FIRST ORDER LOGIC

Thomas Tiam-Lee, PhD

Norshuhani Zamin, PhD

Last update: October 20, 2023

Limitations of Propositional Logic

- Some real-world ideas are impractical to represent in propositional logic.
  - **Example:**
    - All fruits are good for our health.
      - *Apple is good for our health*
      - *Banana is good for our health*
      - *Pear is good for our health*
      - *Strawberry is good for our health*
      - *and so on...*

Need to make a proposition for all of these.

Limitations of Propositional Logic

- Some real-world ideas are impractical to represent in propositional logic.
  - **Example:**
    - If some students in a class talks, then the class is noisy.
      - *If Akmal talks, then the class is noisy.*
      - *If Farhana talks, then the class is noisy.*
      - *If Yin May talks, then the class is noisy.*
      - *If Jimmy talks, then the class is noisy.*
      - *and so on...*

Need to make a proposition for all of these.

Modeling: First Order Logic

- Extension of propositional logic
- Allows for predicates, variables, constants and quantifiers

- **Example:**
  - $\forall X\ (\mathbf{Fruit}(X) \rightarrow \mathbf{GoodForHealth}(X))$
    - *X is a fruit implies than X is good for health*
  - $(\exists X\ \mathbf{Talks}(X) \wedge \mathbf{StudentOfClass}(X, Y)) \rightarrow \mathbf{Noisy}(Y)$
    - *If there exists an X such that X talks and X is a student in the class Y, then Y is noisy*

First Order Logic: Syntax

- A **term** can be:
  - Constant symbol (e.g., arithmetic, linguistic)
  - Variable (e.g., X, Y, P, Q)
  - Function of terms (e.g., Sum(3, X), like(P, sandwich)

- A **formula** can be:
  - Atomic formula: **predicate applied to terms** (e.g., Knows(X, math))
  - Connectives applied to formulas (e.g., Student(X) $\rightarrow$ Knows(X, math))
  - Quantifiers applied to formulas (e.g, $\forall X$ Student(X) $\rightarrow$ Knows(X, math)

First Order Logic: Semantics

- **Predicate**: represents a relation or property that can be attributed to one or more objects within the domain of discourse.
  - Have no inherent meaning; its semantic is explicitly defined.
  - Arguments of a predicate can be a **constant** or a **variable**.

- Example:
  - Fruit(X) means that X is a fruit.
  - Parent(X, Y) means that X is a parent of Y.

- Make sure that predicate semantics are **consistent**!

First Order Logic: Semantics

- **Quantifiers:** Used to express the extent to which a given predicate holds true.

  - **Universal Quantifier (∀)**: means the statement is true for **all** possible substitutions of the variable
  - **Existential Quantifier (∃)**: means the statement is true for **at least one** substitution of the variable

First Order Logic: Semantics

- **Examples:**
  - All students know math.
    - $\forall X$ Student(X) $\rightarrow$ Knows(X, math)
  - There exists a student who knows math.
    - $\exists X$ Student(X) $\wedge$ Knows(X, math)

- Almost all of the time, $\forall$ **is paired with** $\rightarrow$ while $\exists$ **is paired with** $\wedge$!

Quantifier Properties

- De Morgan's Law for quantifiers:
    1. $\neg\exists X$ P(X) is equivalent to $\forall X \neg$ P(X)
    2. $\neg\forall X$ P(X) is equivalent to $\exists X \neg$ P(X)

Examples:

