphi(a)$, but $\not\models \exists x[\varphi(x)] \to \varphi(x)$.

- E.g.: 'If there is an even number, then 3 is even' is false.

Theorem 49 (Substitution for atomic formulas)**.** *Let ψ' be built from the formulas $\varphi_1, \ldots, \varphi_k$ in the same way as ψ has been built from the atomic formulas $P_1(r_1, \ldots, r_{n_1}), \ldots, P_k(r_1, \ldots, r_{n_k})$ respectively. If $\models \psi'$ then $\models \psi$.*

Corollary 50. *All theorems in Lecture 2 hold for any formulas φ and ψ (not only P and Q).*

Corollary 51. *All axioms of propositional logic hold for any formulas φ, ψ and χ of the predicate calculus.*

Chapter 4

Proof theory

4.1 Provability and deducibility

Provability and deducibility

One can select a small number of axiom schemas and rules of inference such that:

1. All valid formulas can be obtained by finitely many applications of the rules to instances of the axioms.
2. For any premises $\varphi_1, \ldots, \varphi_m$, all their valid consequences can be obtained by finitely many applications of the rules to $\varphi_1, \ldots, \varphi_m$ and the axioms.

There are systematic procedures of searching for a deduction of ψ given premisses $\varphi_1, \ldots, \varphi_m$.

However, A. Church (1936) and A. Turing (1936-37) proved independently that a decision procedure for the predicate calculus does not exist.

The procedures we will see yield a positive test for validity, but it does not give a negative test for validity.

We have selected the following axiom schemata for propositional logic:

1. $A \to (B \to A)$
2. $(A \to B) \to ((A \to (B \to C)) \to (A \to C))$
3. $A \to (B \to A \land B)$

4a. $A \land B \to A$

4b. $A \land B \to B$

5a. $A \to A \lor B$

5b. $B \to A \lor B$

6. $(A \to C) \to ((B \to C) \to (A \lor B \to C))$
7. $(A \to B) \to ((A \to \neg B) \to \neg A)$

8. $\neg\neg A \to A$
9. $(A \to B) \to ((B \to A) \to (A \leftrightarrow B))$

10a. $(A \leftrightarrow B) \to (A \to A)$

10b. $(A \leftrightarrow B) \to (B \to A)$

and the rule of inference:

MP: From $A \to B$ and A infer B.

We add two axiom schemata:

11. $\forall x[\varphi(x)] \to \varphi(r)$
12. $\varphi(r) \to \exists x[\varphi(x)]$

where r is a free individual variable or an individual constant.

We also add two rules of inference:

$\forall$: From $\psi \to \varphi(a)$ infer $\psi \to \forall x[\varphi(x)]$

$\exists$: From $\varphi(a) \to \psi$ infer $\exists x[\varphi(x)] \to \psi$

Proof

Definition 52 (Proof)**.** A (Hilbert-type) **proof** of a formula φ is a finite list $\varphi_1, \ldots, \varphi_k$ of formulas such that:

1. $\varphi = \varphi_k$ is the last formula in the list, and
2. each formula in the list is either:
   (a) an axiom, or
   (b) is obtained by an application of an inference rule to formulas preceding it in the list.

Definition 53. A formula φ is **provable** if and only if there exists a (Hilbert-type) proof of φ.

Notation: $\vdash \varphi$.

Example 54. Let us show that $P \to P$ is provable.

1.	$P \to (P \to P)$	(Ax. 1)
2.	$(P \to (P \to P)) \to ((P \to ((P \to P) \to P)) \to (P \to P))$	(Ax. 2)
3.	$(P \to ((P \to P) \to P)) \to (P \to P)$	(MP: 1, 2)
4.	$P \to ((P \to P) \to P)$	(Ax. 1)
5.	$P \to P$	(MP: 3, 4)

Example 55. Let us show that $\vdash \forall x[P(x) \to P(x)]$.

