LOGICS FOR COMPUTER SCIENCE: Classical and Non-Classical

Anita Wasilewska
Chapter 8
Classical Predicate Semantics and Proof Systems

CHAPTER 8 SLIDES

Slides Set 2
Classical Semantics

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*.
Classical Semantics

We define an interpretation $I$ by interpreting predicate and functional symbols as a concrete relation and function defined in a certain set $U \neq \emptyset$.

Constants symbols are interpreted as elements of the set $U$.

The set $U$ is called the universe of the interpretation $I$.

These two items specify a structure

$$M = (U, I)$$

for the language $\mathcal{L}_{CON}(P, F, C)$.
The **semantics** for a **first order** (predicate) language \( \mathcal{L} \) in general, and for the first order **classical logic** in particular, is **defined**, after Tarski (1936), in terms of

the **structure** \( M = [U, I] \)

an **assignment** \( s \) of \( \mathcal{L} \)

a **satisfaction relation** \( (M, s) \models A \) between structures, assignments and formulas of \( \mathcal{L} \)

The definition of the structure \( M = [U, I] \) 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

Definition
Given a predicate language

\[ \mathcal{L} = \mathcal{L}_{\text{CON}}(P, F, C) \]

A structure for \( \mathcal{L} \) is a pair

\[ M = [U, I] \]

where \( U \) is a non empty set called a universe
\( I \) is an assignment called an interpretation of the language \( \mathcal{L}(P, F, C) \) in the universe \( U \)

The structure \( M = [U, I] \) components are defined as follows
Structure Definition

Structure $M = [U, I]$ Components

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

   $P_I \subseteq U^n$

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

   $f_I : U^n \rightarrow U$

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

   $c_I \in U$
Structure Example

Example
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 \( #R = 2, \ #f = 1, \ #g = 2, \text{ and } \ c, d \in C \)
We define a structure \( M = [U, I] \) 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

\[
R_I : \leq
\]
Structure Example

We interpret $f$ as a function $f_I : \{1, 3, 5, 6\} \rightarrow \{1, 3, 5, 6\}$ such that

$$f_I(x) = 5 \quad \text{for all} \quad x \in \{1, 3, 5, 6\}$$

We put $g_I : \{1, 3, 5, 6\} \times \{1, 3, 5, 6\} \rightarrow \{1, 3, 5, 6\}$ such that

$$g_I(x, y) = 1 \quad \text{for all} \quad x \in \{1, 3, 5, 6\}$$

The constant $c$ becomes $c_I = 3$, and $d_I = 6$

We write the structure $M$ as

$$M = \left[ \{1, 3, 5, 6\} \leq, \ f_I, \ g_I, \ c_I = 3, \ d_I = 6 \right]$$
Assignment - Interpretation of Variables

Definition

Given a first order language

\[ \mathcal{L} = \mathcal{L}(P, F, C) \]

with the set \( \text{VAR} \) of variables

Let \( M = [U, I] \) be a structure for \( \mathcal{L} \) with the universe \( U \neq \emptyset \)

An assignment of \( \mathcal{L} \) in \( M = [U, I] \) is any function

\[ s : \text{VAR} \rightarrow U \]

The assignment \( s \) is also called an interpretation of variables \( \text{VAR} \) of \( \mathcal{L} \) in the structure \( M = [U, I] \)
Assignment - Interpretation

Let \( M = [U, I] \) be a structure for \( \mathcal{L} \) and

\[
   s : \text{VAR} \rightarrow U
\]

be an assignment of variables VAR of \( \mathcal{L} \) in the structure \( M \).

Let \( T \) be the set of all terms of \( \mathcal{L} \).

