CHAPTER 8

Classical Predicate Semantics and Proof Systems

ch8

1 Formal Predicate Languages

firstlan

Propositional languages are also called zero order languages, as opposed to predicate languages that are called first order languages. The same applies to the use of terms propositional and predicate logic; they are often called zero order and first order logics and we will use both terms equally.

We define a *predicate language* $\mathcal{L}$ following the pattern established by the propositional languages definitions. The predicate language $\mathcal{L}$ is more complicated in its structure and hence its *alphabet* $\mathcal{A}$ is much richer. The definition of its set of *formulas* $\mathcal{F}$ is more complicated. In order to define the set $\mathcal{F}$ we introduce an additional set $\mathbf{T}$, called a set of *terms* of the predicate language $\mathcal{L}$. We single out this set not only because we need it for the definition of formulas, but also because of its role in the development of other notions of predicate logic.

We will work with different predicate languages, depending on what applications we have in mind. All of these languages have some common features, and we begin with a following general definition.

def:predlan

Definition 1

By a **predicate language** $\mathcal{L}$ *we understand a triple*

$$\mathcal{L} = (\mathcal{A}, \mathbf{T}, \mathcal{F}), \tag{1}$$ pl

where $\mathcal{A}$ *is a predicate alphabet,* $\mathbf{T}$*, is the set of terms, and* $\mathcal{F}$ *is a set of formulas.*

The **components** of $\mathcal{L}$ are as follows.

1. **Alphabet** $\mathcal{A}$ is the set

$$\mathcal{A} = VAR \cup CON \cup PAR \cup \mathbf{Q} \cup \mathbf{P} \cup \mathbf{F} \cup \mathbf{C}, \tag{2}$$ alphabet

where VAR is set of predicate variables, CON is a set of propositional connectives, PAR a set of parenthesis, $\mathbf{Q}$ a set of quantifiers, $\mathbf{P}$, a set of predicate

symbols, **F** a set of functions symbols, and **C** a set of constant symbols. We assume that all of the sets defining the alphabet are disjoint.

Predicate Variables VAR

We assume that we always have a countably infinite set VAR of predicate variables, called usually **variables**. We denote variables by $x, y, z, ...$, with indices, if necessary, what we often express by writing

$$VAR = \{x_1, x_2, ...\}.$$

Propositional connectives CON

We define the set of propositional connectives CON in the same way as in the case of the propositional languages. It means that we assume that CON is *non-empty* and *finite set* and that consider only the connectives with one or two arguments, i.e.

$$CON = C_1 \cup C_2$$

where C_1 is a finite set (possibly empty) of unary connectives, C_2 is a finite set (possibly empty) of binary connectives of the language $\mathcal{L}$.

Parenthesis PAR

As in the propositional case, we adopt the signs (and) for our parenthesis., i.e. we define the set PAR as

$$PAR = \{(,)\}.$$

The set of propositional connectives CON defines a *propositional part* of the predicate logic language. What really differ one predicate language from the other is the choice of additional symbols to the symbols described above. These are called quantifiers symbols, predicate symbols, function symbols, and constant symbols. I.e. a particular predicate language is determined by specifying the following sets of symbols.

Quantifiers Q

We adopt two quantifiers; $\forall$ (for all, the universal quantifier) and $\exists$ (there exists, the existential quantifier), i.e. we have the following set of quantifiers

$$\mathbf{Q} = \{\forall, \exists\}.$$

In a case of the classical logic and the logics that extend it, it is possible to adopt only one quantifier and to define the other in terms of it and propositional connectives. It is impossible in a case of many non-classical logics, for example the intuitionistic logic. But even in the case of classical logic two quantifiers express

better the common intuition, so we assume that we have two of them.

Predicate symbols P

Predicate symbols represent relations. We assume that we have an non empty, finite or countably infinite set **bf P** of predicate, or relation symbols. We denote predicate symbols by $P, Q, R, ...$, with indices, if necessary, what we often express by writing

$$\mathbf{P} = \{P_1, P_2, ...\}.$$

Each predicate symbol $P \in \mathbf{P}$ has a positive integer $\#P$ assigned to it; if $\#P = n$ then say P is called an n-ary (n - place) predicate (relation) symbol.

Function symbols F

We assume that we have a finite (may be empty) or countably infinite set **F** of function symbols. When the set **F** is empty we say that we deal with a *language without functional symbols.* We denote functional symbols by $f, g, h, ...$, with indices, if necessary, what we often express by writing

$$\mathbf{F} = \{f_1, f_2, ...\}.$$

Similarly, as in the case of predicate symbols, each function symbol $f \in \mathbf{F}$ has a positive integer $\#f$ assigned to it; if $\#f = n$ then say f is called an n-ary (n - place) function symbol.

Constant symbols C

We also assume that we have a finite (may be empty) or countably infinite set **C** of constant symbols. The elements of **C** are denoted by $c, d, e...$, with indices, if necessary, what we often express by writing

$$\mathbf{C} = \{c_1, c_2, ...\}.$$

When the set **C** is empty we say that we deal with a *language without constant symbols.*

Sometimes the constant symbols are defined as 0-ary function symbols, i.e. $\mathbf{C} \subseteq \mathbf{F}$. We single them out as a separate set for our convenience.

Observe that what distinguishes now one predicate language $\mathcal{L}$ form the pother is the choice of the components CON, and **P, F, C** of its alphabet $\mathcal{A}$. We hence will write

$$\mathcal{L}_{CON}(\mathbf{P}, \mathbf{F}, \mathbf{C}) \qquad (3) \quad \boxed{\texttt{pl1}}$$

to denote the predicate language $\mathcal{L}$ determined by **P, F, C** and the set of propositional connectives CON.

Once the set of propositional connectives is fixed, the predicate language is determined by the sets **P, F** and **C** and we l write

$$\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C}) \tag{4}$$ pl2

for the predicate language $\mathcal{L}$ determined by **P, F** and **C** (with a fixed set of propositional connectives). If there is no danger of confusion, we may abbreviate $\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ to just $\mathcal{L}$.

We sometimes allow the same symbol to be used as an n-place relation symbol, and also as an m-place one; no confusion should arise because the different uses can be told apart easily. Similarly for function symbols.

Having defined the basic elements of syntax, the alphabet, we can now complete the formal definition of the predicate language by defining two more complex components: the set **T** of all terms and the set $\mathcal{F}$ of all well formed formulas of the language $\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$.

2. Terms T

The set **T** of terms of a predicate language $\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ is defined as follows.

terms **Definition 2 (Terms)**

*Given a predicate language $\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ with an alphabet $\mathcal{A}$. The set **T** of terms of $\mathcal{L}$ is the smallest set $T \subseteq \mathcal{A}^*$ meeting the conditions:*

(i) any variable is a term, i.e. $VAR \subseteq \mathbf{T}$;

(ii) any constant symbol is a term, i.e. $\mathbf{C} \subseteq \mathbf{T}$;

(iii) if f is an nplace function symbol, i.e. $f \in \mathbf{F}$ and $\#f = n$ and $t_1, t_2, ..., t_n \in \mathbf{T}$, then $f(t_1, t_2, ..., t_n) \in \mathbf{T}$.

Example 1

Let $f \in \mathbf{F}, \#f = 1$, i.e. f is a one place function symbol. Let x, y be predicate variables, c, d constants, i.e. $x, y \in VAR$, $c, d \in \mathbf{C}$. The following expressions are terms:

$$x,\ y,\ f(x),\ f(y),\ f(c),\ f(d),\ f(f(x),)\ f(f(y)),\ ff((c)),\ f(f(d)), ...etc.$$

Example 2

*If $\mathbf{F} = \emptyset, \mathbf{C} = \emptyset$, then the set **T** of terms consists of variables only, i.e.*

$$\mathbf{T} = VAR = \{x_1, x_2, ...\}.$$

From the above we get the following observation.

Remark 1

For any predicate language $\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$, the set $\mathbf{T}$ of its terms is always non-empty.

Example 3

If $f \in \mathbf{F}, \#f = 1$, $g \in \mathbf{F}, \#g = 2$, $x, y \in VAR, c, d \in \mathbf{C}$, then some of the terms are the following:

$$f(g(x,y)),\ f(g(c,x)),\ g(f(f(c)), g(x,y)),\ g(c, g(x, f(c))).$$

From time to time, the logicians are and we may be informal about how we write terms. For instance, if we denote a two place function symbol g by $+$, we may write $x + y$ instead $+(x, y)$. Because in this case we can think of $x + y$ as an unofficial way of designating the "real" term $+(x, y)$, or even $g(x, y)$.

2. **Formulas $\mathcal{F}$**

Before we define the set of formulas, we need to define one more set; the set of atomic, or elementary formulas. They are the "smallest" formulas as were the propositional variables in the case of propositional languages.

Atomic formulas

An atomic formula of a predicate language $\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ is any element of the alphabet $\mathcal{A}^*$ of the form

$$R(t_1, t_2, ..., t_n),$$

where $R \in \mathbf{P}, \#R = n$, i.e. R is n-ary relational symbol and $t_1, t_2, ..., t_n$ are terms. The set of all atomic formulas is denoted by $A\mathcal{F}$ and is defines as

$$A\mathcal{F} = \{R(t_1, t_2, ..., t_n) \in \mathcal{A}^* : R \in \mathbf{P},\ t_1, t_2, ..., t_n \in \mathbf{T},\ \#R = n,\ n \geq 1\}. \quad (5)$$ aform

Example 4

Consider a language

$$\mathcal{L}(\emptyset, \{P\}, \emptyset),$$

for $\#P = 1$, i.e. a language without neither functional, nor constant symbols, and with one, one-place predicate symbol P. The set of atomic formulas contains all formulas of the form $P(x)$, for x any variable, i.e.

$$A\mathcal{F} = \{P(x) : x \in VAR\}.$$

Example 5

Let now

$$\mathcal{L} = \mathcal{L}(\{f, g\}, \{R\}, \{c, d\}),$$

for #f = 1, #g = 2 , #R = 2, i.e. $\mathcal{L}$ has two functional symbols: one -place symbol f and two-place symbol g; one two-place predicate symbol R, and two constants: c,d. Some of the atomic formulas in this case are the following.

$$R(c, d),\ R(x, f(c)),\ R(f(g(x, y)), f(g(c, x))),\ R(y, g(c, g(x, f(c)))).$$

Given a predicate language

$$\mathcal{L} = \mathcal{L}_{CON}(\mathbf{P}, \mathbf{F}, \mathbf{C}),$$

where CON is *non-empty, finite set* of propositional connectives such that $CON = C_1 \cup C_2$ for C_1 a finite set (possibly empty) of unary connectives, C_2 a finite set (possibly empty) of binary connectives of the language $\mathcal{L}$. We define the set $\mathcal{F}$ of all well formed formulas of the predicate language $\mathcal{L}$ as follows.

def:form **Definition 3 (Formulas)**

The set $\mathcal{F}$ of all well formed formulas, called shortly set of formulas, of the language $\mathcal{L}_{CON}\mathbf{P}, \mathbf{F}, \mathbf{C})$ is the smallest set meeting the following conditions:

1. *any atomic formula of $\mathcal{L}_{CON}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ is a formula, i.e.*

$$A\mathcal{F} \subseteq \mathcal{F};$$

2. *if A is a formula of $\mathcal{L}_{CON}(\mathbf{P}, \mathbf{F}, \mathbf{C})$, $\bigtriangledown$ is an one argument propositional connective, then $\bigtriangledown A$ is a formula of $\mathcal{L}_{CON}(\mathbf{P}, \mathbf{F}, \mathbf{C})$, i.e. if the following recursive condition holds*

$$\textit{if} \quad A \in \mathcal{F}, \quad \bigtriangledown \in C_1, \quad \textit{then} \quad \bigtriangledown A \in \mathcal{F};$$

3. *if A, B are formulas of $\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$, $\circ$ is a two argument propositional connective, then $(A \circ B)$ is a formula of $\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$, i.e. if the following recursive condition holds*

$$\textit{if} \quad A \in \mathcal{F}, \quad \bigtriangledown \in C_2, \quad \textit{then} \quad (A \circ B) \in \mathcal{F};$$

4. *if A is a formula of $\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ and x is a variable, then $\forall xA, \exists xA$ are formulas of $\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$, i.e. if the following recursive condition holds*

$$\textit{if} \quad A \in \mathcal{F}, \quad x \in VAR, \quad \forall, \exists \in \mathbf{Q}, \quad \textit{then} \quad \forall xA, \quad \exists xA \in \mathcal{F}.$$

In formulas $\forall xA$, $\exists xA$, the formula A is in the **scope of the quantifier** $\forall$, $\exists$, *respectively.*

Example 6

Let $\mathcal{L}$ be a language with with the set $\{\cap, \cup, \Rightarrow, \neg\}$ of connectives and with two functional symbols: one -place and one two-place, one two-place predicate symbol, and two constants. We write $\mathcal{L}$ as

$$\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}(\{R\}, \{f, g\}, \{c, d\},)$$

where $\#f = 1$, $\#g = 2$, $\#R = 2$. Some of the formulas of $\mathcal{L}$ are the following.

$$R(c, f(d)), \quad \exists x R(x, f(c)), \quad \neg R(x, y), \quad \forall z(\exists x R(x, f(c)) \Rightarrow \neg R(x, y)),$$

$$(R(c, d) \cap \exists x R(x, f(c))), \quad \forall y R(y, g(c, g(x, f(c)))), \quad \forall y \neg \exists x R(x, y).$$

The formula $R(x, f(c))$ is in a **scope** of the quantifier $\exists x$ in $\exists x R(x, f(c))$.

The formula $(\exists x R(x, f(c)) \Rightarrow \neg R(x, y))$ isn't in a **scope** of any quantifier.

The formula $(\exists x R(x, f(c)) \Rightarrow \neg R(x, y))$ is in the **scope** of $\forall$ in $\forall z(\exists x R(x, f(c)) \Rightarrow \neg R(x, y))$.

modal **Example 7**

Let $\mathcal{L}$ be a language with with the set $\{\neg, \Box, \Diamond, \cap, \cup, \Rightarrow\}$ of connectives and **P**, **F**, *and* **C** *the same as in previous exercise, i.e.*

$$\mathcal{L} = \mathcal{L}_{\{\neg,\Box,\Diamond,\cap,\cup,\Rightarrow\}}(\{R\}, \{f, g\}, \{c, d\},)$$

where $\#f = 1$, $\#g = 2$, $\#R = 2$.

$\mathcal{L}$ is now a language of some first order **modal** *logic. Some of the formulas of $\mathcal{L}$ are the following.*

$$\Diamond \neg R(c, f(d)), \quad \Diamond \exists x \Box R(x, f(c)), \quad \neg \Diamond R(x, y), \quad \forall z(\exists x R(x, f(c)) \Rightarrow \neg R(x, y)),$$

$$(R(c, d) \cap \exists x R(x, f(c))), \quad \forall y \Box R(y, g(c, g(x, f(c)))), \quad \Box \forall y \neg \Diamond \exists x R(x, y).$$

The formula $\Box R(x, f(c))$ is in a **scope** of the quantifier $\exists x$ in $\Diamond \exists x \Box R(x, f(c))$.

The formula $(\exists x R(x, f(c)) \Rightarrow \neg R(x, y))$ isn't in a **scope** of any quantifier.

The formula $(\exists x R(x, f(c)) \Rightarrow \neg R(x, y))$ is in the **scope** of $\forall z$ in $\forall z(\exists x R(x, f(c)) \Rightarrow \neg R(x, y))$. Formula $\neg \Diamond \exists x R(x, y)$ is in the **scope** of $\forall y$ in $\Box \forall y \neg \Diamond \exists x R(x, y)$.

Given a predicate language $\mathcal{L} = (\mathcal{A}, T, \mathcal{F})$, we must distinguish between formulas like $P(x, y)$, $\forall x P(x, y)$ and $\forall x \exists y P(x, y)$.

This is done by introducing the notion of **free** and **bound variables**, **open** and **closed** formulas (sentences). Before we formulate proper definitions, here are some simple observations.

1. Some formulas are without quantifiers.
For example formulas $R(c_1, c_2),\ R(x, y),\ (R(y, d) \Rightarrow R(a, z))$. A formula without quantifiers is called an **open formula**.

Variables x, y in $R(x, y)$ are called **free variables**. The variables y in R(y, d) and z in R(a,z) are also **free**.

2. Quantifiers **bind** variables within formulas.

The variable x is **bounded** by $\exists x$ in the formula $\exists x R(x, y)$, the variable y is **free**. The variable y is **bounded** by $\forall y$ in the formula $\forall y R(x, y)$, the variable y is **free**.

3. The formula $\exists x \forall y R(x, y)$ does not contain any free variables, neither does the formula $R(c_1, c_2)$. A formula without any free variables is called a **closed formula** or a **sentence**.