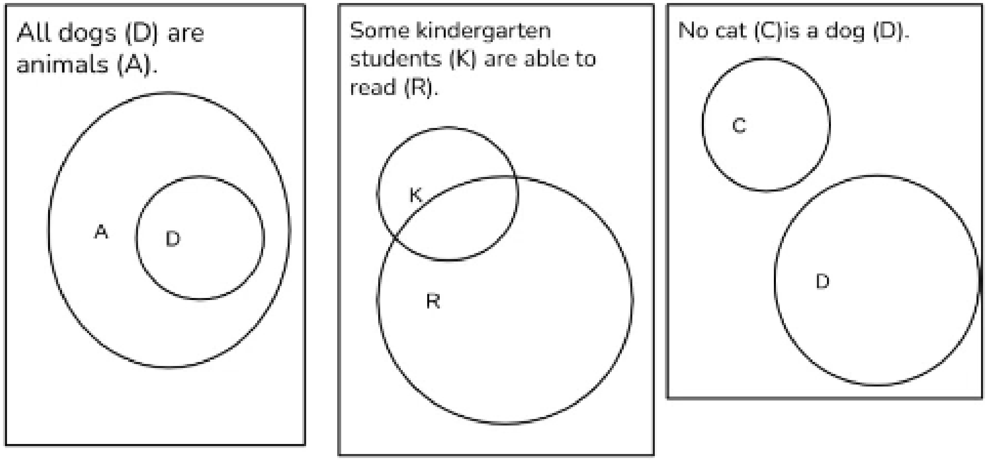
1. Nobody likes tax = Everybody do not like tax

   ~∃X like_tax(X) ≡ ∀X ~like_tax(X) * by law **#1**
2. Not everybody play soccer = Somebody do not play soccer

   ~∀X play(X,soccer) ≡ ∃X ~play(X,soccer) * by law **#2**

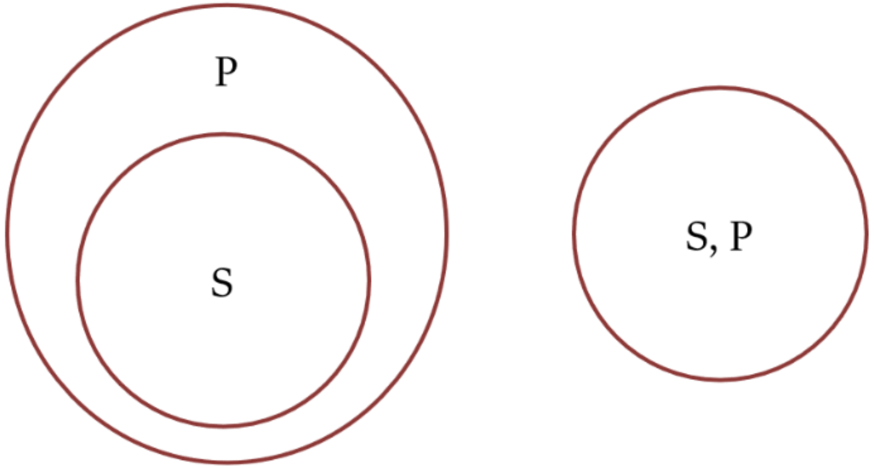
Theory of Sets

Venn Diagram Examples



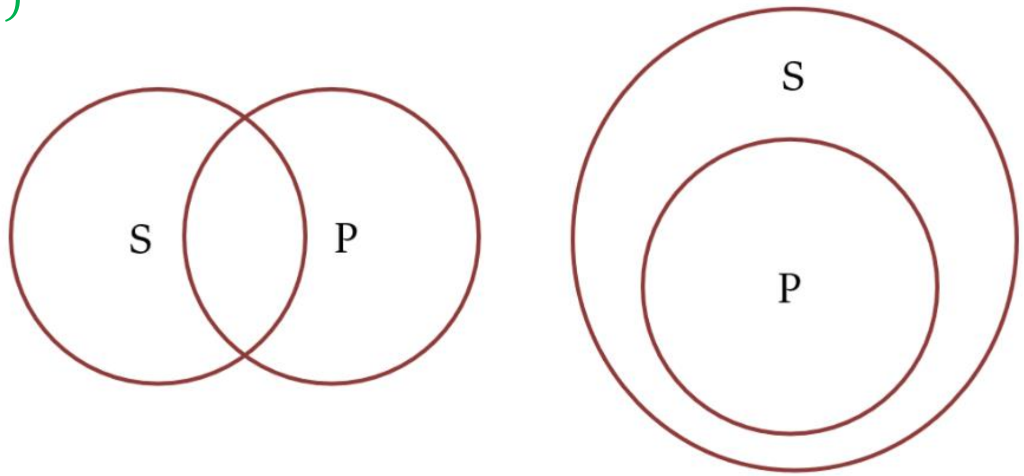
$$\forall X\ \exists Y\ \ Knows(X,Y)\ not\ the\ same\ as\ \forall Y\ \exists X\ \ Knows(X,Y)$$

All S are P - We can showcase it using two possible Venn Diagrams.