By definition of terms,

\[
   \text{VAR} \subseteq T
\]

We use the interpretation \( I \) of the structure \( M = [U, I] \) to extend the assignment \( s \) to the set the set \( T \) of all terms of the language \( \mathcal{L} \).
Interpretation of Terms

Notation
We denote the extension of the assignment $s$ of the set the set $T$ by $s_I$ rather then by $s^*$ as we did before

$s_I$ associates with each term $t \in T$ an element $s_I(t) \in U$ of the universe of the structure $M = [U, I]$

We define the extension $s_I$ of $s$ by the induction of the length of the term $t \in T$ and call it an interpretation of terms of $L$ in a structure $M = [U, I]$
Interpretation of Terms

Definition
Given a language $L = L(P, F, C)$ and a structure $M = [U, I]$
Let a function
$$s : \text{VAR} \rightarrow U$$
be any assignment of variables $\text{VAR}$ of $L$ in $M$
We extend $s$ to a function
$$s_I : T \rightarrow U$$
called an interpretation of terms of $L$ in $M$
Interpretation of Terms

We define the function $s_I$ by induction on the complexity of terms as follows

1. For any $v \, x \in \text{VAR}$,
   \[ s_I(x) = s(x) \]

2. for any $c \in \text{C}$,
   \[ s_I(c) = c_I; \]

3. for any $t_1, t_2, \ldots, t_n \in \text{T}, \ n \geq 1, \ f \in \text{F}$, such that $\#f = n$
   \[ s_I(f(t_1, t_2, \ldots, t_n)) = f_I(s_I(t_1), s_I(t_2), \ldots, s_I(t_n)) \]
Example
Consider a language

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

for \( \#P = \#R = 2, \ \#f = 1, \ \#h = 2 \)

Let \( M = [Z, I] \), where \( Z \) is the set on integers and the interpretation \( I \) for elements of \( F \) and \( C \) is as follows

\( f_I : Z \rightarrow Z \) is given by formula \( f(m) = m + 1 \) for all \( m \in Z \)

\( h_I : Z \times Z \rightarrow Z \) is given by formula \( f(m, n) = m + n \) for all \( m, n \in Z \)
Interpretation of Terms Example

Let \( s \) be any assignment \( s : \text{VAR} \rightarrow \mathbb{Z} \) such that \( s(x) = -5, \ s(y) = 2 \) and \( t_1, t_2 \in T \)

Let \( t_1 = h(y, f(f(x))) \) and \( t_2 = h(f(x), h(x, f(y))) \)

We evaluate

\[
s_I(t_1) = s_I(h(y, f(x))) = h_I(s_I(y), f_I(s_I(x))) =
\]

\[
+(2, f_I(-5)) = 2 - 4 = -2
\]

and

\[
s_I(t_2) = s_I(h(f(x), h(x, f(y)))) =
\]

\[
+(f_I(-5), (+(-5, 3)) = -4 + (-5 + 3) = -6
\]
Observation

Given \( t \in T \)
Let \( x_1, x_2, \ldots, x_n \in VAR \) be all variables appearing in \( t \)
We write it as
\[
t(x_1, x_2, \ldots, x_n)
\]

Observation

For any term \( t(x_1, x_2, \ldots, x_n) \in T \), any structure \( M = [U, I] \)
and any assignments \( s, s' \) of \( \mathcal{L} \) in \( M \), the following holds
If \( s(x) = s'(x) \) for all \( x \in \{x_1, x_2, \ldots, x_n\} \), i.e.
if the assignments \( s, s' \) \textbf{agree} on all variables appearing in \( t \),
then
\[
s_I(t) = s'_I(t)
\]
Notation

Thus for any $t \in T$, the function $s_I : T \rightarrow U$ depends on only a finite number of values of $s(x)$ for $x \in VAR$

Notation

Given a structure $M = [U, I]$ and an assignment $s : VAR \rightarrow U$ We write

$$s(\overline{a})$$

to denote any assignment

$$s' : VAR \rightarrow U$$

such that $s, s'$ agree on all variables except on $x$ and such that

$$s'(x) = a$$

for certain $a \in U$
Classical Satisfaction

We introduce now a notion of a satisfaction relation \((M, s) \models A\) that acts between structures, assignments and formulas of \(L\).

It is the satisfaction relation that allows us to distinguish one semantics for a given \(L\) from the other, and consequently one logic from the other.