Sometimes in order to distinguish more easily *which variable* is **free** and which is **bound** in the formula we might use the bold face type for the quantifier bound variables and write the formulas as follows.

$$(\forall \mathbf{x} Q(\mathbf{x}, y), \quad \exists \mathbf{y} P(\mathbf{y}), \quad \forall \mathbf{y} R(\mathbf{y}, g(c, g(x, f(c))))),$$

$$(\forall \mathbf{x} P(\mathbf{x}) \Rightarrow \exists \mathbf{y} Q(x, \mathbf{y})), \quad (\forall \mathbf{x}(P(\mathbf{x}) \Rightarrow \exists \mathbf{y} Q(\mathbf{x}, \mathbf{y})))$$

Observe that the formulas $\exists \mathbf{y} P(\mathbf{y}),\ (\forall \mathbf{x}(P(\mathbf{x}) \Rightarrow \exists \mathbf{y} Q(\mathbf{x}, \mathbf{y})))$ are **closed**. We call a close formula a **sentence**.

Example 8

Consider atomic formulas: $P(y), Q(x, c), R(z), P_1(g(x, y), z)$. *Here are some non atomic formulas formed out of them.*

1. $(P(y) \cup \neg Q(x, c)) \in \mathcal{F}$. *This is an* **open** *formula A with two free variables x,y. We denote A this as formula* $A(x, y)$.

2. $\exists \mathbf{x}(P(y) \cup \neg Q(\mathbf{x}, c)) \in \mathcal{F}$. *We write* $\mathbf{x}$ *to denote that x is a* **bound** *variable. The variable y is* **free**. *This is a formula B with one free variable y. We denote B as a formula* $B(y)$.

3. $\forall \mathbf{y}(P(\mathbf{y}) \cup \neg Q(x, c)) \in \mathcal{F}$. *The variable y is* **bound**, *the variable x is* **free**. *We denote this formula by for example* $A_1(x)$.

4. $\forall \mathbf{y} \exists \mathbf{x}(P(y) \cup \neg Q(\mathbf{x}, c)) \in \mathcal{F}$ *has no free variables. It is a* **closed** *formula called also a* **sentence**.

Exercise 1

Given the following formulas of $\mathcal{L}$*:*

$P(x, f(c, y)),\ \exists c P(x, f(c, y)),\ \forall x f(x, P(c, y)),\ \exists x P(x, f(c, y)) \Rightarrow \forall y P(x, f(c, y))$.

1. Indicate whether they are, or are not well formed formulas of $\mathcal{F}$. For those which are not in $\mathcal{F}$ write a correct formula.
2. For each correct, or corrected formula identify all components: connectives, quantifiers, predicate and function symbols, and list all its terms.
3. For each formula identify its s free and bound variables. State which are open and which are closed formulas (sentences), if any.

Solution
Formula $A_1 = P(x, f(c, y))$.
It is a correct **atomic** formula. P is a 2 argument *predicate* symbol, f is a 2 argument *function* symbol, c is a *constant*. We write it symbolically: $P \in \mathbf{P},\ f \in \mathbf{F},\ c \in \mathbf{C}$. It is an **open** formula with two *free variables* x,y. We denote it by $A_1(x, y)$. It has no *bound variables*.

Formula $A_2 = \exists c P(x, f(c, y))$.

It is a not a correct formula, i.e. $\exists c P(x, f(c, y)) \notin \mathcal{F}$. The expression $\exists c$ has no meaning because c is a constant, not a variable.

The corrected formulas are: $B_1 = \exists x P(x, f(c, y)),\ B_2 = \exists y P(x, f(c, y))$, and formulas $B = \exists z P(z, f(c, y))$ for any variable z different then x and y.

None of the correct formulas are open. Variable y is free in $B_1 = B_1(y)$, variable x is free in $B_2 = B_2(x)$, both variables x and y are free in all formulas $B = B(x, y)$. All formulas are nether close, nor open. The *terms* appearing in any of them are the same as in $A_1 = P(x, f(c, y))$ and are: $x, y, c, f(c, y)$.

Formula $A_3 = \forall x f(x, P(c, y))$.

It is a not a correct formula, i.e. $\forall x f(x, P(c, y)) \notin \mathcal{F}$. The function symbol f in front $f(x, P(c, y))$ indicate a term and terms are not formulas. Moreover, the atomic formula $P(c, y)$ can't be put inside a term!

Formula $A_4 = \exists x P(x, f(c, y)) \Rightarrow \forall y P(x, f(c, y))$.

It is a not a correct formula. The correct formula is $A = (\exists x P(x, f(c, y)) \Rightarrow \forall y P(x, f(c, y)))$. It has two free variables x and y and we write it as $A = A(x, y)$.

Informally, in the formula $P(x, y)$ both variables x and y are called **free** variables. They are not in the scope of any quantifier. The formula of that type (without quantifiers) is an **open** formula.

The formal definition of the set of free variables of a formula is the following.

bfvar **Definition 4 (Free and Bound Variables)**

The set $FV(A)$ of free variables of a formula A is defined by the induction of the degree of the formula as follows.

1. *If A is an atomic formula, i.e. $A \in \mathcal{AF}$, then $FV(A)$ is just the set of variables appearing in the expression A;*

2. *for any unary propositional connective, i.e any* $\bigtriangledown \in C_1$,
   $FV(\bigtriangledown A) = FV(A)$,
   i.e. the free variables of $\bigtriangledown A$ *are the free variables of* A*;*

3. *for any binary propositional connective, i.e any* $\circ \in C_2$,
   $FV(A \circ B) = FV(A) \cup FV(B)$,
   i.e. the free variables of $(A \circ B)$ *are the free variables of* A *together with the free variables of* B*;*

4. $FV(\forall xA) = FV(\exists xA) = FV(A) - \{x\}$,
   i.e. the free variables of $\forall xA$ *and* $\exists xA$ *are the free variables of* A*, except for* x*.*

A formula with no free variables is called a **sentence**.
A variable is called **bound** *if it is* **not free**.
A formula with no bound variables is called an **open formula**.

Example 9 *The formulas* $\exists xQ(c, g(x, d))$, $\neg\forall x(P(x) \Rightarrow \exists y(R(f(x), y) \cap \neg P(c)))$ *are* **sentences**. *The formulas* $Q(c, g(x, d))$, $\neg(P(x) \Rightarrow (R(f(x), y) \cap \neg P(c)))$ *are* **open** *formulas. The formulas* $\exists xQ(c, g(x, y))$, $\neg(P(x) \Rightarrow \exists y(R(f(x), y) \cap \neg P(c)))$ *are* **neither** *sentences* **nor** *open formulas. They contain some free and some bound variables; the variable* y *is free in the first formula, the variable* x *is free in the second.*

The definition 1 defines a predicate language $\mathcal{L} = \mathcal{L}_{CON}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ (3) with its sets of predicate, function and constant symbol possibly countably infinite sets. We use its most general case with sets of predicate, function and constant symbol all countably infinite sets for defining all relevant notions concerning provability and semantics. In particular, we will define in detail the classical semantics for this most general form of $\mathcal{L}$ and prove the completeness theorem for classical predicate logic based on it.

When we deal with formal theory $\mathbf{Th}(SA)$ with a set SA of specific axioms we restrict the language $\mathcal{L}_{CON}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ to the symbols characteristic for that theory. We hence introduce the following definition.

langtheory **Definition 5**

Given a language $\mathcal{L}_{CON}(\mathbf{P}, \mathbf{F}, \mathbf{C}) = (\mathcal{A}, \mathbf{T}, \mathcal{F})$.

Let $\mathcal{F}_0 \subseteq \mathcal{F}$ *be a non-empty, finite subset of formulas of* $\mathcal{L}$. *Denote by* $\mathbf{P}_0$, $\mathbf{F}_0, \mathbf{C}_0$ *the sets of all predicate, function, and constant symbols appearing in the formulas from the set* $\mathcal{F}_0$. *The language*

$$\mathcal{L}_{CON}(\mathbf{P}_0, \mathbf{F}_0, \mathbf{C}_0)$$

is called a **language defined** *by the set $\mathcal{F}_0$ of formulas.*

Example 10 *Consider a language $\mathcal{L} = \mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}}(\mathbf{P},\mathbf{F},\mathbf{C})$ and a following set $\mathcal{F}_0$ of formulas of $\mathcal{L}$*

$\mathcal{F}_0 = \{\exists x Q(c, g(x, d)),\ \neg\forall x(P(x) \Rightarrow \exists y(R(f(x), y) \cap \neg P(e))), \neg(F(a) \cap R(y, h(c))\}$.

A **language defined** *by the set $\mathcal{F}_0$ of formulas is*

$$\mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}}(\{P, R, Q, F\},\ \{g, f, h\},\ \{a, c, d, e\}),$$

where # Q = #R = 2, #P =# F = 1, # g = 2, #f = # h = 1.

It is common practice to use the notation

$$A(x_1, x_2, ..., x_n) \tag{6}$$ fvar

to indicate that $FV(A) \subseteq \{x_1, x_2, ..., x_n\}$ without implying that all of $x_1, x_2, ..., x_n$ are actually free in A. This is similar to the practice in algebra of writing $p(x_1, x_2, ..., x_n)$ for a polynomial p in the variables $x_1, x_2, ..., x_n$ without implying that all of them have nonzero coefficients.

x/t **Definition 6 (Replacing x by t in A)**

If $A(x)$ is a formula, and t is a term then $A(x/t)$ or, more simply, $A(t)$ denotes the result of replacing all occurrences of the free variable x by the term t throughout. When using the notation $A(t)$ we always assume that none of the variables in t occur as bound variables in A.

The assumption that none of the variables in t occur as bound variables in A is essential, otherwise by substituting t on the place of x we would distort the meaning of $A(t)$. Let $t = y$ and $A(x)$ is $\exists y(x \neq y)$, i.e. the variable y in t is bound in A. The substitution of t for x produces a formula $A(t)$ of the form $\exists y(y \neq y)$, which has a different meaning than $\exists y(x \neq y)$.

But if $t = z$, i.e. the variable z in t is not bound in A, then $A(x/t) = A(t)$ is $\exists y(z \neq y)$ and express the same meaning as $A(x)$.

Remark that if for example $t = f(z, x)$ we obtain $\exists y(f(z, x) \neq y)$ as a result of substitution of $t = f(z, x)$ for x in $\exists y(x \neq y)$.

This notation is convenient because we can agree to write as

$$A(t_1, t_2, ..., t_n) \quad \text{or} \quad A(x_1/t_1, x_2/t_2, \ldots, x_n/t_n)$$

a result of substituting in A the terms $t_1, t_2, \ldots, t_n$ for all free occurrences (if any) of $x_1, x_2, \ldots, x_n$, respectively. But when using this notation we always assume that none of the variables in $t_1, t_2, ..., t_n$ occur as bound variables in A.

The above assumption that none of the variables in $t_1, t_2, ..., t_n$ occur as bound variables in A is often expressed using the notion: *$t_1, t_2, ..., t_n$ are free for all theirs variables in A* which is defined formally as follows.

termfree **Definition 7 (Term t free for y in A)**

If $A \in \mathcal{F}$ and t is a term, then t is said to be **free for** *y if no free occurrence of y lies within the scope of any quantifier bounding variables in t.*

Example 11 *Let A , B be the formulas*

$$\forall y P(f(x,y),y), \quad \forall y P(f(x,z),y),$$

respectively. The term $t = f(x,y)$ is free for x and is not free for y in A. The term $t = f(x,z)$ is free for x and z in B. The term $t = y$ is not free neither for x nor for z in A, B.

Example 12

Let A be a formula

$$(\exists x Q(f(x), g(x,z)) \cap P(h(x,y),y)).$$

The term $t_1 = f(x)$ is not free for x in A; the term $t_2 = g(x,z)$ is free for z only, term $t_3 = h(x,y)$ is free for y only because x occurs as a bound variable in A; term t_4.

Definition 8 (Replacement)

If $A(x), A(x_1, x_2, ..., x_n) \in \mathcal{F}$ and $t, t_1, t_2, ..., t_n \in T$, then $A(x/t)$, $A(x_1/t_1, x_2/t_2, \ldots, x_n/t_n)$ or, more simply just

$$A(t), \ A(t_1, t_2, ..., t_n)$$

denotes the result of replacing all occurrences of the free variables x, $x_1, x_2, ..., x_n$, by the terms t, $t, t_1, t_2, ..., t_n$, respectively, assuming that $t, t_1, t_2, ..., t_n$ are free for all theirs variables in A.

Classical Restricted Domain Quantifiers

We often use logic symbols, while writing mathematical statements. For example mathematicians in order to say "all natural numbers are greater then zero and some integers are equal 1" often write it as

$$x \geq 0, \forall_{x \in N} \ \text{ and } \ \exists_{y \in Z}, \ y = 1.$$

Some of them, who are more "logic oriented", would also write it as

$$\forall_{x \in N} \ x \geq 0 \ \cap \ \exists_{y \in Z} \ y = 1,$$

or even as

$$(\forall_{x \in N} \ x \geq 0 \ \cap \ \exists_{y \in Z} \ y = 1).$$

None of the above symbolic statements are formulas of the *predicate language* $\mathcal{L}$. These are mathematical statement written with mathematical and logic symbols. They are written with different degree of "logical precision", the last being, from a logician point of view the most precise.

Observe that the quantifiers in $\forall_{x\in N}$ and $\exists_{y\in Z}$ used in all of them are not the one used in the predicate language $\mathcal{L}$, which admits only quantifiers $\forall x$ and $\exists y$, for any variables $x, y \in VAR$. The quantifiers $\forall_{x\in N},\ \exists_{y\in Z}$ are called **quantifiers with restricted domain**. The first is restricted to the domain of natural numbers, the second to the integers. The restriction of the quantifier domain can, and often is given by more complicated statements. For example we say "for all $x > 2$" and write $\forall_{x>2}$, or we say "exists $x > 2$ and at same time $x + 2 < 8$" and write symbolically $\exists_{(x>2\cap x+2<8)}$.

Our goal now is to correctly "translate " mathematical and natural language statement into formulas of the predicate language $\mathcal{L}$ of the classical predicate logic with the the set $\{\neg.\cap, \cup, \Rightarrow\}$ of propositional connectives. We say "classical predicate logic" to express that we define all notions for the classical semantics to be defined formally in the next section 2. One can extend these notions to non-classical logics, but we describe and will talk only about classical case. We introduce the quantifiers with restricted domain into the classical predicate logic language by expressing them within the language $\mathcal{L}$ as follows.

def:rq **Definition 9**

Given a classical predicate logic language

$$\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P}, \mathbf{F}, \mathbf{C}).$$

The quantifiers $\forall_{A(x)},\ \exists_{A(x)}$ *are called* **quantifiers with restricted domain**, *or* **restricted quantifiers**, *where* $A(x) \in \mathcal{F}$ *is any formula with any free variable* $x \in VAR$.

A formula $\forall_{A(x)}B(x)$ *is an abbreviation of a formula* $\forall x(A(x) \Rightarrow B(x)) \in \mathcal{F}$. *We write it symbolically as*

$$\forall_{A(x)}\ B(x) = \forall x(A(x) \Rightarrow B(x)). \tag{7}$$ resq1

A formula $\exists_{A(x)}B(x)$ *stands for a formula* $\exists x(A(x) \cap B(x)) \in \mathcal{F}$. *We write it symbolically as*

$$\exists_{A(x)}\ B(x) = \exists x(A(x) \cap B(x)) \tag{8}$$ resq2

The definition 9 of restricted quantifiers is obviously faithful to our intuitive meaning of quantifiers. We use informally a symbol = to stress that we they are in a sense equivalent with respect to classical semantics. We call (7) and (8) **transformations rules** for restricted quantifiers.

Exercise 2

Given a mathematical statement **S** *written with logical symbols*

$$(\forall_{x\in N}\ x \geq 0\ \cap\ \exists_{y\in Z}\ y = 1)$$

1. Translate it into a proper logical formula with restricted domain quantifiers i.e. into a formula of $\mathcal{L}$ *that uses the restricted domain quantifiers.*

2. Translate your restricted domain quantifiers logical formula into a correct logical formula **without** *restricted domain quantifiers, i.e. into a formula of* $\mathcal{L}$*.*

Solution
We proceed to write this and other similar problems solutions in a sequence of steps.

STEP 1. We identify *basic statements* in **S**, i.e. mathematical statements that involve only relations. They will be translated into *atomic formulas*. We identify the *relations* in the basic statements and choose the *predicate symbols* as their names. We identify all *functions* and *constants* (if any) in the basic statements and choose the *function symbols* and *constant symbols* as their names.

The basic statements in **S** are: $x \in N,\ x \geq 0,\ y \in Z,\ y = 1$. The relations are: $\in N,\ \in Z,\ \geq,\ =$. We use one argument predicates symbols N, Z for $\in N, \in Z$, respectively. We use two argument predicate symbols G for $\geq$, and E for $=$. There are no functions. We have two constant symbols c_1, c_2 for numbers 0 and 1, respectively.

STEP 2. We write the basic statements as *atomic formulas* of $\mathcal{L}$.

We write $N(x), Z(x)$ for $x \in N, x \in Z$, respectively. $G(x, c_1)$ for $x \geq 0$ and $E(y, c_2)$ for $y = 1$. These are all atomic formulas.

STEP 3. We re-write the statement **S** a logical formula with restricted domain quantifiers.

The statement **S** becomes a restricted quantifiers formula:

$$(\forall_{N(x)}\ G(x, c_1)\ \cap\ \exists_{Z(y)}\ E(y, c_2)).$$

STEP 4. We apply (7) and (8) to the formula from STEP 3. and obtain a formula A of $\mathcal{L}$ as a representation of the given mathematical statement **S**.