1.	$P(a) \to P(a)$	(See previous example)
2.	$(P(a) \to P(a)) \to ((Q(b) \to Q(b)) \to (P(a) \to P(a)))$	(Ax. 1)
3.	$(Q(b) \to Q(b)) \to (P(a) \to P(a))$	(MP: 1, 2)
4.	$(Q(b) \to Q(b)) \to \forall x[P(x) \to P(x)]$	(R$\forall$: 4)
5.	$Q(b) \to Q(b)$	(See previous example)
6.	$\forall x[P(x) \to P(x)]$	(MP: 4, 5)

Deduction

Definition 56 (Deduction)**.** A (formal) **deduction** of ψ from $\varphi_1, \ldots, \varphi_m$ is a finite list $\psi_1, \ldots \psi_k$ of formulas such that:

1. $\psi = \psi_k$ is the last formula in the list, and
2. each formula in the list is either:
   (a) one of $\varphi_1, \ldots, \varphi_m$,
   (b) one of the axioms, or
   (c) is obtained by an application of an inference rule to formulas preceding it in the list.

Definition 57. A formula ψ is (formally) **deducible** from $\varphi_1, \ldots, \varphi_m$ if and only if there exists a (formal) deduction of ψ from $\varphi_1, \ldots, \varphi_m$.

Notation: $\varphi_1, \ldots, \varphi_m \vdash \psi$.

When $m = 0$, the notions of deduction and deducibility reduce to the notions of proof and provable, respectively.

Example 58. Let us show that $R(a)$ is deducible from $P(a) \to (Q(a) \to R(a))$ and $P(a) \wedge Q(a)$, i.e.:

$$P(a) \to (Q(a) \to R(a)), P(a) \wedge Q(a) \vdash R(a)$$

1.	$P(a) \wedge Q(a)$	(Hyp.)
2.	$P(a) \to (Q(a) \to R(a))$	(Hyp.)
3.	$(P(a) \wedge Q(a)) \to P(a)$	(Ax. 4a)
4.	$P(a)$	(MP: 1, 3)
5.	$(P(a) \wedge Q(a)) \to Q(a)$	(Ax. 4b)
6.	$Q(a)$	(MP: 1, 5)
7.	$Q(a) \to R(a)$	(MP: 2, 4)
8.	$R(a)$	(MP: 6, 7)

Example 59. Let us show that $\forall x[P(x) \to Q(x)], P(c) \vdash Q(c)$.

1.	$\forall x[P(x) \to Q(x)]$	(Hyp.)
2.	$P(c)$	(Hyp.)
3.	$\forall x[P(x) \to Q(x)] \to (P(c) \to Q(c))$	(Ax. 11)
4.	$P(c) \to Q(c)$	(MP: 1, 3)
5.	$Q(c)$	(MP: 2, 4)

4.2 Natural deduction

Motivation

Hilbert-type proofs are difficult:

- What axiom should I choose?
- What rule should I use?

There are several other proof systems. The most known are:

- Natural deduction
- Resolution
- Tableaux

Natural deduction

Natural deduction intends to model the valid principles of reasoning used in informal proofs.

They are also designed to be relatively simple.

Therefore, natural deduction has an elegant but restricted collection of inference rules.

Many of the already know inferences steps (e.g., DeMorgan Laws) are not allowed as single steps; they must be justified in terms of more basic steps.

Conjunction rules

Conjunction elimination ($\wedge$ Elim):

$$
\begin{array}{l|l}
 & P_1 \wedge \cdots \wedge P_i \wedge \cdots \wedge P_n \\
 & \vdots \\
\triangleright & P_i
\end{array}
$$

It allows us to infer any conjunct P_i from a conjunctive sentence $P_1 \wedge \cdots \wedge P_i \wedge \cdots \wedge P_n$ that we have already obtained in the proof.

Conjunction introduction ($\wedge$ Intro):

$$
\begin{array}{l|l}
 & P_1 \\
 & \Downarrow \\
 & P_n \\
 & \vdots \\
\triangleright & P_1 \wedge \cdots \wedge P_n
\end{array}
$$

Conjunction introduction allows us to infer a conjunction $P_1 \wedge \cdots \wedge P_n$ provided we have already obtained each conjunct P_1 trough P_n in the proof. (The order the conjuncts have been proved is not important.)

