Predicate Logic

Aritra Hazra

Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur,
Paschim Medinipur, West Bengal, India - 721302.

Email: *aritrah@cse.iitkgp.ac.in*

Autumn 2020

From Propositional Logic to Predicate Logic

Example

1. Wherever Ankush goes, so does the pet dog. Ankush goes to school. So, the dog goes to school.
2. No contractors are dependable. Some engineers are contractors. Therefore, some engineers are not dependable.
3. All actresses are graceful. Anushka is a dancer. Anushka is an actress. Therefore, some dancers are graceful.
4. Every passenger either travels in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore, some passengers travel in second class.

From Propositional Logic to Predicate Logic

Example

1. Wherever Ankush goes, so does the pet dog. Ankush goes to school. So, the dog goes to school.
2. No contractors are dependable. Some engineers are contractors. Therefore, some engineers are not dependable.
3. All actresses are graceful. Anushka is a dancer. Anushka is an actress. Therefore, some dancers are graceful.
4. Every passenger either travels in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore, some passengers travel in second class.

Propositional Logic Insufficiency

- **Quantifications:** ‘some’, ‘none’, ‘all’, ‘every’, ‘wherever’ etc.
- **Associations:** ‘x goes to some place y’, ‘z travels in first class’ etc.

Predicate Logic Argument Formulation

Example

Wherever Ankush goes, so does the pet dog. Ankush goes to school. So, the dog goes to school.

Predicate Logic Argument Formulation

Example

Wherever Ankush goes, so does the pet dog. Ankush goes to school. So, the dog goes to school.

Formal Constructs and Fundamentals

Following are the representational extensions made in First-Order Logic (Predicate Logic) over Propositional Logic constructs:

Predicate Logic Argument Formulation

Example

Wherever Ankush goes, so does the pet dog. Ankush goes to school. So, the dog goes to school.

Formal Constructs and Fundamentals

Following are the representational extensions made in First-Order Logic (Predicate Logic) over Propositional Logic constructs:

New Additions: Variables (for e.g., `x`, `y`) and Constants (for e.g., `Ankush`, `Dog`)

Predicate Logic Argument Formulation

Example

Wherever Ankush goes, so does the pet dog. Ankush goes to school. So, the dog goes to school.

Formal Constructs and Fundamentals

Following are the representational extensions made in First-Order Logic (Predicate Logic) over Propositional Logic constructs:

New Additions: Variables (for e.g., `x`, `y`) and Constants (for e.g., `Ankush`, `Dog`)

Functional Symbols: Functional constructs returning Non-Boolean values (for e.g., `Age(x)` indicates 'the age of `x`')

Predicate Logic Argument Formulation

Example

Wherever Ankush goes, so does the pet dog. Ankush goes to school. So, the dog goes to school.

Formal Constructs and Fundamentals

Following are the representational extensions made in First-Order Logic (Predicate Logic) over Propositional Logic constructs:

New Additions: Variables (for e.g., $\mathtt{x}$, $\mathtt{y}$) and Constants (for e.g., $\mathtt{Ankush}$, $\mathtt{Dog}$)

Functional Symbols: Functional constructs returning Non-Boolean values (for e.g., $\mathtt{Age(x)}$ indicates 'the age of $\mathtt{x}$')

Predicate Symbols: Constructs indicating associations having Boolean outcomes (for e.g., $\mathtt{goes(x, y)}$ indicates '$\mathtt{x}$ goes to the place $\mathtt{y}$')

Predicate Logic Argument Formulation

Example

Wherever Ankush goes, so does the pet dog. Ankush goes to school. So, the dog goes to school.

Formal Constructs and Fundamentals

Following are the representational extensions made in First-Order Logic (Predicate Logic) over Propositional Logic constructs:

New Additions: Variables (for e.g., $\mathtt{x}$, $\mathtt{y}$) and Constants (for e.g., $\mathtt{Ankush}$, $\mathtt{Dog}$)

Functional Symbols: Functional constructs returning Non-Boolean values (for e.g., $\mathtt{Age}(\mathtt{x})$ indicates 'the age of $\mathtt{x}$')

Predicate Symbols: Constructs indicating associations having Boolean outcomes (for e.g., $\mathtt{goes}(\mathtt{x}, \mathtt{y})$ indicates '$\mathtt{x}$ goes to the place $\mathtt{y}$')

Connectors: Well-defined connectors, such as, $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (implication), $\leftrightarrow$ (if and only if) etc.

Predicate Logic Argument Formulation

Example

Wherever Ankush goes, so does the pet dog. Ankush goes to school. So, the dog goes to school.

Formal Constructs and Fundamentals

Following are the representational extensions made in First-Order Logic (Predicate Logic) over Propositional Logic constructs:

New Additions: Variables (for e.g., $\mathtt{x}$, $\mathtt{y}$) and Constants (for e.g., Ankush, Dog)

Functional Symbols: Functional constructs returning Non-Boolean values (for e.g., $\mathtt{Age(x)}$ indicates 'the age of $\mathtt{x}$')

Predicate Symbols: Constructs indicating associations having Boolean outcomes (for e.g., $\mathtt{goes(x, y)}$ indicates '$\mathtt{x}$ goes to the place $\mathtt{y}$')

Connectors: Well-defined connectors, such as, $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (implication), $\leftrightarrow$ (if and only if) etc.

Quantifiers: Existantial ($\exists$, i.e. there exists) and Universal ($\forall$, i.e. for all)

Predicate Logic Argument Formulation: *Example-1*

Example

Wherever Ankush goes, so does the pet dog. Ankush goes to school. So, the dog goes to school.

Predicate Logic Argument Formulation: *Example-1*

Example

Wherever Ankush goes, so does the pet dog. Ankush goes to school. So, the dog goes to school.

Logical Formulation

Variables: x and y

Constants: Ankush, Dog and School

Predicate: goes(x, y): x goes to y

Predicate Logic Argument Formulation: *Example-1*

Example

Wherever Ankush goes, so does the pet dog. Ankush goes to school. So, the dog goes to school.

Logical Formulation

Variables: $\texttt{x}$ and $\texttt{y}$

Constants: $\texttt{Ankush}$, $\texttt{Dog}$ and $\texttt{School}$

Predicate: $\texttt{goes}(\texttt{x}, \texttt{y})$: $\texttt{x}$ goes to $\texttt{y}$

Formula:

$$\begin{aligned} F_1 &: \forall x\, (\texttt{goes}(\texttt{Ankush}, x) \rightarrow \texttt{goes}(\texttt{Dog}, x)) \\ F_2 &: \texttt{goes}(\texttt{Ankush}, \texttt{School}) \\ G &: \texttt{goes}(\texttt{Dog}, \texttt{School}) \end{aligned}$$

Requirement: To prove whether $(F_1 \wedge F_2) \rightarrow G$ is valid

Predicate Logic Argument Formulation: *Example-2*

Example

No contractors are dependable. Some engineers are contractors. Therefore, some engineers are not dependable.

Predicate Logic Argument Formulation: *Example-2*

Example

No contractors are dependable. Some engineers are contractors. Therefore, some engineers are not dependable.