A formula $A \in \mathcal{F}$ that corresponds to **S** is

$$(\forall x\ (N(x) \Rightarrow G(x, c_1))\ \cap\ \exists y\ (Z(y) \cap E(y, c_2))).$$

Here is a perfectly acceptable short solution. We presented the long solution in order to explain all steps needed to be performed when one writes a solution.

Short Solution
The *basic statements* in **S** are: $x \in N,\ x \geq 0,\ y \in Z, y = 1$. The corresponding *atomic formulas* of $\mathcal{L}$ are: $N(x),\ G(x, c_1),\ Z(y),\ E(y, c_2)$, respectively.

The statement **S** becomes becomes a restricted quantifiers formula $(\forall_{N(x)} G(x, c_1)\ \cap\ \exists_{Z(y)}\ E(y, c_2))$. Applying restricted quantifiers definition 9 and transformation rules (7), (8) we get a following formula $A \in \mathcal{F}$

$$(\forall x(N(x) \Rightarrow G(x, c_1)) \cap \exists y(Z(y) \cap E(y, c_2))).$$

2 Classical Semantics

sec:clsem

The notion of predicate tautology is much more complicated then that of the propositional. Predicate tautologies are also called *valid formulas*, or *laws of quantifiers* to distinguish them from the propositional case. The formulas of a predicate language $\mathcal{L}$ have meaning only when an *interpretation* is given for all its symbols. We define an interpretation I by interpreting predicate, functional symbols as a concrete relation, function defined in a certain set $U \neq \emptyset$, and constants symbols as elements of the set U. The set U is called the *universe* of the *interpretation* I. These two items specify a *structure* $\mathbf{M} = (U, I)$ for the language $\mathcal{L}$.
.

The semantics for a first order language $\mathcal{L}$ in general, and for the first order classical logic in particular, is defined, after Tarski (1936) in terms of the *structure* $\mathbf{M} = [U, I]$, an assignment s of $\mathcal{L}$, and a *satisfaction relation* $(\mathbf{M}, s) \models A$ between structures, assignments and formulas of $\mathcal{L}$.

The definition of a structure $\mathbf{M} = [U, I]$ and the assignment s of $\mathcal{L}$ is common for different predicate languages and for different semantics and we define them as follows.

structure

Definition 10 (Structure)

Given a predicate language $\mathcal{L} = \mathcal{L}_{CON}(\mathbf{P}, \mathbf{F}, \mathbf{C})$. A **structure** *for $\mathcal{L}$ is a pair*

$$\mathbf{M} = [U, I],$$

where U is a non empty set called a **universe** *and I is an assignment called an* **interpretation** *of the language $\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ in the universe U defined as follows.*

1. *I assigns to any predicate symbol* $P \in \mathbf{P}$ *a relation* P_I *defined in the universe* U. *I.e. for any* $P \in \mathbf{P}$, *if* $\#P = n$, *then*

$$P_I \subseteq U^n.$$

2. *I assigns to any functional symbol* $f \in \mathbf{F}$ *a function* f_I *defined in the universe* U. *I.e. for any* $f \in \mathbf{F}$, *if* $\#f = n$, *then*

$$f_I : \quad U^n \longrightarrow U.$$

3. *I assigns to any constant symbol* $c \in \mathbf{C}$ *an element* c_I *of the universe. I.e for any* $c \in \mathbf{C}$,

$$c_I \in U.$$

Example 13

Let $\mathcal{L}$ *be a language with one two-place predicate symbol, two functional symbols: one -place and one two-place, and two constants, i.e.*

$$\mathcal{L} = \mathcal{L}(\{R\}, \{f, g\}, \{c, d\},)$$

where $\#R = 2$, $\#f = 1$, $\#g = 2$, *and* $c, d \in \mathbf{C}$.

We define a structure $\mathbf{M}$ *= [U, I] as follows. We take as the universe the set* $U = \{1, 3, 5, 6\}$. *The predicate* R *is interpreted as* $\leq$, *what we write as* $R_I : \leq$. *We interpret* f *as a function* $f_I : \{1, 3, 5, 6\} \longrightarrow \{1, 3, 5, 6\}$ *such that* $f_I(x) = 5$ *for all* $x \in \{1, 3, 5, 6\}$, *and we put* $g_I : \{1, 3, 5, 6\} \times \{1, 3, 5, 6\} \longrightarrow \{1, 3, 5, 6\}$ *such that* $g_I(x, y) = 1$ *for all* $x \in \{1, 3, 5, 6\}$. *The constant* c *becomes* $c_I = 3$, *and* $d_I = 6$. *We write the structure* $\mathbf{M}$ *as*

$$\mathbf{M} = [\{1, 3, 5, 6\} \quad \leq,\ f_I,\ g_I,\ c_I = 3,\ d_I = 6]$$

Exercise 3

Given a language

$$\mathcal{L} = \mathcal{L}(\{R\}, \{g\}, \emptyset,)$$

where $\#R = 2$, $\#g = 2$. *Define two structures for* $\mathcal{L}$, *both with infinite universe: one infinitely countable and one uncountable.*

Solution

There are many such structures. Here are two of the very simple.

$\mathbf{M_1} = [N,\ , \leq,\ +]$, where N is the set of natural numbers, and for example $\mathbf{M_2} = [R,\ , \leq,\ +]$, where R is the set of real numbers.

assign **Definition 11 (Assignment)**

Given a first order language $\mathcal{L} = \mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ with the set VAR of variables. Let $\mathbf{M} = [U, I]$ be a structure for $\mathcal{L}$ with the universe $U \neq \emptyset$.

An **assignment** *of $\mathcal{L}$ in $\mathbf{M} = [U, I]$ is any function*

$$s: \ VAR \ \longrightarrow \ U \tag{9}$$ assing

The assignment s is also called an **interpretation** *of variables VAR of $\mathcal{L}$ in a structure $\mathbf{M} = [U, I]$.*

Let $\mathbf{M} = [U, I]$ be a structure for $\mathcal{L}$ and $s : VAR \ \longrightarrow \ U$ be an **assignment** of variables VAR of $\mathcal{L}$ in a structure $\mathbf{M}$.

Let $\mathbf{T}$ be the set of all terms of $\mathcal{L}$. By definition 2, $VAR \subset \mathbf{T}$. We use the interpretation I to **extend** the assignment s to the set the set $\mathbf{T}$ of all terms of $\mathcal{L}$. Because of that we denote this extension by s_I rather then by s^* as we did before. The extension s_I of s is hence a mapping from $\mathbf{T}$ to U. It associates with each $t \in \mathbf{T}$ an element $I(t) \in U$. We denote this element $s_I(t)$ by t_I. We define the extension $s_I(t) = t_I$ of s by the induction of the length of the term $t \in \mathbf{T}$ and call it an *interpretation of terms* of $\mathcal{L}$ in a structure $\mathbf{M} = [U, I]$.

i-terms **Definition 12 (Interpretation of Terms)**

Given a language $\mathcal{L} = \mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ and a structure $\mathbf{M} = [U, I]$ for $\mathcal{L}$. Let an assignment

$$s : VAR \ \longrightarrow \ U$$

be any interpretation of variables VAR of $\mathcal{L}$ (assignment) in $\mathbf{M}$.

We **extend** *s to a function*

$$s_I : \mathbf{T} \ \longrightarrow \ U \tag{10}$$ sI

called an **interpretation s_I of terms** *of $\mathcal{L}$ in $\mathbf{M}$.*

The function s_I is defined by induction on the complexity of terms as follows.

1. *For any $x \in VAR$,*

$$s_I(x) = s(x);$$

2. *for any $c \in \mathbf{C}$,*

$$s_I(c) = c_I;$$

3. *for any $t_1, \ t_2, \ \ldots, t_n \in \mathbf{T}, \ n \geq 1, \ f \in \mathbf{F}$, such that $\#f = n$, and for any term $t = f(t_1, \ t_2, \ \ldots, t_n)$, we put*

$$s_I(t) = f_I(s_I(t_1), \ s_I(t_2), \ \ldots, \ s_I(t_n)),$$

i.e. we define

$$s_I(f(t_1, \ t_2, \ \ldots, t_n)) = f_I(s_I(t_1), \ s_I(t_2), \ \ldots, \ s_I(t_n)),$$

for any $t_1, \ t_2, \ \ldots, t_n \in \mathbf{T}, \ f \in \mathbf{T}$, such that $\#f = n$.

Exercise 4

Consider a language

$$\mathcal{L} = \mathcal{L}(\{P, R\},\ \{f,\ g,\ h,\},\ \{c, d\})$$

for # P = # R = 2, #f = # g=1, # h= 2.

Let $\mathbf{M} = [\{0, 1\}, I]$*, where the interpretation I is defined as follows.*

$P_I = \{(0, 0)\}$, $R_I = \{(0, 0), (1, 1)\}$,

$f_I(0) = 0$, $f_I(1) = 0$, $g_I(0) = 1$, $g_I(1) = 1$, h_I *is given by a formula*

$h_I(x, y) = x$ *for all* $(x, y) \in \{0, 1\} \times \{0, 1\}$*, and* $c_I = 1$, $d_I = 0$.

The assignment $s : VAR \longrightarrow \{0, 1\}$ *is such that* $s(x) = s(y) = s(z) = 1 = 1$*, for* $x, y, z \in VAR$ *(and any values for all other variables).*

Given a set $\mathbf{T}_0$ *of terms, evaluate* $s_I(t)$ *for all* $t \in \mathbf{T}_0$.

$\mathbf{T}_0 = \{z, y, x, c,\ f(c), f(x), g(z), f(g(d)), g(f(g(z))), h(c, f(g(d))), h(f(x), g(z))\}$.

Solution

First we evaluate terms that are variables and constants of $\mathcal{L}$ using the formulas 1. and 2. of definition 12: $s_I(x) = s(x), s_I(c) = c_I$, respectively and obtain: $s_I(z) = s(z) = 1$, $s_I(y) = s(y) = 1$, $s_I(x) = s(x) = 1$, $s_I(c) = c_I = 1$, $s_I(d) = d_I = 2$. We use the formula $s_I(f(t_1,\ t_2,\ \ldots, t_n)) = f_I(s_I(t_1),\ s_I(t_2),\ \ldots,\ s_I(t_n)$ to evaluate the rest of terms in $\mathbf{T}_0$ and obtain:

$s_I f(c) = f_I(s_I(c)) = f_I(c_I)) = f_I(1) = 0$, $s_I f(x) = f_I(s_I(x)) = f_I(1) = 0$

$s_I g(z) = g_I(s_I(c)) = g_I(1) = 1$, $s_I(f(g(d)) = f_I(s_I(f(g(d))) = f_I(g_I(s_I(d))) = f_I(g_I(c_I)) = f_I(g_I(1)) = f_I(1) = 0$,

$s_I g(f(g(z)) = g_I(f_I(g_I(s_I(z)))) = g_I(f_I(g_I(1))) = g_I(f_I(1)) = g_I(0) = 1$,

$s_I(h(c, f(g(d)))) = h_I(s_I(c), s_I(f(g(d)))) = h_I(c_I, f_I(g_I(s_I(d))))$

$= h_I(1, f_I(g_I(0))) = h_I(1, f_I(1)) = h_I(1, 0) = 1$,

$s_I(h(f(x), g(z))) = h_I(f_I(s_I(x)), g_I(s_I(x))) = h_I(f_I(1), g_I(1)) = h_I(0, 1) = 0$.

Observe that the interpretation of predicate symbols is irrelevant when evaluating an interpretation of terms, as the terms do not involve the predicate symbols.

Example 14

Consider a language

$$\mathcal{L} = \mathcal{L}(\{P,\ R\},\ \{f, h\},\ \emptyset\ )$$

for # P = # R = 2, #f = 1, # h= 2.

Let $\mathbf{M} = [Z, I]$*, where* Z *is the set on integers and the interpretation* I *for elements of* $\mathbf{F}$ *and* $\mathbf{C}$ *is as follows.*

$f_I : Z \longrightarrow Z$ *is given by formula* $f(m) = m + 1$ *for all* $m \in Z$*.*

$h_I : Z \times Z \longrightarrow Z$ *is given by formula* $f(m, n) = m + n$ *for all* $m, n \in Z$*.*

Let $s : VAR \longrightarrow Z$ *be any assignment such that* $s(x) = -5$, $s(y) = 2$ *and* $t_1, t_2 \in \mathbf{T}$ *be* $t_1 = h(y, f(f(x)))$ *and* $t_2 = h(f(x), h(x, f(y))$*.*

We evaluate:

$$s_I(t_1) = s_I(h(y, f(x)) = h_I(s_I(y), f_I(s_I(x))) = +(2, f_I(-5)) = 2 - 4 = -2,$$

$$s_I(t_2) = s_I(h(f(x), h(x, f(y))) = +(f_I(-5), +(-5, 3)) = -4 + (-5 + 3) = -6.$$

For any $t \in \mathbf{T}$, let $x_1, x_2, \ldots, x_n \in VAR$ be all variables appearing in t, we write it, in a similar way as we did in (6) for variables in formulas, as

$$t(x_1, x_2, \ldots, x_n).$$

Observation 1

For any term $t(x_1, x_2, \ldots, x_n) \in \mathbf{T}$*, any structure* $\mathbf{M} = [U, I]$ *and any assignments* s, s' *of* $\mathcal{L}$ *in* $\mathbf{M}$*, the following holds.*

If $s(x) = s'(x)$ *for all* $x \in \{x_1, x_2, \ldots, x_n\}$*, i.e the assignments* s, s' *agree on all variables appearing in* t*, then,* $s_I(t) = {s'}_I(t)$*.*

Thus for any $t \in \mathbf{T}$, the function $s_I : \mathbf{T} \longrightarrow U$ defined by (10) depends on only a finite number of values of $s(x)$ for $x \in VAR$.

Given a structure $\mathbf{M} = [U, I]$ and an assignment $s : VAR \longrightarrow U$. We write

$$s(^{a}_{x}) \tag{11}$$ forQ

to denote any assignment $s' : VAR \longrightarrow U$ such that s, s' **agree** on all variables except on x, such that $s'(x) = a$, for certain $a \in U$.

Given a first order (predicate) language $\mathcal{L} = \mathcal{L}_{CON}(\mathbf{P}, \mathbf{F}, \mathbf{C})$. The satisfaction relation $(\mathbf{M}, s) \models A$ between structures, assignments and formulas of $\mathcal{L}$ is defined by induction on the complexity of formulas of $\mathcal{L}$. It is the satisfaction relation $(\mathbf{M}, s) \models A$ that allows us to **distinguish** one semantics for a given $\mathcal{L}$ from the other, and consequently one logic from the other. We define now only a classical satisfaction and the notion of classical predicate tautology.

def:sat **Definition 13 (Classical Satisfaction)**

Given a classical predicate (first order) language

$$\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P}, \mathbf{F}, \mathbf{C}). \tag{12}$$ classL

Let $\mathbf{M} = [U, I]$ *be a structure for* $\mathcal{L}$*,* s *be any assignment of* $\mathcal{L}$ *in* $\mathbf{M}$*, i.e.* $s : VAR \longrightarrow U$*. Let* $A \in \mathcal{F}$ *be any formula of* $\mathcal{L}$*. We define a relation*

$$(\mathbf{M}, s) \models A$$

that reads: the assignment s **satisfies** *the formula* A *in* $\mathbf{M}$*, by induction on the complexity of* A *as follows.*

(i) *A is atomic formula (5), i.e. A is $P(t_1, t_2, \ldots, t_n)$ for $P \in \mathbf{P}$, $\#P = n$.*

$(\mathbf{M}, s) \models P(t_1, t_2, \ldots, t_n)$ *if and only if* $(s_I(t_1), s_I(t_2), \ldots, s_I(t_n)) \in P_I$.

(ii) *A is not atomic formula and has one of connectives of $\mathcal{L}$ as the main connective.*

$(\mathbf{M}, s) \models \neg A$ *if and only if* $(\mathbf{M}, s) \not\models A$,

$(\mathbf{M}, s) \models (A \cap B)$ *if and only if* $(\mathbf{M}, s) \models A$ *and* $(\mathbf{M}, s) \models B$,

$(\mathbf{M}, s) \models (A \cup B)$ *if and only if* $(\mathbf{M}, s) \models A$ *or* $(\mathbf{M}, s) \models B$ *or both,*

$(\mathbf{M}, s) \models (A \Rightarrow B)$ *if and only if ether* $(\mathbf{M}, s) \not\models A$ *or else* $(\mathbf{M}, s) \models B$ *or both.*

(iii) *A is not atomic formula and begins with one of the quantifiers.*

$(\mathbf{M}, s) \models \exists x A$ *if and only if there is* s' *such that* s, s' **agree** *on all variables except on* x*, and* $(\mathbf{M}, s') \models A$,

$(\mathbf{M}, s) \models \forall x A$ *if and only if for all* s' *such that* s, s' **agree** *on all variables except on* x*, and* $(\mathbf{M}, s') \models A$.

Observe that that the truth or falsity of $(\mathbf{M}, s) \models A$ depends only on the values of $s(x)$ for variables x which are actually free in the formula A. This is why we often write the condition **(iii)** as

(iii)' $A(x)$ is not atomic formula (with a free variable x) and begins with one of the quantifiers.