Example 60. Give a natural deduction proof that $A \wedge B \wedge C \models B \wedge C$.

Or, equivalently, show that $A \wedge B \wedge C \vdash B \wedge C$ in natural deduction.

$$
\begin{array}{|ll}
1.\ A \wedge B \wedge C & \\
\hline
2.\ B & \wedge \text{ Elim: } 1 \\
3.\ C & \wedge \text{ Elim: } 1 \\
4.\ B \wedge C & \wedge \text{ Intro: } 2,\ 3
\end{array}
$$

Remark: The justification on the left is obligatory, so that we can verify that each proof step is correct.

Disjunction rules

Disjunction introduction (∨ Intro):

$$\begin{array}{l|l} & P_i \\ & \vdots \\ \triangleright & P_1 \vee \cdots \vee P_i \vee \cdots \vee P_n \end{array}$$

Disjunction introduction allows us to infer a disjunct $P_1 \vee \cdots \vee P_i \vee \cdots \vee P_n$ provided that we have already obtained at least on of the disjuncts in the proof. (The order of the disjuncts is not important.)

Disjunction elimination (∨ Elim):

$$\begin{array}{l|l} & P_1 \vee \cdots \vee P_n \\ & \vdots \\ & \quad \begin{array}{|l} \underline{P_1} \\ \vdots \\ S \end{array} \\ & \Downarrow \\ & \quad \begin{array}{|l} \underline{P_n} \\ \vdots \\ S \end{array} \\ & \vdots \\ \triangleright & S \end{array}$$

Disjunction elimination is a formal counterpart of "proof by cases".

It allows us to infer a sentence S from a disjunction $P_1 \vee \cdots \vee P_n$, if we have already obtained S from each disjunct P_1 through P_n individually, in a subproof.

Subproofs

A subproof is a proof that occurs within the context of a larger proof.

A subproof generally begins with an assumption (or hypothesis).

Unlike the premise of the main proof, the assumption of a subproof is temporary.

Throughout the course of the subproof, the assumption acts just like an additional premise.

After the subproof, the assumption of the subproof is no longer in force. We say that the assumption has been **discharged**.

Example 61. Give a natural deduction proof that $(A \wedge B) \vee (C \wedge D) \models B \vee D$.

1. $(A \wedge B) \vee (C \wedge D)$	
2. $A \wedge B$	
3. B	$\wedge$ Elim: 2
4. $B \vee D$	$\vee$ Intro: 3
5. $C \wedge D$	
6. D	$\wedge$ Elim: 5
7. $B \vee D$	$\vee$ Intro: 6
8. $B \vee D$	$\vee$ Elim: 1, 2–4, 5–7

Example 62. Give a natural deduction proof that $(B \wedge A) \vee (A \wedge C) \models A$.

1. $(B \wedge A) \vee (A \wedge C)$	
2. $B \wedge A$	
3. A	$\wedge$ Elim: 2
4. $A \wedge C$	
5. A	$\wedge$ Elim: 4
6. A	$\vee$ Elim: 1,2–3,4–5

Reiteration rule

Reiteration (Reit):

P

$\vdots$

$\triangleright$ P

The reiteration rule allows us to repeat a sentence that has already been obtained in the proof.

Example 63. Give a natural deduction proof that $(B \wedge A) \vee A \models A$.

1. $(B \wedge A) \vee A$	
2. $B \wedge A$	
3. A	$\wedge$ Elim: 2
4. A	
5. A	Reit: 4
6. A	$\vee$ Elim 1, 2–3, 4–5

Negation rules

Negation elimination ($\neg$ Elim):

$\neg\neg P$

$\vdots$

$\triangleright$ P

It allows us to infer P from $\neg\neg P$.

Negation introduction (¬ Intro):

$$\begin{array}{l|l} & \begin{array}{|l} \underline{P} \\ \vdots \\ \bot \end{array} \\ & \vdots \\ \triangleright & \neg P \end{array}$$

It corresponds to "proof by contradiction".

It allows us to infer $\neg P$ provided that we have achieved a contradiction under the assumption that P.