Logical Formulation

Predicates: Assuming the variable as x.

$$\begin{aligned} \texttt{contractor(x)} &: \text{ x is a contractor} \\ \texttt{dependable(x)} &: \text{ x is dependable} \\ \texttt{engineer(x)} &: \text{ x is an engineer} \end{aligned}$$

Predicate Logic Argument Formulation: *Example-2*

Example

No contractors are dependable. Some engineers are contractors. Therefore, some engineers are not dependable.

Logical Formulation

Predicates: Assuming the variable as $\mathtt{x}$.

$$\begin{aligned} \mathtt{contractor(x)} &: \text{ x is a contractor} \\ \mathtt{dependable(x)} &: \text{ x is dependable} \\ \mathtt{engineer(x)} &: \text{ x is an engineer} \end{aligned}$$

Formula:

$$\begin{aligned} F_1 &: \forall x\, (\mathtt{contractor}(x) \rightarrow \neg \mathtt{dependable}(x)) \\ (\text{Alt.}) &: \neg \exists x\, (\mathtt{contractor}(x) \wedge \mathtt{dependable}(x)) \\ F_2 &: \exists x\, (\mathtt{engineer}(x) \wedge \mathtt{contractor}(x)) \\ (\text{Alt.}) &: \exists x\, (\mathtt{engineer}(x) \rightarrow \mathtt{contractor}(x)) \wedge \exists x\, \mathtt{engineer}(x) \\ G &: \exists x\, (\mathtt{engineer}(x) \wedge \neg \mathtt{dependable}(x)) \end{aligned}$$

Requirement: To prove whether $(F_1 \wedge F_2) \rightarrow G$ is valid

Predicate Logic Argument Formulation: *Example-3*

Example

All actresses are graceful. Anushka is a dancer. Anushka is an actress. Therefore, some dancers are graceful.

Predicate Logic Argument Formulation: *Example-3*

Example

All actresses are graceful. Anushka is a dancer. Anushka is an actress. Therefore, some dancers are graceful.

Logical Formulation

Predicates: Assuming the variable as $\mathtt{x}$.

$$\begin{aligned} \mathtt{actress(x)} &: \quad \mathtt{x} \text{ is an actress} \\ \mathtt{graceful(x)} &: \quad \mathtt{x} \text{ is graceful} \\ \mathtt{dancer(x)} &: \quad \mathtt{x} \text{ is a dancer} \end{aligned}$$

Predicate Logic Argument Formulation: *Example-3*

Example

All actresses are graceful. Anushka is a dancer. Anushka is an actress. Therefore, some dancers are graceful.

Logical Formulation

Predicates: Assuming the variable as $\mathtt{x}$.

$$\begin{aligned} \mathtt{actress(x)} &: \text{ x is an actress} \\ \mathtt{graceful(x)} &: \text{ x is graceful} \\ \mathtt{dancer(x)} &: \text{ x is a dancer} \end{aligned}$$

Formula:

$$\begin{aligned} \mathtt{F}_1 &: \forall \mathtt{x}\ (\mathtt{actress(x)} \rightarrow \mathtt{graceful(x)}) \\ \mathtt{F}_2 &: \mathtt{dacncer(Anushka)} \\ \mathtt{F}_3 &: \mathtt{actress(Anushka)} \\ \mathtt{G} &: \exists \mathtt{x}\ (\mathtt{dancer(x)} \wedge \mathtt{graceful(x)}) \end{aligned}$$

Requirement: To prove whether $(\mathtt{F}_1 \wedge \mathtt{F}_2 \wedge \mathtt{F}_3) \rightarrow \mathtt{G}$ is valid

Predicate Logic Argument Formulation: *Example-4*

Example

Every passenger either travels in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore, some passengers travel in second class.

Predicate Logic Argument Formulation: *Example-4*

Example

Every passenger either travels in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore, some passengers travel in second class.

Logical Formulation

Predicates: Assuming the variable as x.

$$
\begin{aligned}
\texttt{pass(x)}: &\quad \texttt{x} \text{ is a passenger} \\
\texttt{frst(x)}: &\quad \texttt{x} \text{ travels in first class} \\
\texttt{scnd(x)}: &\quad \texttt{x} \text{ travels in second class} \\
\texttt{wlty(x)}: &\quad \texttt{x} \text{ is wealthy}
\end{aligned}
$$

Predicate Logic Argument Formulation: *Example-4*

Example

Every passenger either travels in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore, some passengers travel in second class.

Logical Formulation

Predicates: Assuming the variable as x.

pass(x) :	x is a passenger
frst(x) :	x travels in first class
scnd(x) :	x travels in second class
wlty(x) :	x is wealthy

Formula: To prove whether $(F_1 \wedge F_2 \wedge F_3 \wedge F_4) \rightarrow G$ is valid.

$F_1 : \forall x\ [pass(x) \rightarrow (frst(x) \vee scnd(x))]$

$F_1 : \forall x\ [pass(x) \rightarrow ((frst(x) \wedge \neg scnd(x)) \vee (\neg frst(x) \wedge scnd(x)))]$

$F_2 : \forall x\ [pass(x) \rightarrow ((scnd(x) \rightarrow \neg wlty(x)) \wedge (\neg wlty(x) \rightarrow scnd(x)))]$

$F_3 : \exists x\ [pass(x) \wedge wlty(x)]$

$F_4 : \neg \forall x\ [pass(x) \rightarrow wlty(x)]$

(Alt.) $\exists x\ [pass(x) \wedge \neg wlty(x)]$

$G : \exists x\ [pass(x) \wedge scnd(x)]$

Predicate Logic Constructs: Use of Quantifiers

Example

- A. Everyone likes everyone. $\forall x \, \forall y \; \texttt{likes}(x, y)$
- B. Someone likes someone. $\exists x \, \exists y \; \texttt{likes}(x, y)$
- C. Everyone likes someone. $\forall x \; \left(\exists y \; \texttt{likes}(x, y)\right)$
- D. Someone likes everyone. $\exists x \; \left(\forall y \; \texttt{likes}(x, y)\right)$

Predicate Logic Constructs: Use of Quantifiers

Example

- A. Everyone likes everyone. $\forall x\ \forall y\ \texttt{likes}(x, y)$
- B. Someone likes someone. $\exists x\ \exists y\ \texttt{likes}(x, y)$
- C. Everyone likes someone. $\forall x\ (\exists y\ \texttt{likes}(x, y))$
- D. Someone likes everyone. $\exists x\ (\forall y\ \texttt{likes}(x, y))$

Example

- I. Everyone is liked by everyone. $\forall y\ (\forall x\ \texttt{likes}(x, y))$
- II. Someone is liked by someone. $\exists y\ (\exists x\ \texttt{likes}(x, y))$
- III. Someone is liked by everyone. $\exists y\ (\forall x\ \texttt{likes}(x, y))$
- IV. Everyone is liked by someone. $\forall y\ (\exists x\ \texttt{likes}(x, y))$

Predicate Logic Constructs: Use of Quantifiers

Example

A. Everyone likes everyone. $\forall x\ \forall y\ \texttt{likes}(x, y)$
B. Someone likes someone. $\exists x\ \exists y\ \texttt{likes}(x, y)$
C. Everyone likes someone. $\forall x\ (\exists y\ \texttt{likes}(x, y))$
D. Someone likes everyone. $\exists x\ (\forall y\ \texttt{likes}(x, y))$

Example

I. Everyone is liked by everyone. $\forall y\ (\forall x\ \texttt{likes}(x, y))$
II. Someone is liked by someone. $\exists y\ (\exists x\ \texttt{likes}(x, y))$
III. Someone is liked by everyone. $\exists y\ (\forall x\ \texttt{likes}(x, y))$
IV. Everyone is liked by someone. $\forall y\ (\exists x\ \texttt{likes}(x, y))$

Note: Active and Passive Voice statements in English are NOT logically similar!

