Propositional and First Order Logic: Basics

Why Logic?

Express complicated things in a succinct way

Then make inferences.

```
All students like KT.
>>>>I learned something.
Alice does not like KT.
>>>>I learned something.
Is Alice a student?
>>>>No
```

Natural Language

A cent is better than nothing

Nothing is better than world peace

A cent is better than world peace

Natural language is slippery

Ingredients of a logic

Syntax: **defines a set of valid formulas**

What kinds of expressions are valid on this language.

Semantics: for each formula specify a set of models

What each formula means

Reasoning: given a set of formulas which rules can be applied to derive new information

Logics in this Class

- Propositional Logic
- First Order Logic
- Description Logic

Expressivity vs Computational efficiency

Propositional Logic

The simplest, and most abstract logic we can study is called propositional logic.

Definition: A proposition is a statement that can be either true or false; it must be one or the other, and it cannot be both.

- EXAMPLES. The following are propositions:

– All humans are mortal

– Socrates is human

– Socrates is mortal

whereas the following are not:

are you going out somewhere?

– 2+3

Syntax -The Alphabet

The alphabet of the Propositional Logic is composed of:

Connectives:

$\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\Rightarrow$ (implication), $\Leftrightarrow$ (biconditional)

Parenthesis: (,)

Propositional Symbols (or propositional variables): P, Q, R

Syntax -Well formed formulas

Propositional Symbols (atomic formulas): P, Q, R, …

Formula: if f, g are formulas, so are the following:

- $(\neg f)$
- $(f \land g)$
- $(f \lor g)$
- $(f \Rightarrow g)$
- $(f \Leftrightarrow g)$

Nothing else is a formula.

Parenthesis elimination: Precedence

1. $\neg$
2. $\wedge$
3. $\vee$
4. $\Rightarrow$
5. $\Leftrightarrow$

Thus $P \vee Q \rightarrow \neg R$ is equivalent to $(P \vee Q) \rightarrow (\neg R)$.

If the intended meaning is $P \vee (Q \rightarrow \neg R)$ then parentheses must be used.

Syntax -Well formed formulas

Propositional Symbols (atomic formulas or atoms): P, Q, R,…

Formula: if $\boldsymbol{f}, \boldsymbol{g}$ are formulas, so are the following:

- $(\neg \boldsymbol{f})$
- $(\boldsymbol{f} \wedge \boldsymbol{g})$
- $(\boldsymbol{f} \vee \boldsymbol{g})$
- $(\boldsymbol{f} \Rightarrow \boldsymbol{g})$
- $(\boldsymbol{f} \Leftrightarrow \boldsymbol{g})$

A

¬¬A

¬A ⇒ B

¬A ∧ (¬ A ⇒ B) ⇒ ¬ A ∨ C

¬A (¬ A ⇒ B) ⇒ ¬ A ∨ C

¬A (¬ A ⇒ B) ¬ A ∨ C

A+B

Semantics -Interpretation

In expression E: $a * b + 2$,

Assign values to a and b and then evaluate the expression.

For example, if a = 2 and b = 3 then E evaluates to 8

In propositional logic, **truth values** are assigned to the atoms of a formula in order to evaluate the truth value of the formula

Definition: Let **A** be a **formula** and let $\mathbf{P_A}$ be the set of **atoms** appearing in A. An **interpretation** for A is a total function $\mathbf{I_A} : \mathbf{P_A} \rightarrow \{T,F\}$ that assigns one of the truth values T or F to every atom in P_A.

Truth Tables

p	q	$p \wedge q$
F	F	F
F	T	F
T	F	F
T	T	T

p	q	$p \vee q$
F	F	F
F	T	T
T	F	T
T	T	T

p	q	$p \Rightarrow q$
F	F	T
F	T	T
T	F	F
T	T	T

p	q	$p \Leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

p	$\neg p$
F	T
T	F

If p is false, then we don’t care about q, and by default, make $p \Rightarrow q$ evaluate to T in this case

Equivalences

Two formulas are **logically equivalent** if the columns in a truth table giving their truth values agree.