Contradiction rules

Contradiction introduction (⊥ Intro):

$$\begin{array}{l|l} & P \\ & \vdots \\ & \neg P \\ & \vdots \\ \triangleright & \bot \end{array}$$

Contradiction introduction allows us to obtain the contradiction symbol from a proof containing two contradictory sentences P and $\neg P$.

Contradiction elimination (⊥ Elim):

$$\begin{array}{l|l} & \bot \\ & \vdots \\ \triangleright & P \end{array}$$

Contradiction elimination allows us to infer any sentence P from the symbol $\bot$.

Example 64. Give a natural deduction proof that $A \models \neg\neg A$.

1. A	
2. $\neg A$	
3. $\bot$	⊥ Intro: 1,2
4. $\neg\neg A$	¬ Intro: 2–3

Example 65. Give a natural deduction proof that $A \vee B, \neg A, \neg B \models \bot$.

1. $A \vee B$	
2. $\neg A$	
3. $\neg B$	
4. A	
5. $\bot$	⊥ Intro: 2, 4
6. B	
7. $\bot$	⊥ Intro: 3, 6
8. $\bot$	∨ Elim: 1, 4–5, 6–7

Example 66. Give a natural deduction proof that $\neg(P \wedge Q) \models \neg P \vee \neg Q$.

1.	$\neg(P \wedge Q)$	
2.	$\neg(\neg P \vee \neg Q)$	
3.	$\neg P$	
4.	$\neg P \vee \neg Q$	$\vee$ Intro: 3
5.	$\perp$	$\perp$ Intro: 4, 2
6.	$\neg\neg P$	$\neg$ Intro: 3–5
7.	P	$\neg$ Elim: 6
8.	$\neg Q$	
9.	$\neg P \vee \neg Q$	$\vee$ Intro: 8
10.	$\perp$	$\perp$ Intro: 9, 2
11.	$\neg\neg Q$	$\neg$ Intro: 8–10
12.	Q	$\neg$ Elim: 11
13.	$P \wedge Q$	$\wedge$ Intro: 7, 12
14.	$\perp$	$\perp$ Intro: 1, 13
15.	$\neg\neg(\neg P \vee \neg Q)$	$\neg$ Intro: 2–14
16.	$\neg P \vee \neg Q$	$\neg$ Elim: 15

In fact, contradiction elimination is superfluous.

$\vdots$		
$n.$	$\perp$	
$n+1.$	$\neg P$	
$n+2.$	$\perp$	Reit: n
$n+3.$	$\neg\neg P$	$\neg$ Intro: $(n+1)$–$(n+2)$
$n+4.$	P	$\neg$ Elim: $n+3$
$\vdots$		

Conditionals rules

Conditional elimination ($\rightarrow$ Elim):

$P \rightarrow Q$
$\vdots$
P
$\vdots$
$\triangleright$ Q

Conditional elimination formalizes modus ponens. It allows us to infer Q from $P \rightarrow Q$ and P.

Conditional introduction ($\rightarrow$ Intro):

$$\begin{array}{l|l} & \quad P \\ & \quad \vdots \\ & \quad Q \\ \triangleright & P \to Q \end{array}$$

Conditional introduction is the formal counterpart of "conditional proof".

It allows us to infer $P \to Q$ provided we have proved Q under the assumption P.

Example 67. Give a natural deduction proof that $\models A \to \neg\neg A$.

1. A	
2. $\neg A$	
3. $\bot$	$\bot$ Intro: 1, 2
4. $\neg\neg A$	$\neg$ Intro: 2–3
5. $A \to \neg\neg A$	$\to$ Intro: 1–4

Biconditional elimination ($\leftrightarrow$ Elim):

$$\begin{array}{l|l} & P \leftrightarrow Q \\ & \vdots \\ & P \\ & \vdots \\ \triangleright & Q \end{array}$$

Biconditional introduction ($\leftrightarrow$ Intro):

$$\begin{array}{l|l} & \quad P \\ & \quad \vdots \\ & \quad Q \\ & \quad Q \\ & \quad \vdots \\ & \quad P \\ \triangleright & P \leftrightarrow Q \end{array}$$

Universal rules

Universal elimination ($\forall$ Elim):

$$\begin{array}{l|l} & \forall x[S(x)] \\ & \vdots \\ \triangleright & S(c) \end{array}$$

where c is an individual constant, and $S(c)$ stands for the result of replacing free occurrences of x in $S(x)$ with c.