Predicate Logic Constructs: Use of Quantifiers

Example

1. If everyone likes everyone, then someone likes everyone.
   $\big(\forall \mathtt{x}\ (\forall \mathtt{y}\ \mathtt{likes}(\mathtt{x},\mathtt{y}))\big) \rightarrow \big(\exists \mathtt{x}\ (\forall \mathtt{y}\ \mathtt{likes}(\mathtt{x},\mathtt{y}))\big)$
2. If some person is liked by everyone, then that person likes himself/herself.
   $\exists \mathtt{y}\ \big((\forall \mathtt{x}\ \mathtt{likes}(\mathtt{x},\mathtt{y})) \rightarrow \mathtt{likes}(\mathtt{y},\mathtt{y})\big)$

Predicate Logic Constructs: Use of Quantifiers

Example

1. If everyone likes everyone, then someone likes everyone.
   $(\forall x\ (\forall y\ \texttt{likes}(x, y))) \rightarrow (\exists x\ (\forall y\ \texttt{likes}(x, y)))$
2. If some person is liked by everyone, then that person likes himself/herself.
   $\exists y\ ((\forall x\ \texttt{likes}(x, y)) \rightarrow \texttt{likes}(y, y))$

Some Notions over Quantifiers

Contrapositive of $\forall x\ (p(x) \rightarrow q(x))$: $\forall x\ (\neg q(x) \rightarrow \neg p(x))$

Converse of $\forall x\ (p(x) \rightarrow q(x))$: $\forall x\ (q(x) \rightarrow p(x))$

Inverse of $\forall x\ (p(x) \rightarrow q(x))$: $\forall x\ (\neg p(x) \rightarrow \neg q(x))$

Predicate Logic Constructs: Use of Quantifiers

Example

1. If everyone likes everyone, then someone likes everyone.
   $(\forall x\ (\forall y\ \texttt{likes}(x,y))) \rightarrow (\exists x\ (\forall y\ \texttt{likes}(x,y)))$
2. If some person is liked by everyone, then that person likes himself/herself.
   $\exists y\ ((\forall x\ \texttt{likes}(x,y)) \rightarrow \texttt{likes}(y,y))$

Some Notions over Quantifiers

Contrapositive of $\forall x\ (p(x) \rightarrow q(x))$: $\forall x\ (\neg q(x) \rightarrow \neg p(x))$

Converse of $\forall x\ (p(x) \rightarrow q(x))$: $\forall x\ (q(x) \rightarrow p(x))$

Inverse of $\forall x\ (p(x) \rightarrow q(x))$: $\forall x\ (\neg p(x) \rightarrow \neg q(x))$

Negation Law : (DeMorgan's Principle)

- $\neg\forall x\ p(x) \equiv \exists x\ \neg p(x)$ [also written as, $\neg\forall x\ p(x) \Leftrightarrow \exists x\ \neg p(x)$]
- $\neg\exists x\ p(x) \equiv \forall x\ \neg p(x)$ [also written as, $\neg\exists x\ p(x) \Leftrightarrow \forall x\ \neg p(x)$]

(Intuitively, $\forall x$ indicates $\wedge_{i=0}^{\infty} x_i$ and $\exists x$ indicates $\vee_{i=0}^{\infty} x_i$)

Predicate Logic Constructs: Use of Function Symbols

Example

1. If $\mathtt{x}$ is greater than $\mathtt{y}$ and $\mathtt{y}$ is greater than $\mathtt{z}$, then $\mathtt{x}$ is greater than $\mathtt{z}$.
   Predicate: $\mathtt{gt}(\mathtt{x},\mathtt{y})$ denotes '$\mathtt{x}$ is greater than $\mathtt{y}$'
   Formula: $\forall \mathtt{x}\ \forall \mathtt{y}\ \forall \mathtt{z}\ \big(\mathtt{gt}(\mathtt{x},\mathtt{y}) \wedge \mathtt{gt}(\mathtt{y},\mathtt{z}) \rightarrow \mathtt{gt}(\mathtt{x},\mathtt{z})\big)$

Predicate Logic Constructs: Use of Function Symbols

Example

1. If $\mathtt{x}$ is greater than $\mathtt{y}$ and $\mathtt{y}$ is greater than $\mathtt{z}$, then $\mathtt{x}$ is greater than $\mathtt{z}$.
   Predicate: $\mathtt{gt}(\mathtt{x},\mathtt{y})$ denotes '$\mathtt{x}$ is greater than $\mathtt{y}$'
   Formula: $\forall \mathtt{x}\ \forall \mathtt{y}\ \forall \mathtt{z}\ \big(\mathtt{gt}(\mathtt{x},\mathtt{y}) \land \mathtt{gt}(\mathtt{y},\mathtt{z}) \rightarrow \mathtt{gt}(\mathtt{x},\mathtt{z})\big)$
2. The age of a person is greater than the age of his/her child.
   Function Symbol: $\mathtt{Age}(\mathtt{x})$ denotes 'age of the person $\mathtt{x}$'
   Predicate: $\mathtt{child}(\mathtt{x},\mathtt{y})$ denotes '$\mathtt{x}$ is a child of $\mathtt{y}$'
   Formula: $\forall \mathtt{x}\ \forall \mathtt{y}\ \big(\mathtt{child}(\mathtt{x},\mathtt{y}) \rightarrow \mathtt{gt}(\mathtt{Age}(\mathtt{y}),\mathtt{Age}(\mathtt{x}))\big)$