We write this as $p \equiv q$

De Morgan Laws:

- $\neg(P \lor Q) \equiv \neg P \land \neg Q$
- $\neg(P \land Q) \equiv \neg P \lor \neg Q$

Important Logical Equivalences

Domination laws:	$p \vee \mathbf{T} \equiv \mathbf{T}$, $p \wedge \mathbf{F} \equiv \mathbf{F}$
Identity laws:	$p \wedge \mathbf{T} \equiv p$, $p \vee \mathbf{F} \equiv p$
Idempotent laws:	$p \wedge p \equiv p$, $p \vee p \equiv p$
Double negation law:	$\neg(\neg p) \equiv p$
Negation laws:	$p \vee \neg p \equiv \mathbf{T}$, $p \wedge \neg p \equiv \mathbf{F}$

The first of the Negation laws is also called “law of excluded middle”.
Latin: “tertium non datur”.

Commutative laws:	$p \wedge q \equiv q \wedge p$, $p \vee q \equiv q \vee p$
Associative laws:	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
	$(p \vee q) \vee r \equiv p \vee (q \vee r)$
Distributive laws:	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
Absorption laws:	$p \vee (p \wedge q) \equiv p$, $p \wedge (p \vee q) \equiv p$

More Logical Equivalences

TABLE 7 Logical Equivalences Involving Conditional Statements.
$\boxed{p \rightarrow q \equiv \neg p \vee q}$
$p \rightarrow q \equiv \neg q \rightarrow \neg p$
$p \vee q \equiv \neg p \rightarrow q$
$p \wedge q \equiv \neg(p \rightarrow \neg q)$
$\neg(p \rightarrow q) \equiv p \wedge \neg q$
$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$
$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$
$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$
$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$

TABLE 8 Logical Equivalences Involving Biconditional Statements.
$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Inference Rules

F (a set of formulas) +inference rules **derivation** of new formulas

Example:

Modus Ponens (simple edition):

$$\frac{A \qquad A \rightarrow B}{B}$$

Satisfiability, Tautology, Contradiction

A formula is:

- satisfiable, if its truth table contains true at least once. Example: $p \land q$.
- a tautology, if it is always true. Example: $p \lor \neg p$.
- a contradiction, if it always false. Example: $p \land \neg p$.
- a contingency, if it is neither a tautology nor a contradiction. Example: p.

Conjunctive Normal Form

A **literal** is either an atomic formula, or the negation of one.

Examples: p, $\neg p$.

A **clause** is a disjunction of literals.

Example: $p \lor \neg q \lor r$.

A formula in **conjunctive normal form** (CNF) is a conjunction of clauses.

Example: $(p \lor \neg q \lor r) \land (\neg p \lor \neg r)$

Transformation into Conjunctive Normal Form

1. Express all other operators by conjunction, disjunction and negation.
2. Push negations inward by De Morgan’s laws and the double negation law until negations appear **only in literals**.
3. Use the commutative, associative and distributive laws to obtain the correct form.
4. Simplify with domination, identity, idempotent, and negation laws.

Example: Transformation into CNF

Transform the following formula into CNF.

$$\neg(p \rightarrow q) \vee (r \rightarrow p)$$

1. Express implication by disjunction and negation.

$$\neg(\neg p \vee q) \vee (\neg r \vee p)$$

2. Push negation inwards by De Morgan’s laws and double negation.

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

3. Convert to CNF by associative and distributive laws.

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

4. Optionally simplify by commutative and idempotent laws.

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

and by commutative and absorption laws

$$(p \vee \neg r)$$

Limitations of Propositional Logic – Example

Alice and Bob like KT.

$$\text{AliceLikesKT} \wedge \text{BobLikesKT}$$

***All** students like KT.*

$$\text{AliceIsAStudent} \rightarrow \text{AliceLikesKT}$$

$$\text{BobIsAStudent} \rightarrow \text{BobLikesKT}$$

$$\ldots$$

Every integer greater than 2 is the sum of two primes

???

Limitations of Propositional Logic

Propositional Logic has **limited** expressive power

It is more natural to describe the knowledge of the world with Objects and Predicates.

In First Order Logic:

Propositions have more internal structure:

AliceLikesKT ⟹ (Alice,likes, KT)

Quantifiers and variables: *all*, *some* quantifiers apply to variables. *All* applies to each student/integer we don't want to enumerate them all

First Order Logic

First-order logic can be understood as an extension of propositional logic.