$(\mathbf{M}, s) \models \exists x A(x)$ if and only if there is s' such that $s(y) = s'(y)$ for all $y \in VAR - \{x\}$, and $(\mathbf{M}, s') \models A(x)$,

$(\mathbf{M}, s) \models \forall x A$ if and only if for all s' such that $s(y) = s'(y)$ for all $y \in VAR - \{x\}$, and $(\mathbf{M}, s') \models A(x)$.

sat **Exercise 5**

For the structures $\mathbf{M}_i$*, find assignments* $s_i,\ s'_i$ *(*$1 \leq i \leq 4$*), such that*

$$(\mathbf{M}_i, s_i) \models Q(x,c), \quad (\mathbf{M}_i, s'_i) \not\models Q(x,c) \quad \textit{for}\ \ Q \in \mathbf{P},\ c \in \mathbf{C}.$$

The structures $\mathbf{M}_i$ *are defined as follows (the interpretation I for each of them is specified only for symbols in the formula* $Q(x,c)$*, and* N *denotes the set of natural numbers.*

$$\mathbf{M}_1 = [\{1\},\ Q_I :=,\ c_I : 1], \qquad \mathbf{M}_2 = [\{1,2\},\ Q_I :\leq,\ c_I : 1],$$

$$\mathbf{M}_3 = [N,\ Q_I :\geq,\ c_I : 0], \quad \textit{and} \quad \mathbf{M}_3 = [N,\ Q_I :\geq,\ c_I : 1.]$$

Solution
Consider $\mathbf{M}_1 = [\{1\},\ Q_I :=,\ c_I : 1]$. Observe that all $s : VAR \longrightarrow \{1\}$ must be defined by a formula $s(x) = 1$ for all $x \in VAR$. We evaluate (definition 12), $s_I(x) = 1, s_I(c) = c_I = 1$. By definition 13, $(\mathbf{M}_1, s) \models Q(x,c)$ if and only if $(s_I(x),\ s_I(c)) \in Q_I$, i.e. $(1,1) \in =$ what is true as $1 = 1$. We have proved

$$(\mathbf{M}_1, s) \models Q(x,c) \text{ for all assignments } s : VAR \longrightarrow \{1\}.$$

Consider $\mathbf{M}_2 = [\{1,2\},\ Q_I :\leq,\ c_I : 1]$. Let $s : VAR \longrightarrow \{1,2\}$ be any assignment, such that $s(x) = 1$. We evaluate $s_I(x) = 1,\ s_I(c) = 1$ and verify whether $(s_I(x),\ s_I(c)) \in Q_I$, i.e. whether $(1,1) \in \leq$. This is true as $1 \leq 1$. We have found s (in fact uncountably many such s) such that

$$(\mathbf{M}_2, s) \models Q(x,c).$$

Let now s' be any assignment $s' : VAR \longrightarrow \{1,2\}$, such that $s'(x) = 2$. We evaluate $s'_I(x) = 1,\ s'_I(c) = 1$ and verify whether $s'_I(x),\ s'_I(c)) \in Q_I$, i.e. whether $(2,1) \in \leq$. This is not true as $2 \not\leq 1$. We have found $s' \neq s$ (in fact uncountably many such s') such that

$$(\mathbf{M}_2, s') \not\models Q(x,c).$$

Consider $\mathbf{M}_3 = [N,\ Q_I :\geq,\ c_I : 0]$. Let $s : VAR \longrightarrow N$ be any assignment, such that $s(x) = 5$. We evaluate $s_I(x) = 5,\ s_I(c) = 0$. Observe that the condition $(s_I(x),\ s_I(c)) \in Q_I$ holds as $5 \geq 0$ and

$$(\mathbf{M}_3, s) \models Q(x,c).$$

Let now s' be any assignment $s' : VAR \longrightarrow N$. By definition, $s'(x) = n$, for any $n \in N$, and $s'_I(x),\ 0) \in Q_I$ holds for any s' as $n \geq 0$ for all $n \in N$. This proves that there is no s', such that $(\mathbf{M}_3, s') \not\models Q(x,c)$.

Consider $\mathbf{M}_4 = [N,\ Q_I :\geq,\ c_I : 1]$. Let $s : VAR \longrightarrow N$ be any assignment, such that $s(x) = 5$. We evaluate $s_I(x) = 5,\ s_I(c) = 1$. Observe that the condition $(s_I(x),\ s_I(c)) \in Q_I$ holds as $5 \geq 1$ and hence

$$(\mathbf{M}_4, s) \models Q(x, c).$$

Let now s' be any assignment $s' : VAR \longrightarrow N$, such that $s'(x) = 0$. The the condition $(s'_I(x),\ s'_I(c)) \in Q_I$ does not holds as $0 \not\geq 1$ and

$$(\mathbf{M}_4, s') \not\models Q(x, c).$$

Directly from the definition13 we have that the following holds.

sat1 **Example 15**

Let $\mathbf{M}_i$ $(1 \leq i \leq 4)$ be structures in defined the exercise 5 and let corresponding assignments s_i be as defined as its solutions.

1. $(\mathbf{M}_1, s) \models Q(x, c)$, $(\mathbf{M}_1, s) \models \forall x Q(x, c)$, $(\mathbf{M}_1, s) \models \exists x Q(x, c)$.

2. $(\mathbf{M}_2, s) \models Q(x, c)$, $(\mathbf{M}_2, s) \not\models \forall x Q(x, c)$, $(\mathbf{M}_1, s) \models \exists x Q(x, c)$.

3. $(\mathbf{M}_3, s) \models Q(x, c)$, $(\mathbf{M}_3, s) \models \forall x Q(x, c)$, $(\mathbf{M}_3, s) \models \exists x Q(x, c)$.

4. $(\mathbf{M}_4, s) \models Q(x, c)$, $(\mathbf{M}_4, s) \not\models \forall x Q(x, c)$, $(\mathbf{M}_4, s) \models \exists x Q(x, c)$.

model **Definition 14 (Model)**

Given a language $\mathcal{L}$, a formula A of $\mathcal{L}$, and a structure $\mathbf{M} = [U, I]$ for $\mathcal{L}$.

The structure $\mathbf{M}$ is a **model** *for the formula A if and only if $(\mathbf{M}, s) \models A$ for all $s : VAR \longrightarrow U$. We denote it as $\mathbf{M} \models A$.*

For any set $\Gamma \subseteq \mathcal{F}$ of formulas of $\mathcal{L}$, $\mathbf{M}$ is a **model** *for Γ if and only if $\mathbf{M} \models A$ for all $A \in \Gamma$. We denote it as $\mathbf{M} \models \Gamma$.*

We define now a very important semantic notion. It has different names: *logical consequence, logical implication, semantic consequence, logical (semantical) entailment.* We use a name *logical consequence* and define it as follows.

def:lcons **Definition 15 (Logical Consequence)**

For any $A, B \in \mathcal{F}$ and any set $\Gamma \subseteq \mathcal{F}$ of formulas of $\mathcal{L}$, we say that a formula B is a **logical consequence** *of a set Γ and write it as $\Gamma \models B$, if and only if all models of the set Γ are models of the formula B.*

When $\Gamma \models B$ we also say that Γ **logically implies** *B. When $\Gamma = \{A\}$ we write it as $A \models B$ and say A* **logically implies** *B.*

We say that A and B are **logically equivalent** *if and only if $A \models B$ and $A \models B$.*

Directly from the model definition 14 we get the following.

c-model **Definition 16 (Counter Model)**

Given a language $\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P},\mathbf{F},\mathbf{C})$*, a formula* A *of* $\mathcal{L}$*, and a structure* $\mathbf{M} = [U, I]$ *for* $\mathcal{L}$*.*

The structure $\mathbf{M} = [U, I]$ *is a* **counter model** *for the formula* A *if and only if there is an assignment* $s : VAR \longrightarrow U$*, such that* $(\mathbf{M}, s) \not\models A$*.*

We denote it as $\mathbf{M} \not\models A$*.*

For any set $\Gamma \subseteq \mathcal{F}$ *of formulas of* $\mathcal{L}$*,* $\mathbf{M}$ *is a* **counter model** *for* Γ *if and only if there is* $A \in \Gamma$*, such that* $\mathbf{M} \not\models A$*.*

We denote it as $\mathbf{M} \not\models \Gamma$*.*

Observe that if A is a **sentence** (definition 4) then the truth or falsity of $(\mathbf{M}, s) \models A$ is completely independent of s. Hence if $(\mathbf{M}, s) \models A$ for some s, it holds for all s and the following holds.

sentence **Fact 1**

For any formula A *of* $\mathcal{L}$*,*

If A *is a sentence, then if there* s *such that* $(\mathbf{M}, s) \models A$*, then* $\mathbf{M}$ *is a model for* A*, i.e.* $\mathbf{M} \models A$

We transform any formula A of $\mathcal{L}$ into a certain sentence by binding all its free variables. The resulting sentence is called a closure of A and is defined as follows.

cA **Definition 17 (Closure)**

Given a formula A *of* $\mathcal{L}$*.*

By the **closure** *of* A *we mean the formula obtained from* A *by prefixing in universal quantifiers all variables the are free in* A*. If* A *does not have free variables (i.e. is a sentence), the closure if* A *is defined to be* A *itself.*

Obviously, a closure of any formula is always a sentence. For example, if A, B are formulas

$$(P(x_1, x_2) \Rightarrow \neg\exists x_2\ Q(x_1, x_2, x_3)), \quad (\forall x_1 P(x_1, x_2) \Rightarrow \neg\exists x_2\ Q(x_1, x_2, x_3)),$$

their respective **closures** are

$$\forall x_1 \forall x_2 \forall x_3\ ((P(x_1, x_2) \Rightarrow \neg\exists x_2\ Q(x_1, x_2, x_3))),$$

$$\forall x_1 \forall x_2 \forall x_3\ ((\forall x_1 P(x_1, x_2) \Rightarrow \neg\exists x_2\ Q(x_1, x_2, x_3))).$$

models **Example 16**

Let $Q \in \mathbf{P},\ \#Q = 2$ *and* $c \in \mathbf{C}$ *Consider formulas*

$$Q(x,c), \quad \exists x Q(x,c), \quad \forall x Q(x,c)$$

and the structures from exercise 5 defined as follows.

$$\mathbf{M}_1 = [\{1\},\ Q_I :=,\ c_I : 1], \qquad \mathbf{M}_2 = [\{1,2\},\ Q_I :\leq,\ c_I : 1],$$

$$\mathbf{M}_3 = [N,\ Q_I :\geq,\ c_I : 0], \quad \textit{and} \quad \mathbf{M}_4 = [N,\ Q_I :\geq,\ c_I : 1.]$$

Directly from example 15 and Fact 1, we get that:

1. $\mathbf{M}_1 \models Q(x,c), \quad \mathbf{M}_1 \models \forall x Q(x,c), \quad \mathbf{M}_1 \models \exists x Q(x,c).$

2. $\mathbf{M}_2 \not\models Q(x,c), \quad \mathbf{M}_2 \not\models \forall x Q(x,c), \quad \mathbf{M}_2 \models \exists x Q(x,c).$

3. $\mathbf{M}_3 \models Q(x,c), \quad \mathbf{M}_3 \models \forall x Q(x,c), \quad \mathbf{M}_3 \models \exists x Q(x,c).$

4. $\mathbf{M}_4 \not\models Q(x,c), \quad \mathbf{M}_4 \not\models \forall x Q(x,c), \quad \mathbf{M}_4 \models \exists x Q(x,c).$

trueM **Definition 18 (True, False in M)**

Given a structure $\mathbf{M} = [U, I]$ *for* $\mathcal{L}$ *and a formula* A *of* $\mathcal{L}$*. We say that:*

A is **true** *in* $\mathbf{M}$ *(written as* $\mathbf{M} \models A$*) if and only if all assignments* s *of* $\mathcal{L}$ *in* $\mathbf{M}$ *satisfy* A*, i.e. when* $\mathbf{M}$ *is a* **model** *for* A*.*

A is **false** *in* $\mathbf{M}$ *(written as* $\mathbf{M} =| A$*) if and only if no assignment* s *of* $\mathcal{L}$ *in* $\mathbf{M}$ *satisfies* A*.*

By the definition 14 we have that A is **true** in $\mathbf{M}$ only when the structure $\mathbf{M}$ is a **model** for A. This is why we use the notation $\mathbf{M} \models A$ in both cases.

Obviously, if A is not true in $\mathbf{M}$, then it is false, and vice versa. This proves correctness of our definition with respect to the intuitive understanding.

We get directly from definition 18 and the example 16 the following.

tf **Example 17**

Let $\mathbf{M}_1 - \mathbf{M}_4$ *be structures defined in example 5.*

1. Formulas $Q(x,c),\ \forall x Q(x,c),\ \exists x Q(x,c)$ *are all* **true** *in the structures* $\mathbf{M}_1$ *and* $\mathbf{M}_3$*.*

2. Formula $\exists x Q(x,c)$ *is also* **true** *in* $\mathbf{M}_2$ *and in* $\mathbf{M}_3$*.*

3. Formulas $\neg Q(x,c), \neg\forall x Q(x,c), \neg\exists x Q(x,c)$ *are all* **false** *in the structures* $\mathbf{M}_1$ *and* $\mathbf{M}_3$*.*

4. Formula $\neg\exists x Q(x,c)$ *is also* **false** *in* $\mathbf{M}_2$ *and in* $\mathbf{M}_3$*.*

5. Formulas $(Q(x,c) \cap \neg Q(x,c))$, $(\neg \forall x Q(x,c) \cap \forall x Q(x,c))$, *and the formula* $(\exists x Q(x,c) \cap \neg \exists x Q(x,c))$ *are all* **false** *in all structures* $\mathbf{M}_1 - \mathbf{M}_4$.

6. The formula $\forall x Q(x,c)$ *is false in a structure* $\mathbf{M}_5 = [N,\ Q_I :<,\ c_I : 0]$.

Here are some properties of the notions "A is true in $\mathbf{M}$, written symbolically as $\mathbf{M} \models A$, and "A is false in $\mathbf{M}$, written symbolically as $\mathbf{M} =| A$. They are obvious under intuitive understanding of the notion of satisfaction. Their formal proofs are left as exercise for the reader.

TFprop **Property 1 (Truth, Falsity, Satisfaction)**

Given a structure $\mathbf{M} = [U, I]$ *for* $\mathcal{L}$ *and any formulas formula* A, B *of* $\mathcal{L}$. *The following properties hold.*

P1. A *is false in* $\mathbf{M}$ *if and only if* $\neg A$ *is true in* $\mathbf{M}$, *i.e.*

$$\mathbf{M} =| A \quad \text{if and only if} \quad \mathbf{M} \models \neg A.$$

P2. A *is true in* $\mathbf{M}$ *if and only if* $\neg A$ *is false in* $\mathbf{M}$, *i.e.*

$$\mathbf{M} \models A \quad \text{if and only if} \quad \mathbf{M} =| \neg A.$$

P3. It is not the case that both $\mathbf{M} \models A$ *and* $\mathbf{M} \models \neg A$, *i.e. no formula of* $\mathcal{L}$ *can be both true and false in* $\mathbf{M}$, *i.e. there is no formula* A, *such that* $\mathbf{M} \models A$ *and* $\mathbf{M} =| A$.

P4. If $\mathbf{M} \models A$ *and* $\mathbf{M} \models (A \Rightarrow B)$, *then* $\mathbf{M} \models B$.

P5. $(A \Rightarrow B)$ *is false in* $\mathbf{M}$ *if and only if* $\mathbf{M} \models A$ *and* $\mathbf{M} \models \neg B$, *i.e.*

$$\mathbf{M} =| (A \Rightarrow B) \quad \text{if and only if} \quad \mathbf{M} \models A \text{ and } \mathbf{M} \models \neg B.$$

P6. $\mathbf{M} \models A$ *if and only if* $\mathbf{M} \models \forall x A$.

P7. A formula A *its true in* $\mathbf{M}$ *if and only if its closure (definition 17) is true in* $\mathbf{M}$.

def:ptaut **Definition 19 (Valid, Tautology)**

Given a language $\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P}, \mathbf{F}, \mathbf{C})$, *a formula* A *of* $\mathcal{L}$. .

A formula A *is* **predicate tautology** *(is* **valid***) if and only if* $\mathbf{M} \models A$ *for all structures* $\mathbf{M} = [U, I]$, *i.e. when* A *is* **true** *in all structures* $\mathbf{M}$ *for* $\mathcal{L}$.

We write

$$\models A \quad \text{or} \quad \models_p A,$$

to denote that a formula A *is* **predicate tautology** *(is* **valid***).*

We write

$$\models_p A$$

when there is a need to stress a **distinction** between propositional and predicate tautologies, otherwise we will use the symbol $\models$.

Predicate tautologies are also called **laws of quantifiers**.