Predicate Logic Constructs: Use of Function Symbols

Example

1. If $\mathtt{x}$ is greater than $\mathtt{y}$ and $\mathtt{y}$ is greater than $\mathtt{z}$, then $\mathtt{x}$ is greater than $\mathtt{z}$.
   Predicate: $\mathtt{gt}(\mathtt{x},\mathtt{y})$ denotes '$\mathtt{x}$ is greater than $\mathtt{y}$'
   Formula: $\forall \mathtt{x}\ \forall \mathtt{y}\ \forall \mathtt{z}\ \big(\mathtt{gt}(\mathtt{x},\mathtt{y}) \wedge \mathtt{gt}(\mathtt{y},\mathtt{z}) \rightarrow \mathtt{gt}(\mathtt{x},\mathtt{z})\big)$
2. The age of a person is greater than the age of his/her child.
   Function Symbol: $\mathtt{Age}(\mathtt{x})$ denotes 'age of the person $\mathtt{x}$'
   Predicate: $\mathtt{child}(\mathtt{x},\mathtt{y})$ denotes '$\mathtt{x}$ is a child of $\mathtt{y}$'
   Formula: $\forall \mathtt{x}\ \forall \mathtt{y}\ \big(\mathtt{child}(\mathtt{x},\mathtt{y}) \rightarrow \mathtt{gt}(\mathtt{Age}(\mathtt{y}),\mathtt{Age}(\mathtt{x}))\big)$
3. The age of a person is greater than the age of his/her grandchild.
   Formula: $\forall \mathtt{x}\ \forall \mathtt{y}\ \forall \mathtt{z}\ \big((\mathtt{child}(\mathtt{x},\mathtt{y}) \wedge \mathtt{child}(\mathtt{y},\mathtt{z})) \rightarrow \mathtt{gt}(\mathtt{Age}(\mathtt{z}),\mathtt{Age}(\mathtt{x}))\big)$

Predicate Logic Constructs: Use of Function Symbols

Example

1. If x is greater than y and y is greater than z, then x is greater than z.
   Predicate: $\mathtt{gt}(\mathtt{x},\mathtt{y})$ denotes 'x is greater than y'
   Formula: $\forall \mathtt{x}\ \forall \mathtt{y}\ \forall \mathtt{z}\ \big(\mathtt{gt}(\mathtt{x},\mathtt{y}) \wedge \mathtt{gt}(\mathtt{y},\mathtt{z}) \rightarrow \mathtt{gt}(\mathtt{x},\mathtt{z})\big)$
2. The age of a person is greater than the age of his/her child.
   Function Symbol: $\mathtt{Age}(\mathtt{x})$ denotes 'age of the person x'
   Predicate: $\mathtt{child}(\mathtt{x},\mathtt{y})$ denotes 'x is a child of y'
   Formula: $\forall \mathtt{x}\ \forall \mathtt{y}\ \big(\mathtt{child}(\mathtt{x},\mathtt{y}) \rightarrow \mathtt{gt}(\mathtt{Age}(\mathtt{y}),\mathtt{Age}(\mathtt{x}))\big)$
3. The age of a person is greater than the age of his/her grandchild.
   Formula: $\forall \mathtt{x}\ \forall \mathtt{y}\ \forall \mathtt{z}\ \big((\mathtt{child}(\mathtt{x},\mathtt{y}) \wedge \mathtt{child}(\mathtt{y},\mathtt{z})) \rightarrow \mathtt{gt}(\mathtt{Age}(\mathtt{z}),\mathtt{Age}(\mathtt{x}))\big)$
4. The sum of ages of two children are never more than or equal to the sum of ages of their parents.
   Function Symbol: $\mathtt{sum}(\mathtt{x},\mathtt{y})$ denotes 'sum of x and y, i.e. (x+y)'
   Formula: $\forall \mathtt{w}\ \forall \mathtt{x}\ \forall \mathtt{y}\ \forall \mathtt{z}\ \big((\mathtt{child}(\mathtt{w},\mathtt{y}) \wedge \mathtt{child}(\mathtt{w},\mathtt{z}) \wedge \mathtt{child}(\mathtt{x},\mathtt{y}) \wedge \mathtt{child}(\mathtt{x},\mathtt{z})) \rightarrow (\mathtt{gt}(\mathtt{sum}(\mathtt{Age}(\mathtt{y}),\mathtt{Age}(\mathtt{z})),\mathtt{sum}(\mathtt{Age}(\mathtt{w}),\mathtt{Age}(\mathtt{x})))\big)\big)$

Predicate Logic Constructs: Equivalence and Implications

Definitions

Logical Equivalence: Two predicates, $p(x)$ and $q(x)$ are said to be *logically equivalent* when for each $x = A$ in the universe, $(p(A) \leftrightarrow q(A))$ holds. Formally, we express it as, $\forall x \; (p(x) \Leftrightarrow q(x))$.

Predicate Logic Constructs: Equivalence and Implications

Definitions

Logical Equivalence: Two predicates, $\mathtt{p(x)}$ and $\mathtt{q(x)}$ are said to be *logically equivalent* when for each $\mathtt{x} = \mathtt{A}$ in the universe, $(\mathtt{p(A)} \leftrightarrow \mathtt{q(A)})$ holds. Formally, we express it as, $\forall \mathtt{x}\ (\mathtt{p(x)} \Leftrightarrow \mathtt{q(x)})$.

Logical Implication: A predicate, $\mathtt{p(x)}$ is said to *logically imply* another predicate $\mathtt{q(x)}$ when for each $\mathtt{x} = \mathtt{A}$ in the universe, $(\mathtt{p(A)} \rightarrow \mathtt{q(A)})$ holds. Formally, we express it as, $\forall \mathtt{x}\ (\mathtt{p(x)} \Rightarrow \mathtt{q(x)})$.

Predicate Logic Constructs: Equivalence and Implications

Definitions

Logical Equivalence: Two predicates, $p(x)$ and $q(x)$ are said to be *logically equivalent* when for each $x = A$ in the universe, $(p(A) \leftrightarrow q(A))$ holds. Formally, we express it as, $\forall x \ (p(x) \Leftrightarrow q(x))$.

Logical Implication: A predicate, $p(x)$ is said to *logically imply* another predicate $q(x)$ when for each $x = A$ in the universe, $(p(A) \rightarrow q(A))$ holds. Formally, we express it as, $\forall x \ (p(x) \Rightarrow q(x))$.

Some Logical Rules

- $\exists x \ (p(x) \wedge q(x)) \Rightarrow (\exists x \ p(x) \wedge \exists x \ q(x))$
- $(\exists x \ p(x) \wedge \exists x \ q(x)) \not\Rightarrow \exists x \ (p(x) \wedge q(x))$

Predicate Logic Constructs: Equivalence and Implications

Definitions

Logical Equivalence: Two predicates, $p(x)$ and $q(x)$ are said to be *logically equivalent* when for each $x = A$ in the universe, $(p(A) \leftrightarrow q(A))$ holds. Formally, we express it as, $\forall x \; (p(x) \Leftrightarrow q(x))$.

Logical Implication: A predicate, $p(x)$ is said to *logically imply* another predicate $q(x)$ when for each $x = A$ in the universe, $(p(A) \rightarrow q(A))$ holds. Formally, we express it as, $\forall x \; (p(x) \Rightarrow q(x))$.

Some Logical Rules

- $\exists x \; (p(x) \wedge q(x)) \Rightarrow (\exists x \; p(x) \wedge \exists x \; q(x))$
- $(\exists x \; p(x) \wedge \exists x \; q(x)) \not\Rightarrow \exists x \; (p(x) \wedge q(x))$
- $\exists x \; (p(x) \vee q(x)) \Leftrightarrow (\exists x \; p(x) \vee \exists x \; q(x))$ [distributed property of $\exists$ over $\vee$]
- $\forall x \; (p(x) \wedge q(x)) \Leftrightarrow (\forall x \; p(x) \wedge \forall x \; q(x))$ [distributed property of $\forall$ over $\wedge$]

Predicate Logic Constructs: Equivalence and Implications

Definitions

Logical Equivalence: Two predicates, $p(x)$ and $q(x)$ are said to be *logically equivalent* when for each $x = A$ in the universe, $(p(A) \leftrightarrow q(A))$ holds. Formally, we express it as, $\forall x \; (p(x) \Leftrightarrow q(x))$.