In **propositional logic** the atomic formulas have **no internal structure**.

They are propositional variables that are either true or false.

In first-order logic:

- atomic formulas are predicates that assert a relationship among certain elements.
- quantification: the ability to assert that a certain property holds for all elements or that it holds for some element.

First Order Logic –Examples

Alice and Bob like KT.

$$likes(alice,kt) \land likes(bob,kt)$$

***All** students like KT.*

$$\forall x \ Student\ (x) \rightarrow likes(x,KT)$$

Syntax

First order Logic is composed of:

Connectives:

¬ (negation), ∧ (conjunction), ∨ (disjunction), ⇒ (implication), ⇔(biconditional)

Parenthesis: (,)

An infinite set of variables (Vars): x_0, x_1, x_3,...

Quantifier symbols: ∃ (existential quantifier), ∀ (universal quantifier)

Equality symbol: =

Syntax

Constant symbols: John, Mary, 5, 6, Ball123, . . .

Function symbols: FatherOf(·), BestFriendOf(·), Cosine(·), Sum(·, ·),

Predicate symbols: Happy(·), FatherOf(·, ·), Arrives(·, ·, ·)

Functions are different from predicates!

A function takes one or more arguments, and returns a value.

A predicate takes one or more arguments, and is either true or false.

Syntax

Term → ConstantSymbol | Variable | FunctionSymbol(Term, . . . , Term)

$$John$$

$$x$$

$$FatherOf(John)$$

$$WifeOf(FatherOf(x))$$

$$Sum(1, 2)$$

$$Sum(x, Sum(1, 2))$$

Terms are used to represent the objects of our world

Syntax

AtomicFormula → Term = Term | PredicateSymbol(Term, . . . , Term)

$$John = ElderSonOf(Mary)$$

$$John = ElderSonOf(FatherOf(John))$$

$$Sum(1, Sum(2, 3)) = 6$$

$$Happy(John)$$

$$EvenNumber(Sum(1, Sum(2, 3)))$$

$$LivesIn(John, London)$$

$$Arrives(John, Athens, Monday)$$

Atomic Formulas are used to represent facts or simple relationships between the objects of our world

Syntax

Well-formed formulas

Wff → AtomicFormula | (Wff) | ¬Wff | Wff BinaryConnective Wff |

(Quantifier Variable) Wff

WFF Examples

$$\neg Loves(Tony, Mary)$$

$$Loves(Tony, Paula) \lor Loves(Tony, Fiona)$$

$$Loves(John, Paula) \land Loves(John, Fiona)$$

$$(\forall x)(SportsCar(x) \land HasDriven(Mike, x) \Rightarrow Likes(Mike, x))$$

$$(\exists x)(SportsCar(x) \land Owns(John, x))$$

$$\forall x)(Greek(x) \land (\forall y)(IsChildOf(y, x) \Rightarrow UniversityAlumni(y)) \Rightarrow Proud(x))$$

Syntax

A free variable is a variable that is not bound by an quantifier,

e.g. $\exists y$ Likes(x,y): x is free, y is bound.

A **sentence** is a well-formed formula in which all variables are quantified (no free variable)

Example:

Sentence: $(\forall x)(Cat(x) \Rightarrow Mammal(x))$

Not a sentence: Brother(x, John)

Natural Language to FOL

The world is composed of:

objects (Mary, John, vase, desk)

Objects participate in **relations** with other objects . Some of these **relations** are functions.

Examples:

John loves Mary

John is father of Peter

The relations between objects can be **true** or **false.**

Natural Language to FOL

Nouns describe the entities of the domain of interest:

Mary, John, vase

Nouns can be represented with **constants** in FOL

Verbs are used to express an action, an occurrence, or a state of being

John loves Mary

Jack broke the vase

John is sleeping

Verbs can be replaced with **predicates** in FOL

Natural Language to FOL

Nouns describe the entities of the domain of interest:

Mary, John, vase

Nouns can be represented with **constants** in FOL

Verbs are used to express an action, an occurrence, or a state of being

John loves Mary	loves(John, Mary)
Jack broke the vase	broke(Jack,vase)
John is sleeping	Asleep(John)

Verbs can be replaced with **predicates** in FOL

Natural Language to FOL

John is father of Peter (relationship between two objects of the world)

John = fatherOf(Peter)

Here we have two nouns, the relation between them, which is represented with a function and the “is” which is expressed with the “=”.