We define now only a classical satisfaction and the notion of classical predicate tautology.
Classical Satisfaction

Definition
Given a predicate (first order) language $L = L(P, F, C)$
Let $M = [U, I]$ be a structure for $L$ and
$s : VAR \rightarrow U$ be any assignment of $L$ in $M$
Let $A \in F$ be any formula of $L$
We define a satisfaction relation

$$(M, s) \models A$$

that reads: ” the assignment $s$ satisfies the formula $A$ in $M”$$
by induction on the complexity of $A$ as follows
Classical Satisfaction

(i) \( A \) is atomic formula

\((M, s) \models P(t_1, \ldots, t_n) \) if and only if \( (s_i(t_1), \ldots, s_i(t_n)) \in P_i \)

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

\((M, s) \models \neg A \) if and only if \((M, s) \not\models A \)

\((M, s) \models (A \cap B) \) if and only if \((M, s) \models A \) and \((M, s) \models B \)

\((M, s) \models (A \cup B) \) if and only if \((M, s) \models A \) or \((M, s) \models B \) or both

\((M, s) \models (A \Rightarrow B) \) if and only if either \((M, s) \not\models A \) or else \((M, s) \models B \) or both
Classical Satisfaction

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

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

\[(M, s) \models \forall x A \text{ if and only if for all } s' \text{ such that } s, s' \text{ agree on all variables except on } x, \text{ and } (M, s') \models A\]
Classical Satisfaction

Observe that the truth or falsity of \((M, s) \models A\) 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 follows:

(iii)’ \(A(x)\) (with a free variable \(x\)) is not atomic formula and \(A\) begins with one of the quantifiers

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

\[(M, s) \models \forall x A \text{ if and only if for all } s' \text{ such that } s(y) = s'(y) \text{ for all } y \in \text{VAR} - \{x\}, (M, s') \models A(x)\]
Satisfaction Relation Exercise

Exercise
For the structures $M_i$, find assignments $s_i, s'_i$ for $1 \leq i \leq 2$ such that

$$(M_i, s_i) \models Q(x, c), \quad \text{and} \quad (M_i, s'_i) \not\models Q(x, c)$$

where $Q \in \mathcal{P}$, $c \in \mathcal{C}$

The structures $M_i$ are defined as follows (the interpretation $I$ for each of them is specified only for symbols in the atomic formula $Q(x, c)$, and $N$ denotes the set of natural numbers

$M_1 = \{1\}, \quad Q_I :=, \quad c_I : 1$ \quad and \quad $M_2 = \{1, 2\}, \quad Q_I :\leq, \quad c_I : 1$
Satisfaction Relation Exercise

Solution
Given $Q(x,c)$. Consider

$$M_1 = \{1\}, \ Q_I :=, \ c_I : 1$$

Observe that all assignments

$$s : \text{VAR} \longrightarrow \{1\}$$

must be defined by a formula $s(x) = 1$ for all $x \in \text{VAR}$

We evaluate $s_I(x) = 1, \ s_I(c) = c_I = 1$

By definition

$$(M_1, s) \models Q(x, c) \quad \text{if and only if} \quad (s_I(x), s_I(c)) \in Q_I$$

This means that $(1, 1) \models \text{what is true as } 1 = 1$

We have proved

$$(M_1, s) \models Q(x, c) \text{ for all assignments } s : \text{VAR} \longrightarrow \{1\}$$
Satisfaction Relation Exercise

Given \( Q(x,c) \). Consider

\[
M_2 = \{1, 2\}, \; Q_l : \leq, \; c_l : 1
\]

Let \( s : VAR \rightarrow \{1, 2\} \) be any assignment, such that

\[
s(x) = 1
\]

We evaluate \( s_l(x) = 1, \; s_l(c) = 1 \) and verify whether

\((s_l(x), s_l(c)) \in Q_l \) i.e. whether \((1, 1) \in \leq\)

This is true as \(1 \leq 1\)