Logical Implication: A predicate, $p(x)$ is said to *logically imply* another predicate $q(x)$ when for each $x = A$ in the universe, $(p(A) \rightarrow q(A))$ holds. Formally, we express it as, $\forall x \; (p(x) \Rightarrow q(x))$.

Some Logical Rules

- $\exists x \; (p(x) \wedge q(x)) \Rightarrow (\exists x \; p(x) \wedge \exists x \; q(x))$
- $(\exists x \; p(x) \wedge \exists x \; q(x)) \not\Rightarrow \exists x \; (p(x) \wedge q(x))$
- $\exists x \; (p(x) \vee q(x)) \Leftrightarrow (\exists x \; p(x) \vee \exists x \; q(x))$ [distributed property of $\exists$ over $\vee$]
- $\forall x \; (p(x) \wedge q(x)) \Leftrightarrow (\forall x \; p(x) \wedge \forall x \; q(x))$ [distributed property of $\forall$ over $\wedge$]
- $(\forall x \; p(x) \vee \forall x \; q(x)) \Rightarrow \forall x \; (p(x) \vee q(x))$
- $\forall x \; (p(x) \vee q(x)) \not\Rightarrow (\forall x \; p(x) \vee \forall x \; q(x))$

Predicate Logic Constructs: Syntax and Semantics

Variables – Free / Bound (Scopes)

Variables are bounded under the scope of its immediately nested quantifier.

$\forall x \ pred(x, y)$: x is a bound variable and y is a free variable.

$\forall x \ (p(x, y) \land \exists z \ q(x, y, z, w))$: x and z are bounded by $\forall x$ and $\exists z$, respectively, whereas y and w in $q(x, y, z, w)$ are free variables.

$\forall x \ (p(x, y) \land \exists y \ \exists z \ q(x, y, z, w))$: x is bounded by $\forall x$, whereas y in $p(x, y)$ is free. But, both y and z in $q(x, y, z, w)$ is bounded by $\exists y$ and $\exists z$, respectively, whereas w in $q(x, y, z, w)$ is a free variable.

Predicate Logic Constructs: Syntax and Semantics

Variables – Free / Bound (Scopes)

Variables are bounded under the scope of its immediately nested quantifier.

$\forall x \; \mathtt{pred}(x, y)$: x is a bound variable and y is a free variable.

$\forall x \; (p(x, y) \wedge \exists z \; q(x, y, z, w))$: x and z are bounded by $\forall x$ and $\exists z$, respectively, whereas y and w in $q(x, y, z, w)$ are free variables.

$\forall x \; (p(x, y) \wedge \exists y \; \exists z \; q(x, y, z, w))$: x is bounded by $\forall x$, whereas y in $p(x, y)$ is free. But, both y and z in $q(x, y, z, w)$ is bounded by $\exists y$ and $\exists z$, respectively, whereas w in $q(x, y, z, w)$ is a free variable.

Symbols – Functions / Predicates

- Propositional Symbols $\longmapsto$ Predicate Symbols (Boolean outcomes)
- Constant Symbols $\longmapsto$ Function Symbols (Value based outcomes)

Predicate Logic Constructs: Syntax and Semantics

Variables – Free / Bound (Scopes)

Variables are bounded under the scope of its immediately nested quantifier.

$\forall x\ pred(x, y)$: x is a bound variable and y is a free variable.

$\forall x\ (p(x, y) \wedge \exists z\ q(x, y, z, w))$: x and z are bounded by $\forall x$ and $\exists z$, respectively, whereas y and w in $q(x, y, z, w)$ are free variables.

$\forall x\ (p(x, y) \wedge \exists y\ \exists z\ q(x, y, z, w))$: x is bounded by $\forall x$, whereas y in $p(x, y)$ is free. But, both y and z in $q(x, y, z, w)$ is bounded by $\exists y$ and $\exists z$, respectively, whereas w in $q(x, y, z, w)$ is a free variable.

Symbols – Functions / Predicates

- Propositional Symbols $\longmapsto$ Predicate Symbols (Boolean outcomes)
- Constant Symbols $\longmapsto$ Function Symbols (Value based outcomes)

Quantification Eligibility of Variables and Symbols

Variables can be, but Symbols cannot be quantified in First-Order / Predicate Logic.

Incorrect: $\exists p\ \forall x\ [p(x)]$ or $\exists Age\ \forall x\ \exists y\ [gt(Age(x), Age(y))]$

Predicate Logic: Terminalogies

Constant Symbols: $\mathtt{M}, \mathtt{N}, \mathtt{O}, \mathtt{P}, \ldots$

Variable Symbols: $\mathtt{x}, \mathtt{y}, \mathtt{z}, \mathtt{w}, \ldots$

Function Symbols: $\mathtt{F}(\mathtt{x}), \mathtt{G}(\mathtt{x}, \mathtt{y}), \mathtt{H}(\mathtt{x}, \mathtt{y}, \mathtt{z}), \ldots$

Predicate Symbols: $\mathtt{p}(\mathtt{x}), \mathtt{q}(\mathtt{x}, \mathtt{y}), \mathtt{r}(\mathtt{x}, \mathtt{y}, \mathtt{z}), \ldots$

Connectors/Quantifiers: $\neg, \wedge, \vee, \rightarrow$ and $\exists, \forall$

Predicate Logic: Terminalogies

Constant Symbols: $\mathtt{M}, \mathtt{N}, \mathtt{O}, \mathtt{P}, \ldots$

Variable Symbols: $\mathtt{x}, \mathtt{y}, \mathtt{z}, \mathtt{w}, \ldots$

Function Symbols: $\mathtt{F}(\mathtt{x}), \mathtt{G}(\mathtt{x}, \mathtt{y}), \mathtt{H}(\mathtt{x}, \mathtt{y}, \mathtt{z}), \ldots$

Predicate Symbols: $\mathtt{p}(\mathtt{x}), \mathtt{q}(\mathtt{x}, \mathtt{y}), \mathtt{r}(\mathtt{x}, \mathtt{y}, \mathtt{z}), \ldots$

Connectors/Quantifiers: $\neg, \wedge, \vee, \rightarrow$ and $\exists, \forall$

Terms: Variables and Constant Symbols are Terms.
If $\mathtt{t}_1, \mathtt{t}_2, \ldots, \mathtt{t}_k$ are Terms and $\mathtt{F}(\mathtt{x}_1, \mathtt{x}_2, \ldots, \mathtt{x}_k)$ is a Function Symbol, then $\mathtt{F}(\mathtt{t}_1, \mathtt{t}_2, \ldots, \mathtt{t}_k)$ is a Term.