Some S are P –

The most common way to represent this are:



Examples of First Order Logic

Predicate	Meaning
Student(X)	X is a student
Course(Y)	Y is a course
Takes(X, Y)	X has already taken Y (X and Y are not necessarily students nor courses)

- Every student has taken at least one course.

Examples of First Order Logic

Predicate	Meaning
Student(X)	X is a student
Course(Y)	Y is a course
Takes(X, Y)	X has already taken Y (X and Y are not necessarily students nor courses)

- Every student has taken at least one course.
  - $\forall X \; \exists Y \; Student(X) \land Course(Y) \; \rightarrow \; Takes(X, Y)$

Examples of First Order Logic

Predicate	Meaning
Student(X)	X is a student
Course(Y)	Y is a course
Takes(X, Y)	X has already taken Y (X and Y are not necessarily students nor courses)

- Bob takes all courses that Alice has taken.

Examples of First Order Logic

Predicate	Meaning
Student(X)	X is a student
Course(Y)	Y is a course
Takes(X, Y)	X has already taken Y (X and Y are not necessarily students nor courses)

- Bob takes all courses that Alice has taken.
  - $\forall Y$ Course(Y) $\wedge$ Takes(Alice, Y) $\rightarrow$ Takes(Bob, Y)

Examples of First Order Logic

Predicate	Meaning
Student(X)	X is a student
Course(Y)	Y is a course
Takes(X, Y)	X has already taken Y (X and Y are not necessarily students nor courses)

- Some courses are taken by every student.

Examples of First Order Logic

Predicate	Meaning
Student(X)	X is a student
Course(Y)	Y is a course
Takes(X, Y)	X has already taken Y (X and Y are not necessarily students nor courses)

- Some courses are taken by every student.
  - $\forall X \exists Y$ Student(X) $\wedge$ Course(Y) $\rightarrow$ Takes(Y, X)

First Order Vs. Propositional Logic

- You can think of first order logic as just a **syntactic sugar for propositional logic**!

KB — **First Order Logic**

Student(Alice)
Student(Bob)
$\forall X$ Student(X) $\rightarrow$ Smart(X)
$\exists X$ Student(X) $\wedge$ Creative(X)

$\Longleftrightarrow$

KB — **Propositional Logic**

StudentAlice
StudentBob
StudentAlice $\rightarrow$ SmartAlice
StudentBob $\rightarrow$ SmartBob
(StudentAlice $\wedge$ CreativeAlice) $\vee$
(StudentBob $\wedge$ CreativeBob)

Modus Ponens Inference Rule

- Since first order logic is just a syntactic sugar for propositional logic, **we should be able to apply the same inference rules**.

- Modus Ponens is applicable to first order logic.
- For it to be complete, statements must be expressed in Horn clauses.

Unification

- Process of **finding substitutions for variables** in multiple predicates to **make them identical**.

- **Unify p(a,X) and p(a,b)**
  - Answer: {X/b}
- **Unify p(a,X) and p(Y,b)**
  - Answer: {Y/a, X/b}
- **Unify p(a,X) and p(Y,f(Y))**
  - Answer: {Y/a, X/f(a)}
- **Unify p(a,X) and p(X,b)**
  - Answer: Unification failure.
  - Justification: If {X/a} then p(a,a) <> p(a,b)
- **Unify p(a,b) and p(X,X)**
  - Answer: Unification failure.
  - Justification: If {X/a} then p(a,b) <> p(a,a)

Modus Ponens Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\forall X$ Student(X) → Smart(X)
$\forall X \forall Y$ Smart(X) ∧ Parent(Y, X) → Proud(Y)

Modus Ponens Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\forall X$ Student(X) → Smart(X)
$\forall X \forall Y$ Smart(X) ∧ Parent(Y, X) → Proud(Y)