We have found \( s \) such that

\[
(M_2, s) \models Q(x, c)
\]

In fact, have found uncountably many such assignments \( s \)
Satisfaction Relation Exercise

Given $Q(x, c)$ and the structure

$$M_2 = [\{1, 2\}, Q_I : \leq, c_I : 1]$$

Let now $s'$ be any assignment $s' : \text{VAR} \rightarrow \{1, 2\}$ such that $s'(x) = 2$

We evaluate $s'_I(x) = 1, \quad s'_I(c) = 1$

We 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 $s' \neq s$ such that

$$(M_2, s') \not\models Q(x, c)$$

In fact, have found uncountably many such assignments $s'$
Model Definition

Definition
Given a predicate language $\mathcal{L}$, a formula $A \in \mathcal{F}$, and a structure $M = [U, I]$ for $\mathcal{L}$

$M$ is a model for the formula $A$ if and only if $(M, s) \models A$ for all $s : \text{VAR} \rightarrow U$

We denote it as

$M \models A$

For any set $\Gamma \subseteq \mathcal{F}$ of formulas of $\mathcal{L}$,

$M$ is a model for $\Gamma$ if and only if $M \models A$ for all $A \in \Gamma$

We denote it as

$M \models \Gamma$
Counter Model Definition

Definition

Given a predicate language $L$, a formula $A \in \mathcal{F}$, and a structure $M = [U, I]$ for $L$

\[ M \] is a \textbf{counter model} for the formula $A$ if and only if there is an assignment $s : \text{VAR} \rightarrow U$, such that $(M, s) \not\models A$

We denote it as $M \not\models A$
Counter Model Definition

Definition
For any set \( \Gamma \subseteq \mathcal{F} \) of formulas of \( L \),
\( M \) is a **counter model** for \( \Gamma \) if and only if
there is \( A \in \Gamma \), such that \( M \not\models A \)
We denote it as
\[ M \not\models \Gamma \]
Observe that if a formula $A$ is a sentence then the truth or falsity of statement $(M, s) \models A$ is completely independent of $s$.

Hence if $(M, s) \models A$ for some $s$, it holds for all $s$ and the following holds.

**Fact**

For any formula $A$ of $\mathcal{L}$, if $A$ is a sentence, then if there is an $s$ such that $(M, s) \models A$,

then $M$ is a model for $A$, i.e.

$$M \models A$$
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:

**Definition**

Given $A$ of $\mathcal{L}$

By the closure of $A$ we mean the formula obtained from $A$ by prefixing in universal quantifiers all variables that are free in $A$.

If $A$ does not have free variables, i.e. is a sentence, the closure of $A$ is defined to be $A$ itself.

Obviously, a closure of any formula is always a sentence.
Example

Let $A, B$ be formulas

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

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

Their respective closures are

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

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

Example
Let $Q \in P$, $\#Q = 2$ and $c \in C$
Consider formulas

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

and the structures defined as follows.

$$M_1 = [\{1\}, Q_I :=, c_I : 1] \text{ and } M_2 = [\{1, 2\}, Q_I :\leq, c_I : 1]$$

Directly from definition and above Fact we get that:

1. $M_1 \models Q(x, c), \; M_1 \models \forall x Q(x, c), \; M_1 \models \exists x Q(x, c)$

2. $M_2 \not\models Q(x, c), \; M_2 \not\models \forall x Q(x, c), \; M_2 \models \exists x Q(x, c)$
Model, Counter Model Example

Example
Let $Q \in P$, $\#Q = 2$ and $c \in C$
Consider formulas

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

and the structures defined as follows.

$$M_3 = [N, Q_I : \geq, c_I : 0], \quad \text{and} \quad M_4 = [N, Q_I : \geq, c_I : 1]$$

Directly from definition and above Fact we get that:

3. $M_3 \models Q(x, c), \quad M_3 \models \forall xQ(x, c), \quad M_3 \models \exists xQ(x, c)$