Predicate Logic: Terminalogies

Constant Symbols: $\mathtt{M}, \mathtt{N}, \mathtt{O}, \mathtt{P}, \ldots$

Variable Symbols: $\mathtt{x}, \mathtt{y}, \mathtt{z}, \mathtt{w}, \ldots$

Function Symbols: $\mathtt{F}(\mathtt{x}), \mathtt{G}(\mathtt{x}, \mathtt{y}), \mathtt{H}(\mathtt{x}, \mathtt{y}, \mathtt{z}), \ldots$

Predicate Symbols: $\mathtt{p}(\mathtt{x}), \mathtt{q}(\mathtt{x}, \mathtt{y}), \mathtt{r}(\mathtt{x}, \mathtt{y}, \mathtt{z}), \ldots$

Connectors/Quantifiers: $\neg, \wedge, \vee, \rightarrow$ and $\exists, \forall$

Terms: Variables and Constant Symbols are Terms.
If $\mathtt{t}_1, \mathtt{t}_2, \ldots, \mathtt{t}_k$ are Terms and $\mathtt{F}(\mathtt{x}_1, \mathtt{x}_2, \ldots, \mathtt{x}_k)$ is a Function Symbol, then $\mathtt{F}(\mathtt{t}_1, \mathtt{t}_2, \ldots, \mathtt{t}_k)$ is a Term.

Well-Formed Formula: The WFF (or, simply **formula**) is recursively defined as:

- A proposition is a WFF.
- If $\mathtt{t}_1, \mathtt{t}_2, \ldots, \mathtt{t}_k$ are Terms and $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_2, \ldots, \mathtt{x}_k)$ is a Predicate Symbol, then $\mathtt{P}(\mathtt{t}_1, \mathtt{t}_2, \ldots, \mathtt{t}_k)$ is a WFF.
- If $\mathtt{F}_1$, $\mathtt{F}_2$ are WFFs, then $\neg\mathtt{F}_1$, $(\mathtt{F}_1 \wedge \mathtt{F}_2)$, $(\mathtt{F}_1 \vee \mathtt{F}_2)$ and $(\mathtt{F}_1 \rightarrow \mathtt{F}_2)$ are WFFs.
- If $\mathtt{P}(\mathtt{x}, \ldots)$ is a Predicate where $\mathtt{x}$ is a free variable, then $\forall \mathtt{x}\ \mathtt{P}(\mathtt{x}, \ldots)$ and $\exists \mathtt{x}\ \mathtt{P}(\mathtt{x}, \ldots)$ are WFFs.

Predicate Logic: Interpretations and Inferencing

Structures and Notions

Domain, $\mathcal{D}$: Set of elements/values specified for every interpretation

Constants, C: Get assigned values from given domains

Functions, $F(x_1, x_2, \ldots, x_n)$: Mapping defined as, $(\mathcal{D}_1 \times \cdots \times \mathcal{D}_n) \mapsto \mathcal{D}$
(For e.g., 'sum of x and y' $=$ $\mathtt{sum}(x, y) : \mathtt{Int} \times \mathtt{Int} \mapsto \mathtt{Int}$)

Predicates, $P(x_1, x_2, \ldots, x_n)$: Mapping defined as, $(\mathcal{D}_1 \times \cdots \times \mathcal{D}_n) \mapsto \{\mathtt{True}, \mathtt{False}\}$
(For e.g., 'x is greater than y' $=$ $\mathtt{gt}(x, y) : \mathtt{Int} \times \mathtt{Int} \mapsto \{\mathtt{True}, \mathtt{False}\}$)

Predicate Logic: Interpretations and Inferencing

Structures and Notions

Domain, $\mathcal{D}$: Set of elements/values specified for every interpretation

Constants, $\mathtt{C}$: Get assigned values from given domains

Functions, $\mathtt{F}(\mathtt{x}_1, \mathtt{x}_2, \ldots, \mathtt{x}_n)$: Mapping defined as, $(\mathcal{D}_1 \times \cdots \times \mathcal{D}_n) \mapsto \mathcal{D}$
(For e.g., 'sum of $\mathtt{x}$ and $\mathtt{y}$' $= \mathtt{sum}(\mathtt{x}, \mathtt{y}) : \mathtt{Int} \times \mathtt{Int} \mapsto \mathtt{Int}$)

Predicates, $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_2, \ldots, \mathtt{x}_n)$: Mapping defined as, $(\mathcal{D}_1 \times \cdots \times \mathcal{D}_n) \mapsto \{\mathtt{True}, \mathtt{False}\}$
(For e.g., '$\mathtt{x}$ is greater than $\mathtt{y}$' $= \mathtt{gt}(\mathtt{x}, \mathtt{y}) : \mathtt{Int} \times \mathtt{Int} \mapsto \{\mathtt{True}, \mathtt{False}\}$)

Formal Interpretations of a Formula

Valid: A **valid** formula is true for all interpretations.

Invalid: An **invalid** formula is false under at least one interpretation.

Satisfiable: A **satisfiable** formula is true under at least one interpretation.

Unsatisfiable: An **unsatisfiable** formula is false for all interpretations.

Predicate Logic Deductions: Few Examples

Example-1

$\mathtt{F_1}: \quad \forall \mathtt{x}\ (\mathtt{goes}(\mathtt{Ankush}, \mathtt{x}) \rightarrow \mathtt{goes}(\mathtt{Dog}, \mathtt{x}))$

$\mathtt{F_2}: \quad \mathtt{goes}(\mathtt{Ankush}, \mathtt{School})$

$\mathtt{G}: \quad \mathtt{goes}(\mathtt{Dog}, \mathtt{School})$

$\mathtt{Query}: \quad \mathtt{Is}\ (\mathtt{F_1} \wedge \mathtt{F_2}) \rightarrow \mathtt{G}\ \mathtt{valid?}$

Predicate Logic Deductions: Few Examples

Example-1

$F_1 : \forall x \ (\text{goes}(\text{Ankush}, x) \rightarrow \text{goes}(\text{Dog}, x))$ $F_2 : \text{goes}(\text{Ankush}, \text{School})$

$G : \text{goes}(\text{Dog}, \text{School})$ Query : Is $(F_1 \wedge F_2) \rightarrow G$ valid?

Let, the doamin of variable x be $\mathcal{D} = \{\text{School}, \ \text{Ground}, \ \text{Library}, \ \ldots\}$.
Hence, for $x = \text{School}$, we have, $F_1' : \text{goes}(\text{Ankush}, \text{School}) \rightarrow \text{goes}(\text{Dog}, \text{School})$.

Predicate Logic Deductions: Few Examples

Example-1

$\mathtt{F}_1: \quad \forall \mathtt{x}\ (\mathtt{goes}(\mathtt{Ankush}, \mathtt{x}) \rightarrow \mathtt{goes}(\mathtt{Dog}, \mathtt{x}))$ $\qquad \mathtt{F}_2: \quad \mathtt{goes}(\mathtt{Ankush}, \mathtt{School})$

$\mathtt{G}: \quad \mathtt{goes}(\mathtt{Dog}, \mathtt{School})$ $\qquad$ Query : Is $(\mathtt{F}_1 \wedge \mathtt{F}_2) \rightarrow \mathtt{G}$ valid?