Following the notation **T** for the set of all propositional tautologies (chapter **??**) we denote by $\mathbf{T}_p$ the set of all predicate tautologies, i.e.

$$\mathbf{T}_p = \{A \text{ of } \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P},\mathbf{F},\mathbf{C}) : \models_p A\}. \quad (13) \quad \boxed{\texttt{Tp}}$$

Directly from the definition 18, the tautology definition 19 we get the following basic properties of logical consequence as defined by definition 15.

logcons **Property 2**

For any $A, B \in \mathcal{F}$ and any set $\Gamma \subseteq \mathcal{F}$ of formulas of $\mathcal{L}$,

P1. $A \models B$ *if and only if* $\models (A \Rightarrow B)$.

P2. If $A \models B$ *and* A *is true in* $\mathbf{M}$, *then* B *is true in* $\mathbf{M}$.

P2. If $\Gamma \models B$ *and if all formulas in* Γ *are true in* $\mathbf{M}$, *then* B *is true in* $\mathbf{M}$.

We get immediately from the above definition 19 of a following definition of a notion *" A is not a predicate tautology"*.

def:notaut **Definition 20**

For any formula A of predicate language $\mathcal{L}$,
A **is not** *a predicate tautology* ($\not\models A$) *if and only if there is a structure* $\mathbf{M} = (U, I)$ *for* $\mathcal{L}$, *such that* $\mathbf{M} \not\models A$.
We call such structure $\mathbf{M}$ *a* **counter-model** *for* A.

The definition 20 says: to prove that a formula A is not a predicate tautology one has to show a *counter- model* $\mathbf{M} = (U, I)$. It means one has to show a non-empty set U, define an interpretation I, and an assignment $s : VAR \longrightarrow U$ such that $(\mathbf{M}, s) \not\models A$.

We introduce, similarly as in a case of propositional semantic a notion of predicate contradiction.

def:contr **Definition 21 (Contradiction)**

For any formula A of predicate a language $\mathcal{L}$,

A *is a* **predicate contradiction** *if and only if* A *is* **false** *in all structures* $\mathbf{M}$.

We denote it as $=|\ A$ *and write symbolically*

$$=|\ A \quad \textit{if and only if} \quad \mathbf{M}=|\ A, \ \textit{for all structures } \mathbf{M}.$$

When there is a need to distinguish between propositional and predicate contradictions we also use symbol

$$=|_p\ A,$$

where "p" stands for "predicate".

Following the notation $\mathbf{C}$ for the set of all propositional **contradictions** (chapter **??**) we denote by $\mathbf{C}_p$ the set of all predicate contradictions, i.e.

$$\mathbf{C}_p = \{A \ \text{ of } \ \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P},\mathbf{F},\mathbf{C}) : \ =|_p\ A\}. \tag{14}$$

Cp

Directly from the definition 18 and Property 1 we have the following duality property, the same as the one for propositional logic.

Fact 2

For any formula A of predicate a language $\mathcal{L}$*,*

$$A \in \mathbf{T}_p \quad \textit{if and only if} \quad \neg A \in \mathbf{C}_p,$$

$$A \in \mathbf{C}_p \quad \textit{if and only if} \quad \neg A \in \mathbf{T}_p.$$

Obviously, the formulas $(Q(x,c) \cap \neg Q(x,c))$, $(\neg\forall x Q(x,c) \cap \forall x Q(x,c))$, and the formula $(\exists x Q(x,c) \cap \neg\exists x Q(x,c))$ defined in example 17 are not only false in the structures $\mathbf{M}_1 - \mathbf{M}_4$, but are false in all structures $\mathbf{M}$ for $\mathcal{L}$. By definition 21 they all are predicate contradictions. Observe that they all are substitutions of propositional contradictions $(a \cap \neg a)$ or $(\neg a \cap a)$. By the same argument the formulas $(Q(x,c) \cup \neg Q(x,c))$, $(\neg\forall x Q(x,c) \cup \forall x Q(x,c))$, $(\exists x Q(x,c) \cap \neg\exists x Q(x,c))$ are predicate tautologies as they are substitutions of propositional tautologies $(a \cup \neg a)$ or $(\neg a \cup a)$.

We put these examples and observations in a following theorems that establish relationship between propositional and predicate tautologies and contradictions. We write now $\models$, $=|$ do denote respectively propositional tautologies and contradiction, and $\models_p$, $=|_p$ for predicate tautologies and contradictions. We first formalize and prove (theorem 1) the intuitively obvious fact: *if a formula A is a propositional tautology (contradiction), then replacing propositional variables in A by any formulas of a predicate language we obtain a formula which is a predicate tautology (contradiction).*

Example 18

Let consider the following example of a propositional tautology and a propositional contradiction.

$$\models ((a \Rightarrow b) \Rightarrow (\neg a \cup b)) \quad \textit{and} \quad =|\ ((a \cup \neg a) \Rightarrow (\neg b \cap b)).$$

Substituting $\exists x P(x,z)$ *for* a, *and* $\forall y R(y,z)$ *for* b, *we obtain, by theorem 1, that*

$$\models_p \ ((\exists x P(x,z) \Rightarrow \forall y R(y,z)) \Rightarrow (\neg\exists x P(x,z) \cup \forall y R(y,z))) \quad \textit{and}$$

$$=| \ ((\exists x P(x,z) \cup \neg\exists x P(x,z)) \Rightarrow (\neg\forall y R(y,z) \cap \forall y R(y,z))).$$

We put it all in a more formal and more general and precise language as follows.

Given a propositional language $\mathcal{L}_0 = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}$ with the set $\mathcal{F}_0$ of formulas and a predicate language$\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P},\mathbf{F},\mathbf{C})$ with the set $\mathcal{F}$ of formulas. Let $A(a_1,a_2,\ldots,a_n) \in \mathcal{F}_0$ and $A_1,\ A_2,\ \ldots,\ A_n \in \mathcal{F}$. We denote by

$$A(a_1/A_1,\ a_2/A_2,\ \ldots,\ a_n/A_n) \tag{15}$$ sfor

the result of replacing in A the free variables $a_1,a_2,\ldots,\ a_n$ by the formulas $A_1,\ A_2,\ \ldots,\ A_n \in \mathcal{F}$. Of course $A(a_1/A_1,\ a_2/A_2,\ \ldots,\ a_n/A_n) \in \mathcal{F}$.

thm:sub1 **Theorem 1**

Given a propositional language $\mathcal{L}_0$ *with the set* $\mathcal{F}_0$ *of formulas and a predicate language* $\mathcal{L}$ *with the set* $\mathcal{F}$ *of formulas.*

For any $A(a_1,a_2,\ldots a_n) \in \mathcal{F}_0$ *and any* $A_1,A_2,\ldots A_n \in \mathcal{F}$ *the following holds.*

1. *If* $\models\ A(a_1,a_2,\ldots,a_n)$, *then* $\models_p\ A(a_1/A_1,\ a_2/A_2,\ \ldots,\ a_n/A_n)$.

2. *If* $=|\ A(a_1,a_2,\ldots,a_n)$, *then* $=|_p\ A(a_1/A_1,\ a_2/A_2,\ \ldots,\ a_n/A_n)$.

Proof 1. follows directly from satisfaction definition 13. 2. follows from definition 13, property 1, and definition 21.

Some predicate tautologies are, by theorem 1, substitutions of propositional formulas. Visibly a predicate formula $(\forall x\ A(x) \Rightarrow \exists x\ A(x))$ can not be obtained as a substitution of a propositional formula. We prove now that it is a predicate tautology.

p1 **Fact 3**

For any formula $A(x)$ *of* $\mathcal{L}$,

$$\models\ (\forall x\ A(x) \Rightarrow \exists x\ A(x)).$$

Proof
Assume that $\not\models (\forall x\ A(x) \Rightarrow \exists x\ A(x))$. By definition 20 there is a structure $\mathbf{M} = (U,I)$ and $s : VAR \longrightarrow U$, such that $(\mathbf{M},s) \not\models (\forall x\ A(x) \Rightarrow \exists x\ A(x))$. By definition 13, $(\mathbf{M},s) \models \forall x\ A(x)$ and $(\mathbf{M},s) \not\models \exists x\ A(x)$. It means that $(\mathbf{M},s') \models A(x)$ for all s' such that s,s' **agree** on all variables except on x, and it is not true that there is s' such that s,s' **agree** on all variables except

on x, and $(\mathbf{M}, s') \models A(x)$. This is impossible and this contradiction proves $\models (\forall x\ A(x) \Rightarrow \exists x\ A(x))$.

Given a set $\mathcal{F}$ of formulas of a predicate language $\mathcal{L}$. We denote by $O\mathcal{F}$ set of all open formulas of $\mathcal{L}$, i.e. formulas without quantifiers. We prove that any open formula in order to be predicate tautologies must be a substitution definied in theorem 1 of a propositional tautology. I.e. we have the following substitution theorem.

thm:sub2 **Theorem 2**

Any open formula A of a predicate language $\mathcal{L}$ is a predicate tautology if and only if it is a substitution of a propositional tautology as defined in theorem 1.

Proof
Observe that every open formula from $A \in O\mathcal{F}$ is a form
$B(a_1/A_1,\ a_2/A_2,\ \ldots,\ a_n/A_n)$ for certain propositional formula $B(a_1, a_2, \ldots a_n)$, where $A_1,\ A_2,\ \ldots,\ A_n$ are predicate atomic formulas from the set $A\mathcal{F}$ as defined in (5). Theorem 2 follows directly from the following.

lemsub2 **Lemma 1**

Let σ be a one to one mapping from the set V_0 of propositional variables of propositional language $\mathcal{L}_0$ into the set $A\mathcal{F}$ of the atomic formulas of the predicate language $\mathcal{L}$. For any $A(a_1, a_2, \ldots a_n) \in \mathcal{F}_0$,

$$\models A(a_1, a_2, \ldots, a_n),\ \textit{if and only if}\ \models_p A(a_1/\sigma(a_1),\ \ldots,\ a_n/\sigma(a_n)).$$

Proof of lemma
The implication "if $\models A(a_1, a_2, \ldots, a_n)$, then
$\models_p A(a_1/\sigma(a_1),\ \ldots,\ a_n/\sigma(a_n))$" holds as a particular case of theorem 2. We prove now the converse implication by proving its opposite

$$\text{if } \not\models A(a_1, a_2, \ldots, a_n),\ \text{then } \not\models_p A(a_1/\sigma(a_1), \ldots,\ a_n/\sigma(a_n)). \qquad (16)$$ opp

Assume $\not\models A(a_1, a_2, \ldots, a_n)$. There exists a truth assignment $v : V_0 \longrightarrow \{T, F\}$ such that $v^*(A(a_1, a_2, \ldots, a_n)) = F$. We construct a counter model $\mathbf{M}$ for $A(a_1/\sigma(a_1), \ldots,\ a_n/\sigma(a_n))$ as follows. Let $\mathbf{M} = [\mathbf{T}, I]$, where $\mathbf{T}$ is the set of all terms of $\mathcal{L}$, and for any $c \in \mathbf{C},\ f \in \mathbf{F},\ P \in \mathbf{P}$ we put $c_I = c,\ f_I(t_1, t_2, \ldots, t_n) = f(t_1, t_2, \ldots t_n),\ P_I \subseteq \mathbf{T}^{\#P}$.

Let now the s assignment of $\mathcal{L}$ in $\mathbf{M}$ be an identity, i.e. $s : VAR \longrightarrow \mathbf{T}$ is such that $s(x) = x$ for all $x \in VAR$. We extend s to the interpretation of terms (definition 12) as follows.

$s_I(x) = s(x) = x,\ s_I(c) = c_I = c,\ s_I(f(t_1, t_2, \ldots t_n)) = f_I(s_I(t_1),\ \ldots,\ s_I(t_n)) = f(t_1, t_2, \ldots t_n)$, i.e. we have that $s_I(t) = t$ for all $t \in \mathbf{T}$.

We have that for every atomic formula $P(t_1, t_2, \ldots t_n)$ there is exactly one propositional variable a, such that $P(t_1, t_2, \ldots t_n) = \sigma(a)$. We define now that P_I as follows.

$(t_1, t_2, \ldots t_n) \in P_I$ if and only if $P(t_1, t_2, \ldots t_n) = \sigma(a)$ and $v(a) = T$.

$(t_1, t_2, \ldots t_n) \notin P_I$ if and only if $P(t_1, t_2, \ldots t_n) = \sigma(a)$ and $v(a) = F$.

We assumed that $v : V_0 \longrightarrow \{T, F\}$ is such that $v^*(A(a_1, a_2, \ldots, a_n)) = F$. Directly form definition of the assignment s and the interpretation I we have that $([\mathbf{T}, I], s) \not\models A(a_1/\sigma(a_1), \ldots, \ a_n/\sigma(a_n))$. It end the roof of lemma 1 and hence the proof of theorem 2.

Fact 4

The converse implication to (3) is not a predicate tautology, i.e. there is a formula A of $\mathcal{L}$, such that

$$\not\models \ (\exists x \ A(x) \Rightarrow \forall x \ A(x)). \qquad (17)$$ p2

Proof
Observe that to prove (17) we have to provide an example of an instance of a formula $A(x)$ and construct a counter-model $\mathbf{M} = (U, I)$ for it. Let $A(x)$ be an atomic formula $P(x, c)$, for any $P \in \mathbf{P}, \#P = 2$. The instance is

$$(\exists x \ P(x, c) \Rightarrow \forall x \ P(x, c)).$$

We take as $\mathbf{M} = (N, P_I :<, \ c_I : 3)$ for N set of natural numbers. Let s be any assignment $s : VAR \longrightarrow N$. We show now $(\mathbf{M}, s) \models \exists x \ P(x, c)$. Take any s' such that $s'(x) = 2$ and $s'(y) = s(y)$ for all $y \in VAR - \{x\}$. We have $(2, 3) \in P_I$, as $2 < 3$ and hence there exists s' that agrees with s on all variables except on x, and $(\mathbf{M}, s') \models P(x, c)$. But $(\mathbf{M}, s) \not\models \forall x \ P(x, c)$ as for example for s' such that $s'(x) = 5$ and $s'(y) = s(y)$ for all $y \in VAR - \{x\}$, $(2, 3) \notin P_I$, as $5 \not< 3$. This proves that $\mathbf{M} = (N, P_I :<, \ c_I : 3)$ is a counter model for $\forall x \ P(x, c)$. Hence $\not\models \ (\exists x \ A(x) \Rightarrow \forall x \ A(x))$.

The "shorthand" solution is: the formula $(\exists x \ P(x, c) \Rightarrow \forall x \ P(x, c))$ becomes in $\mathbf{M} = (N, P_I :<, \ c_I : 3)$ a mathematical statement (written with logical symbols): $\exists n \ n < 3 \ \Rightarrow \ \forall n \ n < 3$. It is an obviously *false* statement in the set N of natural numbers, as there is $n \in N$, such that $n < 3$, for example $n = 2$, and it is not true that all natural numbers are smaller then 3.

We have to be very careful when we deal with **quantifiers with restricted domain** (definition 9). We adopt the following definition for restricted domain quantifiers.

rtaut **Definition 22 (Restricted Quantifiers Tautology)**

For any formulas
$A(x), B(x) \in \mathcal{F}$ with any free variable $x \in VAR$, and for the restricted domain

quantifies $\forall_{B(x)},\ \exists_{B(x)}$ *we define*

$$\models\ \forall_{B(x)}\ A(x) \quad \textit{if and only if} \quad \models\ \forall x\ (B(x) \Rightarrow A(x)),$$

$$\models \exists_{B(x)}\ A(x) \quad \textit{if and only if} \quad \models\ \exists x\ (B(x) \cap A(x)).$$

The most basic predicate tautology (3) fails when we use the quantifiers with restricted domain. We show now that

$$\not\models\ (\forall_{B(x)}\ A(x) \Rightarrow \exists_{B(x)}\ A(x)). \tag{18}$$ rq1

By definition 22 to prove (18) means to prove that corresponding proper formula of $\mathcal{L}$ obtained by the restricted quantifiers *transformations rules* (48), (49) is not a predicate tautology, i.e. to show that

$$\not\models\ (\forall x(B(x) \Rightarrow A(x)) \Rightarrow \exists x(B(x) \cap A(x))). \tag{19}$$ rq2

In order to prove (19) we have to provide an example of particular formulas $A(x), B(x)$ and to construct a counter model for these particular formulas. We take as $B(x), A(x)$ atomic formulas $Q(x, c), P(x, c)$ and we construct a counter model a corresponding formula

$$(\forall x(Q(x, c) \Rightarrow P(x, c)) \Rightarrow \exists x(Q(x, c) \cap P(x, c))) \tag{20}$$ crq2

as follows. We take $\mathbf{M} = (N, I)$, where N is the set of real numbers and the interpretation I is defined as $Q_I :<,\ P_I :>,\ c_I : 0$. The "shorthand" solution is as follows. The formula 19) becomes a mathematical statement

$$(\forall n\ (n < 0 \Rightarrow n > 0) \Rightarrow \exists_{n \in N}(n < 0 \cap n > 0)).$$

This statement is a *false* in the set N of natural numbers because the statement $n < 0$ is false for all natural numbers and $F \Rightarrow B$ is a true implication for any logical value of B, so $\forall n\ (n < 0 \Rightarrow n > 0)$ is a true statement and $\exists n\ (n < 0 \cap n > 0)$ is obviously false in the set N of natural numbers.