4. $M_4 \not\models Q(x, c), \quad M_4 \not\models \forall xQ(x, c), \quad M_4 \models \exists xQ(x, c)$
True, False in $M$

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

$A$ is true in $M$ and is written as

$$M \models A$$

if and only if all assignments $s$ of $\mathcal{L}$ in $M$ satisfy $A$, i.e. when $M$ is a model for $A$

$A$ is false in $M$ and written as

$$M \not\models A$$

if and only if there is no assignment $s$ of $\mathcal{L}$ in $M$ that satisfies $A$
True, False in $M$

Here are some properties of the notions:

1. "A is true in $M"$ written symbolically as

   \[ M \models A \]

2. "A is false in $M"$ written symbolically as

   \[ M \not\models A \]

They are obvious under intuitive understanding of the notion of satisfaction.
Their formal proofs are left as an exercise.
True, False in $M$ Properties

Properties
Given a structure $M = [U, I]$ and any formulas formula $A, B$ of $L$. The following properties hold

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

$$M \models A \text{ if and only if } M \models \neg A$$

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

$$M \models A \text{ if and only if } M \models \neg A$$

P3. It is not the case that both $M \models A$ and $M \models \neg A$, i.e. there is no formula $A$, such that

$$M \models A \text{ and } M \models \neg A$$
True, False in $\mathbf{M}$ Properties

Properties

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 $\mathbf{M} \models \neg B$

$\mathbf{M} \models (A \Rightarrow B)$ if and only if $\mathbf{M} \models A$ and $\mathbf{M} \models \neg B$

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

P7. A formula $A$ is true in $\mathbf{M}$ if and only if its closure is true in $\mathbf{M}$
Definition

A formula $A$ of $\mathcal{L}$ is a predicate tautology (is valid) if and only if $M \models A$ for all structures $M = [U, I]$.

We also say

A formula $A$ of $\mathcal{L}$ is a predicate tautology (is valid) if and only if $A$ is true in all structures $M$ 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).
Valid, Tautology Definition

We write

\[ \models_p A \]

when there is a **need** to stress a **distinction** between **propositional** and **predicate** tautologies

otherwise we write

\[ \models A \]

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

Following the notation \( T \) we have established for the set of all **propositional** tautologies we denote by \( T_p \) the set of all **predicate** tautologies

We put

\[ T_p = \{ A \text{ of } \mathcal{L} : \models_p A \} \]
Not a Tautology, Counter Model

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

$A$ is not a predicate tautology and denote it by

$$\not\models A$$

if and only if there is a structure $M = [U, I]$ for $\mathcal{L}$, such that

$$M \not\models A$$

We call such structure $M$ a counter-model for $A$
Counter Model

In order to prove that a formula $A$ is not a tautology one has to find a counter-model for $A$

It means one has to define the components of a structure $M = [U, I]$ for $L$, i.e.

a non-empty set $U$, called universe and an interpretation $I$ of $L$ in the universe $U$

Moreover, one has to define an assignment $s : VAR \rightarrow U$ and prove that that

$$(M, s) \notsat A$$
Contradictions

We introduce now a notion of predicate contradiction.

Definition
For any formula $A$ of $\mathcal{L}$, $A$ is a predicate contradiction if and only if $A$ is false in all structures $M$.

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

$$|=\models A \text{ if and only if } M|=\models A, \text{ for all structures } M$$

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

$$|=\models^p A$$
Contradictions

Following the notation $C$ for the set of all propositional contradictions we denote by $C_p$ the set of all predicate contradictions, i.e.

$$C_p = \{ A \text{ of } \mathcal{L}(P, F, C) : \models_p A \}$$

Directly from the contradiction definition we have the following duality property characteristic for classical logic

Fact

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

$$A \in T_p \text{ if and only if } \neg A \in C_p$$

$$A \in C_p \text{ if and only if } \neg A \in T_p$$
Proving Predicate TAutologies