Let, the doamin of variable $\mathtt{x}$ be $\mathcal{D} = \{\mathtt{School},\ \mathtt{Ground},\ \mathtt{Library},\ \ldots\}$.
Hence, for $\mathtt{x} = \mathtt{School}$, we have, $\mathtt{F}'_1: \ \mathtt{goes}(\mathtt{Ankush}, \mathtt{School}) \rightarrow \mathtt{goes}(\mathtt{Dog}, \mathtt{School})$.

Inferencing: $\frac{\substack{F'_1 \\ F_2}}{\therefore\ \mathtt{G}}$, i.e. $\frac{\substack{goes(Ankush, School) \rightarrow goes(Dog, School) \\ goes(Ankush, School)}}{\therefore\ \mathtt{goes}(\mathtt{Dog}, \mathtt{School})}$ (implying $(\mathtt{F}_1 \wedge \mathtt{F}_2) \rightarrow \mathtt{G}$ as **valid**)

Predicate Logic Deductions: Few Examples

Example-1

$\mathtt{F}_1: \quad \forall \mathtt{x}\ (\mathtt{goes}(\mathtt{Ankush}, \mathtt{x}) \rightarrow \mathtt{goes}(\mathtt{Dog}, \mathtt{x}))$ $\quad \mathtt{F}_2: \quad \mathtt{goes}(\mathtt{Ankush}, \mathtt{School})$

$\mathtt{G}: \quad \mathtt{goes}(\mathtt{Dog}, \mathtt{School})$ $\quad$ Query : Is $(\mathtt{F}_1 \wedge \mathtt{F}_2) \rightarrow \mathtt{G}$ valid?

Let, the doamin of variable x be $\mathcal{D} = \{\mathtt{School},\ \mathtt{Ground},\ \mathtt{Library},\ \ldots\}$.
Hence, for $\mathtt{x} = \mathtt{School}$, we have, $\mathtt{F}_1'$: $\mathtt{goes}(\mathtt{Ankush}, \mathtt{School}) \rightarrow \mathtt{goes}(\mathtt{Dog}, \mathtt{School})$.

Inferencing: $\dfrac{F_1' \quad F_2}{\therefore\ \mathtt{G}}$, i.e. $\dfrac{goes(Ankush, School) \rightarrow goes(Dog, School) \quad goes(Ankush, School)}{\therefore\ \mathtt{goes(Dog,School)}}$ (implying $(\mathtt{F}_1 \wedge \mathtt{F}_2) \rightarrow \mathtt{G}$ as **valid**)

Example-2

$\mathtt{F}_1: \quad \forall \mathtt{x}\ (\mathtt{contractor}(\mathtt{x}) \rightarrow \neg \mathtt{dependable}(\mathtt{x}))$

$\mathtt{F}_2: \quad \exists \mathtt{x}\ (\mathtt{engineer}(\mathtt{x}) \wedge \mathtt{contractor}(\mathtt{x}))$

$\mathtt{G}: \quad \exists \mathtt{x}\ (\mathtt{engineer}(\mathtt{x}) \wedge \neg \mathtt{dependable}(\mathtt{x}))$ $\quad$ Query : Is $(\mathtt{F}_1 \wedge \mathtt{F}_2) \rightarrow \mathtt{G}$ valid?

Predicate Logic Deductions: Few Examples

Example-1

$F_1 : \forall x\ (goes(Ankush, x) \rightarrow goes(Dog, x))$ $F_2 : goes(Ankush, School)$

$G : goes(Dog, School)$ Query : Is $(F_1 \wedge F_2) \rightarrow G$ valid?

Let, the doamin of variable x be $\mathcal{D} = \{School,\ Ground,\ Library,\ \ldots\}$.
Hence, for $x = School$, we have, $F_1' : goes(Ankush, School) \rightarrow goes(Dog, School)$.

Inferencing: $\frac{\substack{F_1' \\ F_2}}{\therefore\ G}$, i.e. $\frac{\substack{goes(Ankush,School) \rightarrow goes(Dog,School) \\ goes(Ankush,School)}}{\therefore\ goes(Dog,School)}$ (implying $(F_1 \wedge F_2) \rightarrow G$ as **valid**)

Example-2

$F_1 : \forall x\ (contractor(x) \rightarrow \neg dependable(x))$

$F_2 : \exists x\ (engineer(x) \wedge contractor(x))$

$G : \exists x\ (engineer(x) \wedge \neg dependable(x))$ Query : Is $(F_1 \wedge F_2) \rightarrow G$ valid?

Here, let for $x = A$, we can produce, $\substack{F_1' :\ contractor(A) \rightarrow \neg dependable(A) \\ F_2' :\ engineer(A) \wedge contractor(A)}$.
We can prove, $G' : engineer(A) \wedge \neg dependable(A)$, implying $(F_1 \wedge F_2) \rightarrow G$ as **valid**.

Predicate Logic Deductions: Few Examples

Example-1

$\mathtt{F_1}: \ \forall \mathtt{x}\ (\mathtt{goes(Ankush, x)} \rightarrow \mathtt{goes(Dog, x)})$ $\quad \mathtt{F_2}: \ \mathtt{goes(Ankush, School)}$

$\mathtt{G}: \ \mathtt{goes(Dog, School)}$ $\quad$ Query : Is $(\mathtt{F_1} \wedge \mathtt{F_2}) \rightarrow \mathtt{G}$ valid?

Let, the doamin of variable x be $\mathcal{D} = \{\mathtt{School},\ \mathtt{Ground},\ \mathtt{Library},\ \ldots\}$.
Hence, for $\mathtt{x} = \mathtt{School}$, we have, $\mathtt{F}'_1: \ \mathtt{goes(Ankush, School)} \rightarrow \mathtt{goes(Dog, School)}$.

Inferencing: $\dfrac{F'_1 \quad F_2}{\therefore\ \mathtt{G}}$, i.e. $\dfrac{goes(Ankush, School) \rightarrow goes(Dog, School) \quad goes(Ankush, School)}{\therefore\ \mathtt{goes(Dog, School)}}$ (implying $(\mathtt{F_1} \wedge \mathtt{F_2}) \rightarrow \mathtt{G}$ as **valid**)

Example-2

$\mathtt{F_1}: \ \forall \mathtt{x}\ (\mathtt{contractor(x)} \rightarrow \neg \mathtt{dependable(x)})$

$\mathtt{F_2}: \ \exists \mathtt{x}\ (\mathtt{engineer(x)} \wedge \mathtt{contractor(x)})$

$\mathtt{G}: \ \exists \mathtt{x}\ (\mathtt{engineer(x)} \wedge \neg \mathtt{dependable(x)})$ $\quad$ Query : Is $(\mathtt{F_1} \wedge \mathtt{F_2}) \rightarrow \mathtt{G}$ valid?