The restricted quantifiers law corresponding to the predicate tautology (3) is:

$$\models\ (\forall_{B(x)}\ A(x) \Rightarrow (\exists x\ B(x) \Rightarrow \exists_{B(x)}\ A(x))). \tag{21}$$ rq3

By definition 22 and restricted quantifiers *transformations rules* (7), (8) proving (19) is means proving

$$\models (\forall x(B(x) \Rightarrow A(x)) \Rightarrow (\exists x\ B(x) \Rightarrow \exists x\ (B(x) \cap A(x)))).$$

We leave the proof and an exercise for the reader.

3 Predicate Tautologies

sec:predtaut

We have already proved in Fact 3 the basic predicate tautology

$$\models (\forall x\ A(x) \Rightarrow \exists x\ A(x)).$$

We are going to prove now the following.

Fact 5 (Dictum de Omni)

For any formula $A(x)$ of $\mathcal{L}$,

$$\models (\forall x\ A(x) \Rightarrow A(t)), \qquad \models (\forall x\ A(x) \Rightarrow A(x)), \tag{22}$$ p3

$$\models (A(t) \Rightarrow \exists x\ A(x)), \tag{23}$$ p4

where t is a term, A(t) is a result of substitution of t for all free occurrences of x in A(x), and t is free for x in A(x) (definition 7), i.e. no occurrence of a variable in t becomes a bound occurrence in A(t).

Proof of (22) is constructed in a sequence of steps. We leave details to the reader to complete as an exercise. Here are the steps.

S1 Consider a structure $\mathbf{M} = [U, I]$ and $s : VAR \longrightarrow U$. Let $t,\ u$ be two terms. Denote by t' a result of replacing in t all occurrences of a variable x by the term u, i.e. $t' = t(x/u)$. Let s' results from s by replacing $s(x)$ by $s_I(u)$. We prove by induction over the length of t that

$$s_I(t(x/u)) = s_I(t') = {s'}_I(u). \tag{24}$$ st

S2 Let t be free for x in $A(x)$. $A(t)$ is a results from $A(x)$ by replacing t for all free occurrences of x in $A(x)$, i.e. $A(t) = A(x/t)$. Let $s : VAR \longrightarrow U$ and s' be obtained from s by replacing $s(x)$ by $s_I(u)$. We use (24) and induction on the number of connectives and quantifiers in $A(x)$ and prove

$$(\mathbf{M}, s)\ \models A(x/t) \text{ if and only if } (\mathbf{M}, s')\ \models A(x). \tag{25}$$ Mt

S3 Directly from definition 13 and (25) we get that for any $\mathbf{M} = (U, I)$ and any $s : VAR \longrightarrow U$,

$$\text{if } (\mathbf{M}, s)\ \models \forall x A(x), \text{ then } (\mathbf{M}, s)\ \models A(t).$$

This proves that $(\forall x\ A(x) \Rightarrow A(t))$ is a predicate tautology. Observe that a term x is free for x in A(x), so we also get as a particular case of $t = x$ that $\models (\forall x\ A(x) \Rightarrow A(x))$.

Proof of (23) follows from (22), theorem 1, property 1, theorem 5, and definability law (49). We carry it as follows. First we observe that by theorem 1 we

have that $\models ((\forall x \neg A(x) \Rightarrow \neg A(t)) \Rightarrow (A(t) \Rightarrow \neg \forall x \neg A(x)))$ as a substitution of propositional tautology $((a \Rightarrow \neg b) \Rightarrow (b \Rightarrow \neg a))$. By just proved (22) we have that $\models (\forall x \neg A(x) \Rightarrow \neg A(t))$ for $A(x)$ being a formula $\neg A(x)$. By $P4$ in property 1, we get $\models (A(t) \Rightarrow \neg \forall x \neg A(x))$. We apply the existential quantifier definability law (49) and equivalence substitution theorem 5 and get $\models (A(t) \Rightarrow \exists x\ A(x))$. This ends the proof of (22).

Remark the restrictions in (22) and (23) are essential. Here is a simple example explaining why they are needed in (22). The example for (23) is similar.

Let $A(x)$ be a formula $\neg \forall y\ P(x,y)$, for $P \in \mathbf{P}$. Notice that a term $t = y$ is *not free for* y in A(x). Consider (22) $A(x) = \neg \forall y\ P(x,y)$ and $t = y$.

$$(\forall x \neg \forall y\ P(x,y) \Rightarrow \neg \forall y\ P(y,y)), \tag{26}$$ Pp3

Take $\mathbf{M} = [N, I]$ for I such that $P_I := =$. Obviously, $\mathbf{M} \models \forall x \neg \forall y\ P(x,y)$ as $\forall m \neg \forall n (m = n)$ is a true mathematical statement in the set N of natural numbers. $\mathbf{M} \not\models \neg \forall y\ P(y,y)$ as $\neg \forall n\ (n = n)$ is a false statement for $n \in N$. Hence $\mathbf{M}$ is a counter model for for (26) and we proved that without the restriction (22) does not hold.

Here are some useful and easy to prove properties of the notion "t free for x in $A(x)$" (definition 7).

termprop **Property 3 (t free for x in $A(x)$)**

For any formula $A \in \mathcal{F}$ and any term $t \in \mathbf{T}$ the following properties hold.

P1. A closed tern t, i.e. term with no variables is free for any variable x in A.

P2. A term t is free for any variable in A if none of the variables in t is bound in A.

P3. Term $t = x$ is free for x in any formula A.

P4. Any term is free for x in A if A contains no free occurrences of x.

Here are some more important predicate tautologies.

Generalization

For any formulas $A(x), B(x), A, B$ of $\mathcal{L}$, where A, B does not contain any free occurrences of x,

$$\models ((B \Rightarrow A(x)) \Rightarrow (B \Rightarrow \forall x\ A(x))), \tag{27}$$ gen2

$$\models ((B(x) \Rightarrow A) \Rightarrow (\exists x B(x) \Rightarrow A)), \tag{28}$$ gen3

Distributivity 1

For any formulas $A(x), B(x), A, B$ of $\mathcal{L}$, such that A , B does not contain any free occurrences of x,

$$\models \ (\forall x(A \Rightarrow B(x)) \Rightarrow (A \Rightarrow \forall x \ B(x))), \tag{29}$$ Mdistr

$$\models \forall x(A(x) \Rightarrow B) \Rightarrow (\exists x A(x) \Rightarrow B) \tag{30}$$ 1dis1

$$\models \ \exists x(A(x) \Rightarrow B) \Rightarrow (\forall x A(x) \Rightarrow B) \tag{31}$$ 1dis2

The restrictions that the formulas A, B do not contain any free occurrences of x is essential for both *Generalization* and *Distributivity 1* tautologies.

Here is a simple example explaining why they are needed in (29). The relaxation of the assumption that A, B do not contain any free occurrences of x would lead to the following disaster. Let A and B(x) be both atomic formula P(x). Thus x is free in A and we have the following instance of (29).

$$(\forall x(P(x) \Rightarrow P(x)) \Rightarrow (P(x) \Rightarrow \forall x \ P(x))).$$

Observe that $\forall x(P(x) \Rightarrow P(x))$ is a predicate tautology. Take $\mathbf{M} = [N, I]$ for I such that $P_I = ODD$, where $ODD \subseteq N$ is the set of odd numbers. Let $s : VAR \longrightarrow N$. By definition if I, $s_I(x) \in P_I$ if and only if $s_I(x) \in ODD$. Then obviously $(\mathbf{M}, s) \not\models \forall x \ P(x)$ and $\mathbf{M} = [N, I]$ is a counter model for (29) as $(\mathbf{M}, s) \models \forall x(P(x) \Rightarrow P(x))$.

The examples for (30), (31), and (29) similar.

Distributivity 2

For any formulas $A(x), B(x)$ of $\mathcal{L}$,

$$\models \ (\exists x \ (A(x) \cap B(x)) \ \Rightarrow \ (\exists x A(x) \cap \exists x B(x))), \tag{32}$$ 2dis1

$$\models \ ((\forall x A(x) \cup \forall x B(x)) \Rightarrow \forall x \ (A(x) \cup B(x))), \tag{33}$$ 2dis2

$$\models (\forall x(A(x) \Rightarrow B(x)) \Rightarrow (\forall x A(x) \Rightarrow \forall x B(x))) \tag{34}$$ 2dis3

The converse mplications to (32), (33), (34) are not a predicate tautologies , i.e. there are formulas $A(x), B(x)$, such that

$$\not\models \ ((\exists x A(x) \cap \exists x B(x)) \Rightarrow \exists x \ (A(x) \cap B(x))). \tag{35}$$ n2dis1

$$\not\models \ (\forall x \ (A(x) \cup B(x)) \Rightarrow (\forall x A(x) \cup \forall x B(x))), \tag{36}$$ n2dis2

$$\not\models ((\forall x A(x) \Rightarrow \forall x B(x)) \Rightarrow \forall x(A(x) \Rightarrow B(x))) \tag{37}$$ n2dis3

To prove (35), (36) we have to find particular formulas $A(x), B(x) \in \mathcal{F}$ and counter models $\mathbf{M} = [U, I]$ for these particular cases.

Consider (35). We take as $A(x), B(x)$ atomic formulas $Q(x, c), P(x, c)$. The particular case of (35) is now a formula

$$((\exists x P(x, c) \cap \exists x Q(x, c)) \Rightarrow \exists x \ (P(x, c) \cap Q(x, c))).$$

Take $\mathbf{M} = [R, I]$ where R is the set of real numbers, and the interpretation I is $Q_I :>,\ P_I :<,\ c_I : 0$. The particular case formula becomes an obviously *false* mathematical statement

$$((\exists_{x\in R}\ x > 0 \cap \exists_{x\in R}\ x < 0) \Rightarrow \exists_{x\in R}\ (x > 0 \cap x < 0)).$$

Consider (36). We take as Let $A(x), B(x)$ be atomic formulas $Q(x, c), R(x, c)$. The particular case of (36) is now a formula

$$(\forall x Q(x, c) \cup R(x, c)) \Rightarrow (\forall x Q(x, c) \cup \forall x R(x, c))).$$

Take $\mathbf{M} = (R, I)$ where R is the set of real numbers and $Q_I :\geq,\ R_I :<,\ c_I : 0$. The particular formula becomes an obviously *false* mathematical statement

$$(\forall_{x\in R}\ (x \geq 0 \cup x < 0) \Rightarrow (\forall_{x\in R}\ x \geq 0 \cup \forall_{x\in R}\ x < 0)).$$

De Morgan

For any formulas $A(x), B(x)$ of $\mathcal{L}$,

$$\models (\neg\forall x A(x) \Rightarrow \exists x \neg A(x)), \quad (38) \quad \boxed{\texttt{1mall}}$$

$$\models (\neg\exists x A(x) \Rightarrow \forall x \neg A(x)), \quad (39) \quad \boxed{\texttt{1mex}}$$

$$\models (\exists x \neg A(x) \Rightarrow \neg\forall x A(x)), \quad (40) \quad \boxed{\texttt{2mall}}$$

$$\models (\forall x \neg A(x)) \Rightarrow \neg\exists x A(x)). \quad (41) \quad \boxed{\texttt{1mex}}$$

We prove (38) as an example.The proofs of all other laws are similar. Assume that (38) does not hold. By definition 16 there is $\mathbf{M} = (U, I)$ and $s : VAR \longrightarrow U$, such that $(\mathbf{M}, s) \models \neg\forall x A(x))$ and $(\mathbf{M}, s) \not\models \exists x \neg A(x)$.
Consider $(\mathbf{M}, s) \models \neg\forall x A(x)$. By satisfaction definition 13, $(\mathbf{M}, s) \not\models \forall x A(x)$. This holds only if for all s', such that s, s' agree on all variables except on x, $(\mathbf{M}, s') \not\models A(x)$.
Consider $(\mathbf{M}, s) \not\models \exists x \neg A(x)$. This holds only if there is no s', such that $(\mathbf{M}, s') \models \neg A(x)$, i.e. there is no s', such that $(\mathbf{M}, s') \not\models A(x)$. This means that for all s', $(\mathbf{M}, s') \models A(x)$. Contradiction with $(\mathbf{M}, s') \not\models A(x)$.

Quantifiers Alternations

For any formula $A(x, y)$ of $\mathcal{L}$,

$$\models (\exists x \forall y A(x, y) \Rightarrow \forall y \exists x A(x, y)). \quad (42) \quad \boxed{\texttt{qalt}}$$

The converse implications to (42) is not a predicate tautology. Take as $A(x, y)$ an atomic formulas $R(x, y)$. Take $\mathbf{M} = (R, I)$ where R is the set of real numbers and $R_I :<$. The instance of (42) particular formula becomes a mathematical statement

$$(\forall y \exists x (x < y) \Rightarrow \exists x \forall y (x < y))$$

that obviously *false* in the set of real numbers. We proved

$$\not\models (\forall y \exists x A(x, y) \Rightarrow \exists x \forall y A(x, y)). \quad (43) \quad \boxed{\texttt{nqalt}}$$

3.1 Equational Laws of Quantifiers

The most frequently used laws of quantifiers have a form of a *logical equivalence*, symbolically written as $\equiv$. This not a new logical connective. This is a very useful symbol. It has the same properties as the equality $=$ and can be used in the same way we use the equality symbol $=$.

Note that we use the same equivalence symbol $\equiv$ and the tautology symbol $\models$ for propositional and predicate languages and semantics when there is no confusion. Formally we define the predicate equivalence as follows.

def1:peq **Definition 23 (Logical equivalence)**

For any formulas $A,\ B \in \mathcal{F}$ *of the* **predicate language** $\mathcal{L}$*,*

$$A \equiv B \quad \textit{if and only if } \models (A \Rightarrow B) \textit{ and } \models (B \Rightarrow A).$$

Remark that our predicate language $\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ we defined the semantics for (definition 13) does not include the equivalence connective $\Leftrightarrow$. If it does we extend the satisfaction definition 13 in a natural way and can adopt the following definition 24 of logical equivalence that is obviously equivalent definition to the propositional one and to our definition 23.

def2:peq **Definition 24**

For any formulas $A, B \in \mathcal{F}$ *of the* **predicate language** $\mathcal{L}$*,*

$$A \equiv B \quad \textit{if and only if } \models (A \Leftrightarrow B).$$

We re-write the basic theorem 1 establishing relationship between propositional and some predicate tautologies as follows.

thm1 **Theorem 3 (Tautologies)**

If a formula A *is a propositional tautology, then by substituting for propositional variables in* A *any formula of the predicate language* $\mathcal{L}$ *we obtain a formula which is a predicate tautology.*

Directly from the theorem 3 and logical equivalence definition 23 we get that the following is true.

thm2 **Theorem 4 (Equivalences)**

Given propositional formulas A, B*.*
If $A \equiv B$ *is a propositional equivalence, and* $A',\ B'$ *are formulas of the predicate language* L *obtained by a substitution of any formula of* $\mathcal{L}$ *for propositional variables in* A *and* B*, respectively, then* $A' \equiv B'$ *holds under predicate semantics.*

e2 **Example 19**

Consider the following propositional logical equivalence:

$$(a \Rightarrow b) \equiv (\neg a \cup b).$$

Substituting $\exists x P(x,z)$ for a, and $\forall y R(y,z)$ for b, we get from theorem 4 that the following equivalence holds:

$$(\exists x P(x,z) \Rightarrow \forall y R(y,z)) \equiv (\neg \exists x P(x,z) \cup \forall y R(y,z)).$$

We prove in similar way as in the propositional case (chapter 3) the following.

thm:subeq **Theorem 5 (Equivalence Substitution)**

Let a formula B_1 be obtained from a formula A_1 by a substitution of a formula B for one or more occurrences of a sub-formula A of A_1, what we denote as

$$B_1 = A_1(A/B).$$

Then the following holds for any formulas $A,\ A_1,\ B,\ B_1$ of $\mathcal{L}$.

$$\textit{If} \quad A \equiv B, \quad \textit{then} \quad A_1 \equiv B_1. \tag{44}$$ thm:leqv

Directly from the Dictum de Omi (22) and the Generalization (27) tautologies we get the proof of the following theorem 6 useful for building new logical equivalences from the old, known ones.

thm3 **Theorem 6**

For any formulas $A(x), B(x)$ of $\mathcal{L}$.

$$\textit{if} \ \ A(x) \equiv B(x), \ \ \textit{then} \ \ \forall x A(x) \equiv \forall x B(x),$$

$$\textit{if} \ \ A(x) \equiv B(x), \ \ \textit{then} \ \ \exists x A(x) \equiv \exists x B(x).$$

e3 **Example 20**

We know from the example 19 that the formulas $(\exists x P(x,z) \Rightarrow \forall y R(y,z))$ and $(\neg \exists x P(x,z) \cup \forall y R(y,z))$ are logically equivalent. We get, as the direct consequence of the theorem 6 the following equivalences:

$$\forall z(\exists x P(x,z) \Rightarrow \forall y R(y,z)) \equiv \forall z(\neg \exists x P(x,z) \cup \forall y R(y,z)),$$

Theorem 4 and theorem 6 allow us to use propositional tautologies and predicate formulas to build predicate equivalences. Here is a simple example.