Unification:
{X/Alice}

$$\frac{\text{Student(Alice)}, \forall X\ \text{Student}(X) \rightarrow \text{Smart}(X)}{\text{Smart(Alice)}}$$

Modus Ponens Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\forall X$ Student(X) $\rightarrow$ Smart(X)
$\forall X \forall Y$ Smart(X) $\land$ Parent(Y, X) $\rightarrow$ Proud(Y)
Smart(Alice)

Modus Ponens Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\forall X$ Student(X) $\rightarrow$ Smart(X)
$\forall X \forall Y$ Smart(X) $\wedge$ Parent(Y, X) $\rightarrow$ Proud(Y)
Smart(Alice)

The knowledge base changed, so we need to scan the knowledge base again.

Modus Ponens Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\forall X$ Student(X) $\rightarrow$ Smart(X)
$\forall X \forall Y$ Smart(X) $\wedge$ Parent(Y, X) $\rightarrow$ Proud(Y)
Smart(Alice)

Unification:
{X/Alice, Y/Bob}

$$\frac{\text{Smart(Alice), Parent(Bob, Alice), } \forall X \forall Y \text{ Smart}(X) \wedge \text{Parent}(Y, X) \rightarrow \text{Proud}(Y)}{\text{Proud(Bob)}}$$

Modus Ponens Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\forall X$ Student(X) → Smart(X)
$\forall X \forall Y$ Smart(X) ∧ Parent(Y, X) → Proud(Y)
Smart(Alice)
Proud(Bob)

Modus Ponens Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\forall X$ Student(X) $\rightarrow$ Smart(X)
$\forall X \forall Y$ Smart(X) $\wedge$ Parent(Y, X) $\rightarrow$ Proud(Y)
Smart(Alice)
Proud(Bob)

The knowledge base changed, so we need to scan the knowledge base again.

Modus Ponens Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\forall X$ Student(X) $\rightarrow$ Smart(X)
$\forall X \forall Y$ Smart(X) $\wedge$ Parent(Y, X) $\rightarrow$ Proud(Y)
Smart(Alice)
Proud(Bob)

Converged!

Resolution Inference

- Since first order logic is just a syntactic sugar for propositional logic, **we should be able to apply the same inference rules**.

- Resolution inference is applicable to first order logic.
- Formulas must be converted to Conjunctive Normal Form (CNF).

Converting First Order Logic to CNF

1. Remove the implications

   $p \rightarrow q \equiv \neg p \vee q$

   $p \wedge q \rightarrow r \equiv \neg p \vee \neg q \vee r$

   $p \leftrightarrow q \equiv (\neg p \vee q) \wedge (\neg q \vee p)$

   $p \wedge q \equiv p\ ,\ q$ (separate p and q into individual clause)

2. Push negations inside

   $\neg(p \vee q) \equiv \neg p \wedge \neg q$

   $\neg(p \wedge q) \equiv \neg p \vee \neg q$

3. Remove double negations

   $\neg\neg p \equiv p$

4. Distribute $\vee$ over $\wedge$

   $p \vee (q \vee r) \equiv (p \vee q) \wedge (p \vee r)$

5. Remove all quantifiers

Removing Quantifiers

- **Universal Quantifiers:**
  - Just remove them. It is assumed that all formulas in the knowledge base are universally quantified.
    - $\forall X$ Student(X) becomes Student(X)

- **Existential Quantifiers:**
  - Replace the variable with a Skolem variable* and remove the existential quantifier.
    - $\exists X$ Knows(X, math) becomes Knows(a, math)

*Advanced concept, left as additional reading

Example Conversion to CNF

$\forall X$ Student(X) $\rightarrow$ Smart(X) Initial formula

Example Conversion to CNF

$\forall X$ Student(X) → Smart(X)	Initial formula
$\forall X \neg$ Student(X) ∨ Smart(X)	Remove Implication

Example Conversion to CNF

$\forall X\ \text{Student}(X) \rightarrow \text{Smart}(X)$	Initial formula
$\forall X \neg\ \text{Student}(X) \vee \text{Smart}(X)$	Remove Implication
$\neg\ \textbf{Student}(X) \vee \textbf{Smart}(X)$	Remove quantifiers