Here, let for $\mathtt{x} = \mathtt{A}$, we can produce, $F'_1: \ contractor(A) \rightarrow \neg dependable(A)$; $F'_2: \ engineer(A) \wedge contractor(A)$.
We can prove, $\mathtt{G}': \ \mathtt{engineer(A)} \wedge \neg \mathtt{dependable(A)}$, implying $(\mathtt{F_1} \wedge \mathtt{F_2}) \rightarrow \mathtt{G}$ as **valid**.

Inferencing: $\dfrac{F'_1 \quad F'_2}{\therefore\ \mathtt{G}'}$, because $\dfrac{contractor(A) \rightarrow \neg dependable(A) \quad engineer(A) \wedge contractor(A)}{\therefore\ \neg \mathtt{dependable(A)}}$ and $\dfrac{engineer(A) \wedge contractor(A) \quad \neg dependable(A)}{\therefore\ \mathtt{engineer(A)} \wedge \neg \mathtt{dependable(A)}}$.

Predicate Logic: Inferencing and Deduction Rules

Rule of Universal Specification

Base Rule:

- If $\forall x\ p(x)$ is true, then $p(A)$ is true for each element A from the domain of x.
- If $\exists x\ p(x)$ is true, then $p(A)$ is true for at least one element A from the domain of x.

Predicate Logic: Inferencing and Deduction Rules

Rule of Universal Specification

Base Rule:

- If $\forall x\ p(x)$ is true, then $p(A)$ is true for each element A from the domain of x.
- If $\exists x\ p(x)$ is true, then $p(A)$ is true for at least one element A from the domain of x.

Few Derived Rules:

$$\frac{\forall x\ [p(x) \rightarrow q(x)] \quad p(A)}{\therefore\ q(A)}$$
(*Modus Ponens*)

$$\frac{\forall x\ [p(x) \rightarrow q(x)] \quad \neg q(A)}{\therefore\ \neg p(A)}$$
(*Modus Tollens*)

$$\frac{\forall x\ [(p(x) \vee q(x)) \rightarrow \neg r(x)] \quad r(A)}{\therefore\ \neg p(A)}$$

Predicate Logic: Inferencing and Deduction Rules

Rule of Universal Specification

Base Rule:

- If $\forall x\ p(x)$ is true, then $p(A)$ is true for each element A from the domain of x.
- If $\exists x\ p(x)$ is true, then $p(A)$ is true for at least one element A from the domain of x.

Few Derived Rules:

$$\frac{\begin{array}{c}\forall x\ [p(x) \rightarrow q(x)] \\ p(A)\end{array}}{\therefore\ q(A)}$$
(*Modus Ponens*)

$$\frac{\begin{array}{c}\forall x\ [p(x) \rightarrow q(x)] \\ \neg q(A)\end{array}}{\therefore\ \neg p(A)}$$
(*Modus Tollens*)

$$\frac{\begin{array}{c}\forall x\ [(p(x) \vee q(x)) \rightarrow \neg r(x)] \\ r(A)\end{array}}{\therefore\ \neg p(A)}$$

Rule of Universal Generalization

Base Rule: If $\forall x\ p(x)$ is true, then $p(c)$ is true for an arbitrarily chosen element c from the domain of x.

Predicate Logic: Inferencing and Deduction Rules

Rule of Universal Specification

Base Rule:

- If $\forall x\ p(x)$ is true, then $p(A)$ is true for each element A from the domain of x.
- If $\exists x\ p(x)$ is true, then $p(A)$ is true for at least one element A from the domain of x.

Few Derived Rules:

$$\frac{\forall x\ [p(x) \rightarrow q(x)] \quad p(A)}{\therefore\ q(A)}$$
(*Modus Ponens*)

$$\frac{\forall x\ [p(x) \rightarrow q(x)] \quad \neg q(A)}{\therefore\ \neg p(A)}$$
(*Modus Tollens*)

$$\frac{\forall x\ [(p(x) \vee q(x)) \rightarrow \neg r(x)] \quad r(A)}{\therefore\ \neg p(A)}$$

Rule of Universal Generalization

Base Rule: If $\forall x\ p(x)$ is true, then $p(c)$ is true for an arbitrarily chosen element c from the domain of x.

Few Derived Rules:

$$\frac{\forall x\ [p(x) \rightarrow q(x)] \quad \forall x\ [q(x) \rightarrow r(x)]}{\therefore\ \forall x\ [p(x) \rightarrow r(x)]}$$
(*Universal Syllogism*)

$$\frac{\forall x\ [p(x) \vee q(x)] \quad \forall x\ [(\neg p(x) \wedge q(x)) \rightarrow r(x)]}{\therefore\ \forall x\ [\neg r(x) \rightarrow p(x)]}$$

Limitations of Predicate Logic

Note: Predicate Logic can model **any computable** function.

Extensions to Predicate Logic

Higher-Order Logics: Can also quantify symbols along with quantifying variables.

$$\forall p \; ((p(0) \land (\forall x \; (p(x) \rightarrow p(S(x)))) \rightarrow \forall y \; (p(y)))$$

[Guess what this formula expresses? Hint: A Math Theorem!]

Limitations of Predicate Logic

Note: Predicate Logic can model **any computable** function.

Extensions to Predicate Logic

Higher-Order Logics: Can also quantify symbols along with quantifying variables.

$$\forall \mathtt{p}\ ((\mathtt{p}(0) \wedge (\forall \mathtt{x}\ (\mathtt{p}(\mathtt{x}) \rightarrow \mathtt{p}(\mathtt{S}(\mathtt{x}))))) \rightarrow \forall \mathtt{y}\ (\mathtt{p}(\mathtt{y})))$$

[Guess what this formula expresses? Hint: A Math Theorem!]

Temporal Logics: Can also relate two time universes using additional constructs, such as, `next`, `future`, `always`, `until`.

Limitations of Predicate Logic

Note: Predicate Logic can model **any computable** function.

Extensions to Predicate Logic

Higher-Order Logics: Can also quantify symbols along with quantifying variables.

$$\forall p \ ((p(0) \wedge (\forall x \ (p(x) \rightarrow p(S(x))))) \rightarrow \forall y \ (p(y)))$$

[Guess what this formula expresses? Hint: A Math Theorem!]

Temporal Logics: Can also relate two time universes using additional constructs, such as, `next`, `future`, `always`, `until`.

Unsolvable Problem Specifications

Russell's Paradox: The barber shaves all those who do not shave themselves. Does the barber shaves himself?

Limitations of Predicate Logic

Note: Predicate Logic can model **any computable** function.

Extensions to Predicate Logic

Higher-Order Logics: Can also quantify symbols along with quantifying variables.

$$\forall p \; ((p(0) \wedge (\forall x \; (p(x) \rightarrow p(S(x)))) \rightarrow \forall y \; (p(y)))$$

[Guess what this formula expresses? Hint: A Math Theorem!]

Temporal Logics: Can also relate two time universes using additional constructs, such as, `next`, `future`, `always`, `until`.

Unsolvable Problem Specifications

Russell's Paradox: The barber shaves all those who do not shave themselves. Does the barber shaves himself?

- There is a single barber in the town.
- Those and only those who do not shave themselves are shaved by the barber.
- Then, who shaves the barber? **Undecidable!**

Thank You!