Exercise 6

Prove that for any formulas $A(x),\ B(x)$ *of* $\mathcal{L}$

$$\neg\forall x\neg(A(x)\cup B(x))\equiv\neg\forall x(\neg A(x)\cap\neg B(x)). \tag{45}$$ exer

Solution
By the substituting $A(x)$ for a, and any formula $B(x)$ for b, in the propositional de Morgan Law: $\neg(a\cup b)\equiv(\neg a\cap\neg b)$, we get via theorem 4 that

$$\neg(A(x)\cup B(x))\equiv(\neg A(x)\cap\neg B(x)).$$

Applying the theorem 6 to the above we obtain that

$$\forall x\neg(A(x)\cup B(x))\equiv\forall x(\neg A(x)\cap\neg B(x)).$$

We know, from the propositional logic, that for any propositional variables a, b, $a\equiv b$ if and only if $\neg a\equiv\neg b$. Substituting $\forall x\neg(A(x)\cup B(x))$ and $\forall x(\neg A(x)\cap\neg B(x))$ for a and b, respectively, we get that

$$\forall x\neg(A(x)\cup B(x))\equiv\forall x(\neg A(x)\cap\neg B(x))$$

if and only if

$$\neg\forall x\neg(A(x)\cup B(x))\equiv\neg\forall x(\neg A(x)\cap\neg B).$$

But we have proved that $\forall x\neg(A(x)\cup B)\equiv\forall x(\neg A(x)\cap\neg B)$ holds, so we conclude that the equivalence (45) also holds.

e4 **Exercise 7**

Prove that for any formulas $A(x),\ B$ *of* $\mathcal{L}$

$$\forall x\neg(A(x)\cup B)\equiv\forall x(\neg A(x)\cap\neg B)$$

Solution
By the substituting $A(x)$ for a, and any formula B for b, in the propositional de Morgan law: $\neg(a\cup b)\equiv(\neg a\cap\neg b)$, we get that

$$\neg(A(x)\cup B)\equiv(\neg A(x)\cap\neg B).$$

Applying the theorem 4 to the above we obtain that

$$\forall x\neg(A(x)\cup B)\equiv\forall x(\neg A(x)\cap\neg B).$$

As we can see, it is possible to obtain a fair amount of predicate tautologies from the propositional tautologies and theorems 3, 4 and 6, but as we have proved will for never obtain for example the most basic law: $(\forall xA(x)\Rightarrow\exists xA(x))$, any many the most important others.

We concentrate now only on these laws which have a form of a logical equivalence.They are called the **equational laws for quantifiers**.

Directly from the definition 23 and the de Morgan tautologies (38)-(41) we get one of the most important equational laws, called also De Morgan Laws.

De Morgan Laws

$$\neg\forall x A(x) \equiv \exists x \neg A(x) \quad (46) \quad \boxed{\texttt{mall}}$$

$$\neg\exists x A(x) \equiv \forall x \neg A(x) \quad (47) \quad \boxed{\texttt{mexists}}$$

Now we will apply them to show that the quantifiers can be defined one by the other i.e. that the following Definability Laws hold.

Definability Laws

$$\forall x A(x) \equiv \neg\exists x \neg A(x) \quad (48) \quad \boxed{\texttt{q1}}$$

$$\exists x A(x) \equiv \neg\forall x \neg A(x) \quad (49) \quad \boxed{\texttt{q2}}$$

The law (48) is often used as a definition of the universal quantifier in terms of the existential one (and negation), the law (49) is a **definition** of the existential quantifier in terms of the universal one (and negation).

Proof of (48)
Substituting any formula $A(x)$ for a variable a in the propositional equivalence $a \equiv \neg\neg a$ we get by theorem 4 that $A(x) \equiv \neg\neg A(x)$. Applying the theorem 6 to the above we obtain $\exists x A(x) \equiv \exists x \neg\neg A(x)$. By the de Morgan Law (46) $\exists x \neg\neg A(x) \equiv \neg\forall x \neg A(x)$ and hence $\exists x A(x) \equiv \neg\forall x \neg A(x)$, what ends the proof.

Proof of (49)
We obtain $\forall x A(x) \equiv \forall \neg\neg A(x)$ in a similar way as above. By the de Morgan Law (47), $\forall \neg\neg A(x) \equiv \neg\exists \neg A(x)$ and hence $\forall x A(x) \equiv \neg\exists \neg A(x)$, what ends the proof.

Other important equational laws are the following introduction and elimination laws. We prove later the first two of them. We show that the laws (52) - (57) can be deduced from laws (50) and (51), the de Morgan laws (46), (47), definability laws (48), (49), propositional tautologies and theorems 3, 4, and theorem 5.

Introduction and Elimination Laws

If B is a formula such that B **does not contain any free occurrence** of x, then the following logical equivalences hold.

$$\forall x(A(x) \cup B) \equiv (\forall x A(x) \cup B) \quad (50) \quad \boxed{1}$$

$$\forall x(A(x) \cap B) \equiv (\forall x A(x) \cap B) \quad (51) \quad \boxed{2}$$

$$\exists x(A(x) \cup B) \equiv (\exists x A(x) \cup B) \quad (52) \quad \boxed{3}$$

$$\exists x(A(x) \cap B) \equiv (\exists x A(x) \cap B) \quad (53) \quad \boxed{4}$$

$$\forall x(A(x) \Rightarrow B) \equiv (\exists x A(x) \Rightarrow B) \quad (54) \quad \boxed{5}$$

$$\exists x(A(x) \Rightarrow B) \equiv (\forall x A(x) \Rightarrow B) \quad (55) \quad \boxed{6}$$

$$\forall x(B \Rightarrow A(x)) \equiv (B \Rightarrow \forall x A(x)) \quad (56) \quad \boxed{7}$$

$$\exists x(B \Rightarrow A(x)) \equiv (B \Rightarrow \exists x A(x)) \quad (57) \quad \boxed{8}$$

The equivalences (50)-(53) make it possible to introduce a quantifier that precedes a disjunction or a conjunction into one component on the condition that the other component does not contain any free occurrence of a variable which is bound by that quantifier. These equivalences also make possible to eliminate a quantifier from a component of a disjunction or a conjunction and to place it before that disjunction or conjunction as a whole, on the condition that the other component does not contain any free occurrence of a variable which that quantifier would then bind.

The equivalences (54)-(57)make it possible to introduce a quantifier preceding an implication into the consequent of that implication, on the condition that that antecedent does not contain any free occurrence of a variable which is bound by that quantifier; they also make it possible to introduce a universal quantifier preceding an implication into the consequent of that implication while changing it into an existential quantifier in the process, on the condition that the consequent of that implication does not contain any free occurrence of a variable bound by that quantifier. Equivalences (54)-(57) further enable us to eliminate quantifiers from the antecedent of an implication to the position preceding the whole implication, while changing a universal quantifier into an existential one, and vice versa, in the process, and also to eliminate quantifiers from the consequent of an implication to the position preceding the whole implication; the conditions that the other component of the implication in question does not contain any free occurrence of a variable which that quantifier would then bind, must be satisfied, respectively.

As we said before, the equivalences (50)-(57) are not independent, some of them are the consequences of the others. Assuming that we have already proved (50) and (51), the proof of (52) is as follows.

Proof of (52) $\exists x(A(x) \cup B)$ is logically equivalent, by the definability law (49) to $\neg\forall x\neg(A(x) \cup B)$. By the reasoning presented in the proof of (45) for B instead of $B(x)$, we have that $\neg\forall x\neg(A(x) \cup B) \equiv \neg\forall x(\neg A(x) \cap \neg B)$. By the introduction law (51), $\neg\forall x(\neg A(x) \cap \neg B) \equiv \neg(\forall x\neg A(x) \cap \neg B)$. Substituting $\forall x\neg A(x)$ for a and $\neg B$ for b in propositional equivalence $\neg(a \cap \neg b) \equiv (\neg a \cup \neg\neg b)$, we get, by the theorem 4 that $\neg(\forall x\neg A(x) \cap \neg B) \equiv (\neg\forall x\neg A(x) \cup \neg\neg B)$. In a similar way we prove that $\neg\neg B \equiv B$, by the definability law (49) $\neg\forall x\neg A(x) \equiv \exists x A(x)$, hence by theorem 5 $(\neg\forall x\neg A(x) \cup \neg\neg B) \equiv (\exists x A(x) \cup B)$ and finally, $\exists x(A(x) \cup B) \equiv (\exists x A(x) \cup B)$, what end the proof.

We can write this proof in a shorter, symbolic way as follows:

$$\begin{aligned}
\exists x(A(x) \cup B) &\overset{\text{law 49}}{\equiv} \neg\forall x\neg(A(x) \cup B) \\
&\overset{\text{thm 3, 4}}{\equiv} \neg\forall x(\neg A(x) \cap \neg B) \\
&\overset{\text{law 51}}{\equiv} \neg(\forall x\neg A(x) \cap \neg B) \\
&\overset{\text{(46), thm 5}}{\equiv} (\neg\forall x\neg A(x) \cup \neg\neg B) \\
&\overset{\text{thm 5}}{\equiv} (\exists x A(x) \cup B)
\end{aligned}$$

Distributivity Laws

Let $A(x), B(x)$ be any formulas with a free variable x.
Law of distributivity of **universal quantifier** over **conjunction**

$$\forall x\ (A(x) \cap B(x)) \ \equiv\ (\forall x A(x) \cap \forall x B(x)) \tag{58}$$ univ

Law of distributivity of **existential quantifier** over **disjunction**.

$$\exists x\ (A(x) \cup B(x)) \ \equiv\ (\exists x A(x) \cup \exists x B(x)) \tag{59}$$ exist

Alternations of Quantifiers Laws

Let $A(x, y)$ be any formula with a free variables x,y.

$$\forall x\forall y\ (A(x, y) \ \equiv\ \forall y\forall x\ (A(x, y) \tag{60}$$ alt1

$$\exists x\exists y\ (A(x, y) \ \equiv\ \exists y\exists x\ (A(x, y) \tag{61}$$ alt2

Renaming the Variables

Let $A(x)$ be any formula with a free variable x and let y be a variable that **does not occur** in A(x).
Let $A(y)$ be a result of **replacement** of each occurrence of x by y, then the following holds.

$$\forall x A(x) \equiv \forall y A(y), \tag{62}$$ v1

$$\exists x A(x) \equiv \exists y A(y). \tag{63}$$ v2

Restricted De Morgan Laws

For any formulas $A(x), B(x) \in \mathcal{F}$ with a free variable x,

$$\neg\forall_{B(x)}\ A(x) \equiv \exists_{B(x)}\ \neg A(x), \qquad \neg\exists_{B(x)}\ A(x) \equiv \forall_{B(x)}\neg A(x). \tag{64}$$ RdeMorgan

Here is a poof of first equality. The proof of the second one is similar and is left as an exercise.

$$\neg\forall_{B(x)}\ A(x) \equiv \neg\forall x\ (B(x) \Rightarrow A(x)) \equiv \neg\forall x\ (\neg B(x) \cup A(x)) \equiv \exists x\ \neg(\neg B(x) \cup A(x))$$

$$\equiv \exists x\ (\neg\neg B(x) \cap \neg A(x)) \equiv \exists x\ (B(x) \cap \neg A(x)) \equiv \exists_{B(x)}\ \neg A(x)).$$

Restricted Introduction and Elimination Laws

If B is a formula such that B *does not contain any free occurrence* of x, then the following logical equivalences hold for any formulas $A(x), B(x), C(x)$.

$$\forall_{C(x)}(A(x) \cup B) \equiv (\forall_{C(x)} A(x) \cup B), \tag{65}$$ r1

$$\exists_{C(x)}\ (A(x) \cap B) \equiv (\exists_{C(x)}\ A(x) \cap B), \tag{66}$$ r4

$$\forall_{C(x)}(A(x) \Rightarrow B) \equiv (\exists_{C(x)} A(x) \Rightarrow B), \tag{67}$$ r5

$$\forall_{C(x)}(B \Rightarrow A(x)) \equiv (B \Rightarrow \forall_{C(x)} A(x)). \tag{68}$$ r7

The proofs are similar to the proof of the restricted de Morgan Laws.
The similar generalization of the other *Introduction and Elimination Laws* (51), (52), (55), (57) for restricted domain quantifiers fails. We can easily follow the proof of (18) and construct proper counter-models proving the following.

$$\exists_{C(x)}(A(x) \cup B) \not\equiv (\exists_{C(x)} A(x) \cup B),$$

$$\forall_{C(x)}(A(x) \cap B) \not\equiv (\forall_{C(x)} A(x) \cap B),$$

$$\exists_{C(x)}(A(x) \Rightarrow B) \not\equiv (\forall_{C(x)} A(x) \Rightarrow B),$$

$$\exists_{C(x)}(B \Rightarrow A(x)) \not\equiv (B \Rightarrow \exists x A(x)).$$

Nevertheless it is possible to correctly generalize them all as to cover quantifiers with restricted domain. We show it in a case of (51) and leave the other cases to the reader as an exercise.

res2 **Example 21**

The restricted quantifiers version of (51) is the following.

$$\exists_{C(x)}(A(x) \cup B) \equiv (\exists_{C(x)} A(x) \cup (\exists x\ C(x) \cap B)). \tag{69}$$ r2

We derive (74) as follows.

$$\exists_{C(x)}(A(x) \cup B) \equiv \exists x(C(x) \cap (A(x) \cup B)) \equiv \exists x((C(x) \cap A(x)) \cup (C(x) \cap B))$$

$$\equiv (\exists x(C(x) \cap A(x)) \cup \exists x(C(x) \cap B)) \equiv (\exists_{C(x)} A(x) \cup (\exists x\ C(x) \cap B)).$$

We leave it as an exercise to specify and write references to transformation or equational laws used at each step of our computation.

4 Proof Systems: Soundness and Completeness

We adopt now general definitions from chapter **??** concerning proof systems to the case of classical first order (predicate) logic.

We refer the reader to chapters **??** and **??** for a great array of examples, exercises, homework problems explaining in a great detail all notions we introduce here for the predicate case. The examples and exercises we provide here are not numerous and restricted to the laws of quantifiers.

Given a language $\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P},\mathbf{F},\mathbf{C})$. Any proof system

$$S = (\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P},\mathbf{F},\mathbf{C}), \quad \mathcal{F}, \quad LA, \quad \mathcal{R}) \tag{70}$$

Psys

is a **predicate (first order)** proof system.

The predicate proof system (70) is a **Hilbert** proof system if the set $\mathcal{R}$ of its rules contains the Modus Ponens rule

$$(MP) \quad \frac{A \ ; \ (A \Rightarrow B)}{B},$$

where $A, B \in \mathcal{F}$.

Semantic Link: Logical Axioms LA

We want the set LA of logical axioms to be a non-empty set of classical predicate tautologies (13), i.e.

$$LA \subseteq \mathbf{T}_p,$$

where

$$\mathbf{T}_p = \{A \ \text{ of } \ \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P},\mathbf{F},\mathbf{C}) : \ \models_p A\}.$$

Remark 2

We use symbols $\models_p$, $\mathbf{T}_p$ *to stress the fact that we talk about predicate lah=nguage and classical predicate tautologies.*

Semantic Link 2: Rules of Inference $\mathcal{R}$

We want the the rules of inference $r \in \mathcal{R}$ of S to preserve truthfulness. Rules that do so are called sound. We define it formally as follows.

d:sound **Definition 25 (Sound Rule)**

Given an inference rule $r \in \mathcal{R}$ *of the form*

$$(r) \quad \frac{P_1 \ ; \ P_2 \ ; \ .... \ ; \ P_m}{C},$$

where $P_1.P_2, \ldots, P_m, C \in \mathcal{F}$.

(i) We say that the rule (r) *is* **sound** *if and only if the following condition holds for all structures* $\mathbf{M} = [U, I]$ *for* $\mathcal{L}$.

$$If \ \mathbf{M} \models \{P_1, P_2, .P_m\} \ then \ \mathbf{M} \models C. \tag{71}$$ r-sd

(ii) The rule (r) *is* **not sound** *if and only if there is a structure* $\mathbf{M} = [U, I]$, *such that*

$$\mathbf{M} \models \{P_1, P_2, .P_m\} \ and \ \mathbf{M} \not\models C. \tag{72}$$ n-s

In order to prove that the rule (r) is **sound** we have to show the implication (71). It means, by definitions 14, 18, we have to show that that if all premisses of the rule (r) are **true** in $\mathbf{M} = [U, I]$, so is its conclusion. This also justifies correctness of the definition 25; sound rules do preserve the **truthfulness** as it is defined in our semantics.

r1 **Exercise 8**

Prove the soundness of the rule

$$(r1) \ \frac{\neg\forall x A(x)}{\exists x \neg A(x)}. \tag{73}$$ r1

Proof
Assume that the soundness condition (71) does not hold for for all structures $\mathbf{M} = [U, I]$. It means we assume that there is a structure $\mathbf{M} = [U, I]$, such that $\mathbf{M} \models \neg\forall x A(x)$ and $\mathbf{M} \not\models \exists x \neg A(x)$.