We can also represent it with

isFatherOf(John,Peter)

But it is less accurate.

Function or Predicate? *If we want to use the formula for inference then predicate representation is preferred.*

Natural Language to FOL

Adjectives are words that describe the qualities or states of being of nouns.

Jack is a clever dog.

Adjectives in FOL can be expressed with unary predicates:

Clever(Jack) $\wedge$ Dog(Jack)

NOT: Azor = Clever(Dog) : “Clever” is a predicate.

If an object belongs to a set of objects then this is represented with a unary predicate: Country(Greece)

Natural Language to FOL

Greece and Bulgaria are countries:

Country(Greece) ∧ Country(Bulgaria).

NOT: Country(Greece ∧ Bulgaria)

Observations

A sentence of the form “if….then…” implies the use of “$\rightarrow$”.

A sentence including “...if and only if…” implies the use of “$\Leftrightarrow$”

A sentence including “for all..” implies $\forall$

A sentence including “there exists..” implies $\exists$

Rule of thumb:

The “$\forall$” is combined with “$\rightarrow$”

The “$\exists$” is combined with “$\land$”

Example

Every king is a human

Example

Every king is a human

i.e. **if** someone is a king **then** is human

Example

Every king is a human

i.e. **if** someone is a king **then** is human

i.e. **for every** x **if** x is a king **then** x is a human

Example

Every king is a human

i.e. **if** someone is a king **then** is human

i.e. **for every** x **if** x is a king **then** x is a human

$$(\forall x)(King(x) \Rightarrow Human(x))$$

Quantifiers

The order of quantifiers of the same type is not important. The following formulas are equivalent:

$$(\forall x)((\forall y)Loves(x, y))$$

$$(\forall y)((\forall x)Loves(x, y))$$

$$(\forall x)(\forall y)Loves(x, y)$$

$$(\forall y)(\forall x)Loves(x, y)$$

Quantifiers

We can group them together:

$$(\forall x, y) Loves(x, y)$$

$$(\forall y, x) Loves(x, y)$$

Quantifiers

The order of quantifiers of **different** type is important. The following formulas are **not** equivalent:

$$(\forall x)(\exists y)Loves(x, y)$$

$$(\exists x)(\forall y)Loves(x, y)$$

Semantics

In the **propositional logic**, an **interpretation** is an assignment of **truth values to atoms**.

In the **first-order logic**, since there are **variables involved**, we have to do more than that.

To define an interpretation for a formula in the first- order logic, we have to specify:
–**the domain**

–an **assignment** to constants, function symbols, and predicate symbols occurring in the formula

Semantics (informal definition)

An interpretation of a formula *F* in the first-order logic consists of: a nonempty **domain** $\mathcal{D}$, and an **interpretation** function ***I*** that assigns “values” to each constant, function symbol, and predicate symbol occurring in *F* as follows:

To each constant, we assign an element in $\mathcal{D}$.

To each n-place function symbol, we assign a mapping from $\mathcal{D}^n$ to $\mathcal{D}$. ($\mathcal{D}^n = \{(x_1,\ldots,x_n) | x_1,\ldots,x_n \in \mathcal{D}\}$).

To each n-place predicate symbol, we assign a mapping from $\mathcal{D}^n$ to {T,F}.

–When we evaluate the truth value of a formula in an interpretation over the domain $\mathcal{D}$, ($\forall$ x) will be interpreted as “for all elements x in $\mathcal{D}$” and ($\exists$x) as “there is an element in $\mathcal{D}$.”

Semantics (informal definition)

For every interpretation of a formula over a domain $\mathcal{D}$, the formula can be evaluated to T or F according to the following rules:

–If the truth values of formulas *G* and *H* are evaluated, then the truth values of the formulas ~*G*, (*G* binaryConnective *H*) are evaluated by using the rules that hold in the propositional logic.

–($\forall$x)*G* is evaluated to T if the truth value of *G* is evaluated to T for every x in $\mathcal{D}$; otherwise, it is evaluated to F.