We prove, as an example the following basic predicate tautology

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

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

Proof
Assume that $\not\models (\forall x A(x) \Rightarrow \exists x A(x))$
It means that there is a structure

$$M = [U, I] \text{ and } s : \text{VAR} \rightarrow U$$

such that

$$(M, s) \not\models (\forall x A(x) \Rightarrow \exists x A(x))$$
Proving Predicate Tautologies

Observe that \((M, s) \not\models (\forall x A(x) \Rightarrow \exists x A(x))\) is equivalent to

\[(M, s) \not\models \forall x A(x) \text{ and } (M, s) \not\models \exists x A(x)\]

By definition, \((M, s) \not\models \forall x A(x)\) means that \((M, s') \models A(x)\) for all \(s'\) such that \(s, s'\) agree on all variables except on \(x\)

At the same time \((M, s) \not\models \exists x A(x)\) means that it is not true that there is \(s'\) such that \(s, s'\) agree on all variables except on \(x\), and \((M, s') \models A(x)\). This contradiction proves

\[\models (\forall x A(x) \Rightarrow \exists x A(x))\]
Disapproving Predicate Tautologies

We show now, as an example of a **counter model** construction that the converse implication to

\[ \models (\forall x \ A(x) \Rightarrow \exists x \ A(x)) \]

is not a predicate tautology i.e. the following holds

**Fact**
There is a formula \( A \) of \( \mathcal{L} \), such that

\[ \not\models (\exists x \ A(x) \Rightarrow \forall x \ A(x)) \]

**Proof**
Observe that to prove the **Fact** we have to provide an example of an **instance** of a formula \( A(x) \) and construct a **counter model** \( M = [U, I] \) for it
Proving Predicate Tautologies

Let $A(x)$ be an atomic formula

$$P(x, c) \quad \text{for any} \quad P \in P, \quad \#P = 2$$

The needed instance is a formula

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

We take as its counter model a structure

$$M = \langle N, P_I :<, c_I : 3 \rangle$$

where $N$ is set of natural numbers. We want to show

$$M \nvDash (\exists x \ P(x, c) \Rightarrow \forall x \ P(x, c))$$

It means we have to define an assignment $s$ such that $s: VAR \rightarrow N$ and

$$(M, s) \nvDash (\exists x \ P(x, c) \Rightarrow \forall x \ P(x, c))$$
Proving Predicate Tautologies

Let \( s \) be any assignment \( s : \text{VAR} \rightarrow N \)

We show now

\[
(M, s) \models \exists x \ P(x, c)
\]

Take any \( s' \) such that

\[
s'(x) = 2 \quad \text{and} \quad s'(y) = s(y) \quad \text{for all} \quad y \in \text{VAR} - \{x\}
\]

We have \( (2, 3) \in P_I \), as \( 2 < 3 \)

Hence we proved that there exists \( s' \) that agrees with \( s \) on all variables except on \( x \) and

\[
(M, s') \models P(x, c)
\]
Proving Predicate Tautologies

But at the same time

\[(M, s) \not\models \forall x P(x, c)\]

as for example for \(s'\) such that

\[s'(x) = 5 \quad \text{and} \quad s'(y) = s(y) \quad \text{for all} \quad y \in \text{VAR} - \{x\}\]

We have that \((2, 3) \notin P_I, \quad \text{as} \quad 5 \nless 3\)

This proves that the structure

\[M = [N, P_I : <, \ c_I : 3 ]\]

is a **counter model** for \(\forall x P(x, c)\)

Hence we proved that

\[\not\models (\exists x A(x) \Rightarrow \forall x A(x))\]
Proving Predicate Tautologies

Short Hand Solution of

\[ \neg (\exists x \, P(x, c) \Rightarrow \forall x \, P(x, c)) \]

We take as its **counter model** a structure

\[ M = [ N, \, P_I : \prec, \, c_I : 3 ] \]

where \( N \) is set of natural numbers

The formula

\[ (\exists x \, P(x, c) \Rightarrow \forall x \, P(x, c)) \]

becomes in \( M = (N, P_I : \prec, \, c_I : 3) \) 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 than 3.