Let $\mathbf{M} \models \neg\forall x A(x)$. By definition 14, for all $s : VAR \longrightarrow U$ we have $(\mathbf{M}, s) \models \neg\forall x A(x))$. Hence by satisfaction definition 13, $(\mathbf{M}, s) \not\models \forall x A(x)$. This holds only if for all s', such that s, s' agree on all variables except on x, $(\mathbf{M}, s') \not\models A(x)$. Observe that $\mathbf{M} \not\models \exists x \neg A(x)$ only if there is no s', such that $(\mathbf{M}, s') \models \neg A(x)$, i.e. there is no s', such that $(\mathbf{M}, s') \models A(x)$. This means that for all s', $(\mathbf{M}, s') \models A(x)$. Contradiction with $(\mathbf{M}, s') \not\models A(x)$.

r2 **Exercise 9**

Prove the soundness of the rule

$$(r2) \ \frac{\forall x A(x)}{\exists x A(x)}. \tag{74}$$ r2

Proof
Assume that the soundness condition (71) does not hold for for all structures $\mathbf{M} = [U, I]$. It means we assume that there is a structure $\mathbf{M} = [U, I]$, such that $\mathbf{M} \models \forall x A(x)$ and $\mathbf{M} \not\models \exists x A(x)$.

Let $\mathbf{M} \models \forall x A(x)$. By definition 14, for all $s : VAR \longrightarrow U$ we have $(\mathbf{M}, s) \models \forall x A(x))$.

By definition 13, $(\mathbf{M}, s) \models \forall x\ A(x)$ and $(\mathbf{M}, s) \not\models \exists x\ A(x)$. It means that $(\mathbf{M}, s') \models A(x)$ for all s' such that s, s' agree on all variables except on x, and it is not true that there is s' such that s, s' agree on all variables except on x, and $(\mathbf{M}, s') \models A(x)$. This is impossible and this contradiction proves soundness of $(r2)$.

er3 **Exercise 10**

Prove that the rule

$$(r3)\quad \frac{\exists x A(x)}{\forall x A(x)}. \tag{75}$$ r3

is not sound.

Proof
Observe that to prove that the rule (75) is not sound we have to provide an example of an instance of a formula $A(x)$ and prove (ii) of definition 25 for it.

Let $A(x)$ be an atomic formula $P(x, c)$, for any $P \in \mathbf{P}, \#P = 2$. We take as $\mathbf{M} = (N, P_I :<,\ c_I : 3)$ for N set of natural numbers. Let s be any assignment $s : VAR \longrightarrow N$. Obviously $(\mathbf{M}, s) \models \exists x\ P(x, c)$.

Take any s' such that $s'(x) = 2$ and $s'(y) = s(y)$ for all $y \in VAR - \{x\}$. We have $(2, 3) \in P_I$, as $2 < 3$ and hence there exists s' that agrees with s on all variables except on x, and $(\mathbf{M}, s') \models P(x, c)$. But $(\mathbf{M}, s) \not\models \forall x\ P(x, c)$ as for example for s' such that $s'(x) = 5$ and $s'(y) = s(y)$ for all $y \in VAR - \{x\}$, $(2, 3) \notin P_I$, as $5 \not< 3$.

This proves that $\mathbf{M} = (N, P_I :<,\ c_I : 3)$ is a model for $(\exists x\ P(x, c)$ and hence $\not\models\ \forall x\ A(x))$.

The "shorthand" solution is: the formula $(\exists x\ P(x, c)$ becomes in $\mathbf{M} = (N, P_I :<$, $c_I : 3)$ a true mathematical statement (written with logical symbols): $\exists n\ n < 3$. The formula $(\forall x\ P(x, c)$ becomes a mathematical frmula $\forall n\ n < 3$ which is an obviously *false* statement in the set N of natural numbers, as there is $n \in N$, such that $n < 3$, for example $n = 2$, and it is not true that all natural numbers are smaller then 3. So the rule $(r3)$ is not sound.

d:ssound **Definition 26 (Strongly Sound Rule)**

An inference rule $r \in \mathcal{R}$ of the form

$$(r)\quad \frac{P_1\ ;\ P_2\ ;\ ....\ ;\ P_m}{C}$$

is **strongly sound** *if the following condition holds for all structures $\mathbf{M} = [U, I]$ for $\mathcal{L}$.*

$$\mathbf{M} \models \{P_1, P_2, .P_m\} \quad \textit{if and only if} \quad \mathbf{M} \models C. \tag{76}$$ sseq

We can, and we do state it informally as: " an inference rule $r \in \mathcal{R}$ is strongly sound when the conjunction of all its premisses is logically equivalent to its conclusion". We denote it informally as

$$P_1 \cap P_2 \cap \ \ldots \ \cap P_m \equiv C. \tag{77}$$ ss-equiv

Example 22

The sound rule (73)

$$(r1) \quad \frac{\neg\forall x A(x)}{\exists x \neg A(x)}$$

is **strongly sound** *by De Morgan Law (46).*

The sound rule (75)

$$(r2) \quad \frac{\forall x A(x)}{\exists x A(x)}$$

is **not strongly sound** *by exercise 10.*

def:sound **Definition 27 (Sound Proof System)**

Given the **predicate (first order)** *proof system (70)*

$$S = (\mathcal{L}, \ \ \mathcal{F}, \ \ LA, \ \ \mathcal{R}).$$

We say that the proof system S is **sound** *if the following conditions hold.*

(1) $LA \subseteq \mathbf{T}_p$*;*

(2) Each rule of inference $r \in \mathcal{R}$ *is* **sound**.

The proof system S is **strongly sound** *if the condition (2) is replaced by the following condition (2')*

(2') Each rule of inference $r \in \mathcal{R}$ *is* **strongly sound** *under* $\mathbf{M}$.

The set of **all provable expressions** of S is denoted by $\mathbf{P}_S$ and is defined as follows.

$$\mathbf{P}_S = \{A \in \mathcal{F} : \ \vdash_S A\}. \tag{78}$$ S-prov

When we define (develop) a proof system S our first goal is to make sure that it a "sound" one, i.e. that all we prove in it is true. Proving the following theorem establishes this goal.

thm:Ss **Theorem 7 (Soundness Theorem for S)**

Given a predicate proof system S.
For any $A \in \mathcal{F}$, the following implication holds.

$$\textit{If } \vdash_S A \quad \textit{then} \quad \models_p A. \tag{79}$$ s-impl

We write (79) it in a more concise form as

$$\mathbf{P}_S \subseteq \mathbf{T}_p. \tag{80}$$ Ss

Proof
Observe that if we have already proven that S is sound as stated in the definition 27, the proof of the implication (79) is a straightforward application of the mathematical induction over the length of the formal proof of the formula A.

It means that in order to prove the Soundness Theorem 7 for a proof system Sit is enough to verify the two conditions of the definition 27 (1) $LA \subseteq \mathbf{T}_p$ and (2) each rule of inference $r \in \mathcal{R}$ is **sound**.

We again refer the reader to chapter **??** for detailed examples, exercises and problems.

As we can see, proving Soundness Theorem 7 for any proof system we develop is indispensable and the proof is quite easy. The next step in developing a logic (classical predicate logic in our case now) is to answer necessary and a difficult question: *Given a proof system S, about which we know that all it proves it true (tautology). Can we prove all we know to be true (all tautologies)?*

Proving the following theorem establishes this goal.

th:compl **Theorem 8 (Completeness Theorem for S)**

Given a predicate proof system S.
For any $A \in \mathcal{F}$, the following holds.

$$\vdash_S A \quad \textit{if and only if} \quad \models_p A. \tag{81}$$ s-comp

We write (81) it in a more concise form as

$$\mathbf{P}_S = \mathbf{T}_p. \tag{82}$$ Scomp

The Completeness Theorem consists of two parts:

Part 1: Soundness Theorem: $\mathbf{P}_S \subseteq \mathbf{T}_p$.

Part 2: Completeness part of the Completeness Theorem: $\mathbf{T}_p \subseteq \mathbf{P}_S$.

Proving the Soundness Theorem for S is usually a straightforward and not a very difficult task. Proving the *Completeness part* of the Completeness Theorem is

always a crucial and very difficult task. There are many methods and techniques for doing so, even for classical proof systems (logics) alone. Non-classical logics often require new sometimes very sophisticated methods. We presented two proofs of the Completeness Theorem for classical propositional Hilbert style proof system in chapter **??**, and a constructive proofs for automated theorem proving systems for classical propositional logic the chapter **??**.

We present a proof of the Completeness Theorem for predicate (first order) logic in the next chapter **??**.

5 Homework Problems

Predicate Languages

1. Given the following formulas $A_1 - A_5$ of a predicate language $\mathcal{L}$.

$$A_1 = R(x, y, g(c, x)),\ A_2 = \exists x P(x, f(x, y)),\ A_3 = \exists d R(x, y, g(c, d)),$$

$$A_4 = \forall z(f(x, P(c, y)),\quad A_5 = \exists y P(x, f(c, y)) \cup \forall y P(x, f(c, y)).$$

   (a) Indicate whether they are, or are not well formed formulas of $\mathcal{F}$. For those which are not in $\mathcal{F}$ write a correct formula.

   (b) For each correct, or corrected formula identify all components: connectives, quantifiers, predicate and function symbols, and list all its terms.

   (c) For each formula identify its s free and bound variables. State which are open and which are closed formulas (sentences), if any.

   (d) Describe a language defined by the set $\mathcal{F}_0 = \{A_1,\ A_2,\ \ldots\ A_5\}$ of formulas that are correct or corrected.

2. For the following mathematical statements write their corresponding formulas of predicate language $\mathcal{L}$.

   (a) $\forall_{n>1}(n + 3 < 8 \cup \exists_{x \in R}\ x + n > 8)$

   (b) $\forall_{x \in R}\ \exists_{n \in N}(x + n > 0 \Rightarrow \exists_{m \in N}(m = x + n))$

   (c) If all natural numbers are smaller then zero, then the sum of any two integers is smaller then zero.

   (d) For all natural numbers The following implication holds for all natural numbers: if $n > 0$, then there is a real number x, such that $n + x = 0$ or there is an integer m, such that $m > 0$.

3. For each of the following formulas (some with restricted quantifiers) write 2 corresponding natural language sentences.

(a) $\forall x(P(x) \Rightarrow \exists y Q(x,y))$.

(b) $\forall x \exists y(P(x) \cap \neg Q(x,y))$.

(c) $\forall_{A(x)} \exists_{A(y)} B(y)$.

(d) $\exists_{P(x)} \forall_{N(x)} R(x,y)$.

4. Is the term $t = f(x,y)$ free for x in the following formulas?

   (a) $(P(x,y) \Rightarrow \forall y P(x,y))$.

   (b) $(\forall y P(x,y) \cup \exists y P(x,y))$.

   (c) $\forall x P(x,y)$.

   (d) $\forall y P(x,y)$.

   (e) $(\forall y Q(y) \Rightarrow P(x,y))$.

5. Justify that for any formula $A \in \mathcal{F}$ and any term $t \in \mathbf{T}$ the following facts hold.

   (a) A closed tern t, i.e. term with no variables is free for any variable x in A.

   (b) A term t is free for any variable in A if none of the variables in t is bound in A.

   (c) Term $t = x$ is free for x in any formula A.

   (d) Any term is free for x in A if A contains no free occurrences of x.

6. Translate the following formulas in everyday English.

   (a) $\forall x(P(x) \cap \forall y(\neg L(x,y) \Rightarrow \neg H(x)))$, where $P(x)$ means " x is a person", $L(x,y)$means "x likes y" , and $H(x)$ means "x is happy".

   (b) $\forall x((E(x) \cap P(x)) \Rightarrow E(x,b))$, where E(x) means " x is an even integer", P(x) means " x is prime", Q(x,y) means " x is equal to x", and b denotes 2.

   (c) $\neg \forall y((P(y) \cap \forall x(P(x)) \Rightarrow E(x,y)))$, where P(x) means "x is an integer, and $L(x,y)$means "$x \leq y$".

7. Use the restricted quantifiers to translate the following natural language sentences into a proper formulas of a proper formal predicate language $\mathcal{L} = \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\neg\}}(\mathbf{P},\mathbf{F},\mathbf{C})$. In each case specify the sets **P, F, C**.

   (a) Some politician are honest, some are not.

   (b) Any sets that have the same elements are equal.

   (c) Somebody hates everyone who does not hate himself.

   (d) Birds can fly and if anyone can fly Tweety can.

   (e) Anyone who knows logic loves it.

Classical Semantics

1. Given a predicate language $\mathcal{L}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ and a structure $\mathbf{M} = [U, I]$ such that $U = N$ and $P_I : =,\ f_I : +,\ g_I : \cdot,\ a_I : 0,\ b_I : 1$ for N set on natural numbers. For each of the following formula A decide whether $\mathbf{M} \models A$ or not. Do so by examining the corresponding mathematical statement defined by $\mathbf{M}$.

   (a) $\forall x \exists y (P(x, f(y, y)) \cup P(x, f(f(y, y), b)))$.

   (b) $\forall x \exists y (P(g(x, y), a) \Rightarrow (P(x, a) \cup P(y, a)))$.

   (c) $\exists y P(f(y, y), b)$.

2. Let $\mathbf{M} = [U, I]$ be a structure such that $U = Z$ and $P_I : =,\ f_I : +$ for Z set of integers. For each of the following formula A decide whether $\mathbf{M} \models A$ or not. Do so by examining the corresponding mathematical statement defined by $\mathbf{M}$.

   (a) $\forall x \forall y P(f(x, y), f(y, x))$.

   (b) $\forall x \forall y \forall z P(f(x, f(y, z)), f(f(x, y), z))$.

   (c) $\forall x \forall y \exists z P(f(x, z), y)$

3. Let $\mathbf{M} = [U, I]$ be a structure such that $U = N - \{0\}$ or N set of natural numbers and $P_I : =,\ f_I(x, y)$ is x^y For each of the following formula A decide whether $\mathbf{M} \models A$ or not. Do so by examining the corresponding mathematical statement defined by $\mathbf{M}$.

   (a) $\forall x \forall y P(f(x, y), f(y, x))$.

   (b) $\forall x \forall y \forall z P(f(x, f(y, z)), f(f(x, y), z))$.

   (c) $\forall x \forall y \exists z P(f(x, z), y)$.

4. For each formula below, where $P, Q \in \mathbf{P}$, find a structure $\mathbf{M} = [U, I]$ for $\mathcal{L} = \mathcal{L}_{\{\neg, \cap, \cup, \Rightarrow, \neg\}}(\mathbf{P}, \mathbf{F}, \mathbf{C})$ that is its counter model. Justify its correctness.

   (a) $(\ (\forall x P(x) \Rightarrow \forall y Q(y)) \Rightarrow (\forall x (P(x) \Rightarrow Q(x))))$.

   (b) $(\ (\forall x P(x) \cup \forall y Q(y)) \Rightarrow (\forall x (P(x) \cup \forall x Q(x))))$.

5. Show that the following formulas are predicate tautologies for any formulas A, B in $\mathcal{L}$.

   (a) $(\forall x \forall y A(x, y) \Rightarrow \forall y \forall x A(x, y))$.

   (b) $(\exists x \exists y A(x, y) \Rightarrow \exists y \exists x A(x, y))$.

   (c) $(\forall x (A(x) \Rightarrow B(x)) \Rightarrow (\forall x A(x) \Rightarrow \forall x B(x)))$.

6. Prove that the following formulas are not predicate tautologies by finding their proper instances and constructing counter models for them.

(a) $((\exists x A(x) \cap \exists x B(x)) \Rightarrow \exists x\ (A(x) \cap B(x)))$.

(b) $(\forall x\ (A(x) \cup B(x)) \Rightarrow (\forall x A(x) \cup \forall x B(x)))$.

(c) $((\forall x A(x) \Rightarrow \forall x B(x)) \Rightarrow \forall x (A(x) \Rightarrow B(x)))$.

7. Prove that the following formulas are predicate tautologies for any formulas $A(x), B(x), A, B$ of $\mathcal{L}$, such that A , B does not contain any free occurrences of x.

   (a) $(\forall x (A \Rightarrow B(x)) \Rightarrow (A \Rightarrow \forall x\ B(x)))$,

   (b) $(\exists x (A(x) \Rightarrow B) \Rightarrow (\forall x A(x) \Rightarrow B))$.

   (c) $(\forall x (A(x) \Rightarrow B) \Rightarrow (\exists x A(x) \Rightarrow B))$.

8. Prove that the restrictions: "A , B does not contain any free occurrences of x" are essential for all of the following tautologies, i.e. give examples of formulas for which the laws without these restrictions fail and construct counter models for them.

   (a) $(\forall x (A(x) \Rightarrow B) \Rightarrow (\exists x A(x) \Rightarrow B))$.

   (b) $(\exists x (A(x) \Rightarrow B) \Rightarrow (\forall x A(x) \Rightarrow B))$.

9. Prove that the *converse implication* to the formulas listed below are predicate tautologies for any formulas $A(x), B(x), A, B$ of $\mathcal{L}$, such that A, B does not contain any free occurrences of x.

   (a) $(\forall x (A \Rightarrow B(x)) \Rightarrow (A \Rightarrow \forall x\ B(x)))$,

   (b) $(\exists x (A(x) \Rightarrow B) \Rightarrow (\forall x A(x) \Rightarrow B))$.

   (c) $(\forall x (A(x) \Rightarrow B) \Rightarrow (\exists x A(x) \Rightarrow B))$.