Resolution Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\forall X$ Student(X) $\rightarrow$ Smart(X)
$\forall X \forall Y$ Smart(X) $\wedge$ Parent(Y, X) $\rightarrow$ Proud(Y)

Resolution Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\boxed{\forall X \text{ Student}(X) \rightarrow \text{Smart}(X)}$
$\forall X \forall Y \text{ Smart}(X) \land \text{Parent}(Y, X) \rightarrow \text{Proud}(Y)$

Convert to CNF

Resolution Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\neg$Student(X) $\vee$ Smart(X)
$\forall X \forall Y$ Smart(X) $\wedge$ Parent(Y, X) $\rightarrow$ Proud(Y)

Resolution Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\neg$Student(X) $\vee$ Smart(X)
$\boxed{\forall X \forall Y \text{ Smart}(X) \wedge \text{Parent}(Y, X) \rightarrow \text{Proud}(Y)}$

Convert to CNF

Resolution Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\neg$Student(X) $\vee$ Smart(X)
$\neg$Smart(X) $\vee$ $\neg$ Parent(Y, X) $\vee$ Proud(Y)

Resolution Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\neg$Student(X) $\lor$ Smart(X)
$\neg$Smart(X) $\lor$ $\neg$ Parent(Y, X) $\lor$ Proud(Y)

Unification:
$\{X$/Alice$\}$

$$\frac{\text{Student(Alice)}, \neg\text{Student}(X) \lor \text{Smart}(X)}{\text{Smart(Alice)}}$$

Resolution Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\neg$Student(X) $\lor$ Smart(X)
$\neg$Smart(X) $\lor$ $\neg$ Parent(Y, X) $\lor$ Proud(Y)
Smart(Alice)

Resolution Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\neg$Student(X) $\vee$ Smart(X)
$\neg$Smart(X) $\vee$ $\neg$ Parent(Y, X) $\vee$ Proud(Y)
Smart(Alice)

Unification:
{X/Alice, Y/Bob}

$$\frac{\text{Parent(Bob, Alice)}, \neg\text{Smart}(X) \vee \neg\,\text{Parent}(Y, X) \vee \text{Proud}(Y)}{\neg\text{Smart(Alice)} \vee \text{Proud(Bob)}}$$

Resolution Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\neg$Student(X) $\vee$ Smart(X)
$\neg$Smart(X) $\vee$ $\neg$ Parent(Y, X) $\vee$ Proud(Y)
Smart(Alice)
$\neg$Smart(Alice) $\vee$ Proud(Bob)

Resolution Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\neg$Student(X) $\lor$ Smart(X)
$\neg$Smart(X) $\lor$ $\neg$ Parent(Y, X) $\lor$ Proud(Y)
Smart(Alice)
$\neg$Smart(Alice) $\lor$ Proud(Bob)

The knowledge base changed, so we need to scan the knowledge base again.

Resolution Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
$\neg$Student(X) $\vee$ Smart(X)
$\neg$Smart(X) $\vee$ $\neg$ Parent(Y, X) $\vee$ Proud(Y)
Smart(Alice)
$\neg$Smart(Alice) $\vee$ Proud(Bob)

$$\frac{\text{Smart(Alice)}, \neg\text{Smart(Alice)} \vee \text{Proud(Bob)}}{\text{Proud(Bob)}}$$

Resolution Inference Rule Example

KB

Student(Alice)
Parent(Bob, Alice)
¬Student(X) ∨ Smart(X)
¬Smart(X) ∨ ¬ Parent(Y, X) ∨ Proud(Y)
Smart(Alice)
¬Smart(Alice) ∨ Proud(Bob)
Proud(Bob)

Converged!

Acknowledgments

- Stanford University CS221 Autumn 2021 course. Available online at: https://stanford-cs221.github.io/autumn2021
- Previous CSINTSY slides by the following instructors:
  - Raymund Sison, PhD
  - Joanna Pauline Rivera
- Previous slides by the following instructors:
  - Norshuhani Zamin, PhD