This rule allows us to infer $S(c)$ for a particular constant c, given that we already have $S(x)$ for all x.

Universal introduction ($\forall$ Intro):

$$\begin{array}{l|l} & \quad \boxed{c} \\ & \quad \vdots \\ & \quad P(c) \\ \rhd & \forall x[P(x)] \end{array}$$

where c does not occur outside the subproof where it is introduced.

This rule allows us to generalise $P(x)$ to all x in the domain, given that we have already proved $P(c)$ for an arbitrary c.

Example 68. Give a natural deduction proof that $\forall x[P(x) \to Q(x)], \forall x[Q(x) \to R(x)] \models \forall x[P(x) \to R(x)]$.

1.	$\forall x[P(x) \to Q(x)]$	
2.	$\forall x[Q(x) \to R(x)]$	
3.	$\boxed{d}$	
4.	$P(d)$	
5.	$P(d) \to Q(d)$	$\forall$ Elim: 1, 3
6.	$Q(d)$	$\to$ Elim: 4, 5
7.	$Q(d) \to R(d)$	$\forall$ Elim: 2, 3
8.	$R(d)$	$\to$ Elim: 6, 7
9.	$P(d) \to R(d)$	$\to$ Intro: 4–8
10.	$\forall x[P(x) \to R(x)]$	$\forall$ Intro: 3–9

Note that constant d does not appear anywhere outside the subproof where it is introduced.

Existential rules

Existential introduction ($\exists$ Intro):

$$\begin{array}{l|l} & S(c) \\ & \vdots \\ \rhd & \exists x[S(x)] \end{array}$$

where c is an individual constant, and $S(c)$ stands for the result of replacing free occurrences of x in $S(x)$ with c.

Existential elimination ($\exists$ Elim):

$$\begin{array}{l|l} & \exists x[S(x)] \\ & \vdots \\ & \quad \boxed{c}\ \text{S(c)} \\ & \quad \vdots \\ & \quad Q \\ \rhd & Q \end{array}$$

where c does not occur outside the subproof where it is introduced.

Example 69. Give a natural deduction proof that $\forall x[C(x) \to L(x)], \forall x[L(x) \to O(x)], \exists x[C(x)] \models \exists x[L(x) \wedge O(x)]$.

1.	$\forall x[C(x) \to L(x)]$	
2.	$\forall x[L(x) \to O(x)]$	
3.	$\exists x[C(x)]$	
4.	$\boxed{e}\ C(e)$	
5.	$C(e) \to L(e)$	$\forall$ Elim: 1, 4
6.	$L(e)$	$\to$ Elim: 4, 5
7.	$L(e) \to O(e)$	$\forall$ Elim: 2, 4
8.	$O(e)$	$\to$ Elim: 6, 7
9.	$L(e) \land O(e)$	$\land$ Intro: 6, 8
10.	$\exists x[L(x) \land O(x)]$	$\exists$ Intro: 9
11.	$\exists x[L(x) \land O(x)]$	$\exists$ Elim: 3, 4–10

4.3 Exercises

Exercise 4.1. For each of the following, decide whether or not it is true. If it is, give a natural deduction proof. If it is not, give a counterexample.

1. $\forall x[C(x) \leftrightarrow S(x)], \forall x[C(x)] \models \forall x[S(x)]$
2. $\forall x[C(x)], \forall x[S(x)] \models \forall x[C(x) \land S(x)]$
3. $\neg\forall x[C(x)] \models \neg\forall x[C(x) \land S(x)]$

Exercise 4.2. For each of the following, decide whether or not it is true. If it is, give a natural deduction proof. If it is not, give a counterexample.

1. $\forall x[C(x) \lor T(x)], \exists x[\neg C(x)] \models \exists x[\neg T(x)]$
2. $\forall x[C(x) \lor T(x)], \exists x[\neg C(x)] \models \exists x[T(x)]$