–($\exists$x)*G* is evaluated to T if the truth value of *G* is T for at least one x in $\mathcal{D}$; otherwise, it is evaluated to F.

Semantics

Any formula containing free variables cannot be evaluated.

It should be assumed either that formulas do not contain free variables, or that free variables are treated as constants.

Example

G= $(\forall x)(P(x) \rightarrow Q(f(x),a))$

The interpretation I of G:

Domain: D={1,2}

Assignment for a: a=1

Assignment for f: f(1)=2, f(2)=1

Assignment for P, Q:

P(1)=F, P(2)=T, Q(1,1)=T,

Q(1,2)=T, Q(2,l)=F, Q(2,2)=T

Example

G= $(\forall x)(P(x) \rightarrow Q(f(x),a))$

The interpretation I of G:

Domain: D={1,2}

Assignment for a: a=1

Assignment for f: f(1)=2, f(2)=1

Assignment for P, Q:

P(1)=F, P(2)=T, Q(1,1)=T,

Q(1,2)=T, Q(2,l)=F, Q(2,2)=T

If x=1, then:

$P(x) \rightarrow Q(f(x),a)$

$P(1) \rightarrow Q(f(1),a)$

$P(1) \rightarrow Q(2,1)$

$F \rightarrow F$

T.

If x=2, then:

$P(x) \rightarrow Q(f(x),a)$

$P(2) \rightarrow Q(f(2),a)$

$P(2) \rightarrow Q(1,1)$

$T \rightarrow T$

T

Example

G= $(\forall x)(P(x) \to Q(f(x),a))$

The interpretation **I** of G:

Domain: $\mathcal{D}$={1,2}

Assignment for a: a=1

Assignment for f: f(1)=2, f(2)=1

Assignment for P, Q:

P(1)=F, P(2)=T, Q(1,1)=T,

Q(1,2)=T, Q(2,l)=F, Q(2,2)=T

If x=1, then:

$P(x) \to Q(f(x),a)$

$P(1) \to Q(f(1),a)$

$P(1) \to Q(2,1)$

$F \to F$

T.

If x=2, then:

$P(x) \to Q(f(x),a)$

$P(2) \to Q(f(2),a)$

$P(2) \to Q(1,1)$

$T \to T$

T

Since $P(x) \to Q(f(x),a)$ is true for all elements x in $\mathcal{D}$,
then G is true under the interpretation **I**

Example

Every king is a human.

Why not $(\forall x)(King(x) \land Human(x))$?

When does this become true?

Let D={eleni, george}

Equivalences in FOL

$$(\forall x)\phi \equiv \neg(\exists x)\neg\phi$$

$$(\exists x)\phi \equiv \neg(\forall x)\neg\phi$$

$$(\forall x)\neg\phi \equiv \neg(\exists x)\phi$$

$$(\exists x)\neg\phi \equiv \neg(\forall x)\phi$$

$$(\exists x)(\phi \vee \psi) \equiv (\exists x)\phi \vee (\exists x)\psi$$

$$(\exists x)(\phi \wedge \psi) \models (\exists x)\phi \wedge (\exists x)\psi$$

$$(\forall x)\phi \vee (\forall x)\psi \models (\forall x)(\phi \vee \psi)$$

$$(\forall x)(\phi \wedge \psi) \equiv (\forall x)\phi \wedge (\forall x)\psi$$

Resources

Γ. Κολέτσος, Σημειώσεις Μαθηματικής Λογικής.

Lecture 16: Logic 1 - Propositional Logic | Stanford CS221: AI (Autumn 2019)

Lecture 17: Logic 2 - First-order Logic | Stanford CS221: AI (Autumn 2019)

Propositional Logic: Formulas, Models, Tableaux

Λογική Πρώτης Τάξης, Διαφάνειες Μαθήματος, Μ. Κουμπαράκης. (

https://opencourses.uoa.gr/modules/document/file.php/DI115/%CE%94%CE%B9%CE%B1%CF%86%CE%AC%CE%BD%CE%B5%CE%B9%CE%B5%CF%82/fol-syntax1spp.pdf

http://cgi.di.uoa.gr/~ys02/lectures/fol-semantics1spp.pdf)

Common Mistakes in FOL