First Order Logic

Er. Rudra Nepal

Logic in general

- Logics are formal languages for representing information such that conclusions can be drawn
- Syntax defines the sentences in the language
- Semantics define the “meaning” of sentences

Introduction to First-Order Logic

- A formal logic generated by combining predicate logic and propositional logic.
  - Propositional logic is used to assert propositions, which are statements that are either true or false. It deals only with the truth value of complete statements and does not consider relationships or dependencies between objects.
  - Predicate logic is an extension and generalization of propositional logic. Its formulas contain variables which can be quantified.

Propositional logic

- **Logical constants**: true, false
- **Propositional symbols**: P, Q, S, ... (**atomic sentences**)
- Wrapping **parentheses**: (…)
- Sentences are combined by **connectives**:

$\wedge$	...and	[conjunction]
$\vee$	...or	[disjunction]
$\Rightarrow$	...implies	[implication / conditional]
$\Leftrightarrow$	..is equivalent	[biconditional]
$\neg$	...not	[negation]

- **Literal**: atomic sentence or negated atomic sentence

Examples of PL sentences

- P means “It is hot.”
- Q means “It is humid.”
- R means “It is raining.”
- $(P \wedge Q) \rightarrow R$
  - “If it is hot and humid, then it is raining”
- $Q \rightarrow P$
  - “If it is humid, then it is hot”

Propositional logic (PL)

- A simple language useful for showing key ideas and definitions
- User defines a set of propositional symbols, like P and Q.
- User defines the **semantics** of each propositional symbol:
  - P means "It is hot"
  - Q means "It is humid"
  - R means "It is raining"
- A sentence (well formed formula) is defined as follows:
  - A symbol is a sentence
  - If S is a sentence, then $\neg S$ is a sentence
  - If S is a sentence, then (S) is a sentence
  - If S and T are sentences, then $(S \vee T)$, $(S \wedge T)$, $(S \rightarrow T)$, and $(S \leftrightarrow T)$ are sentences
  - A sentence results from a finite number of applications of the above rules

Truth tables for connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

Some terms

- The meaning or **semantics** of a sentence determines its **interpretation**.
- Given the truth values of all symbols in a sentence, it can be "evaluated" to determine its **truth value** (True or False).
- A **model** for a KB is a "possible world" (assignment of truth values to propositional symbols) in which each sentence in the KB is True.

More terms

- A **valid sentence** or **tautology** is a sentence that is True under all interpretations, no matter what the world is actually like or how the semantics are defined. Example: “It’s raining or it’s not raining.”
- An **inconsistent sentence** or **contradiction** is a sentence that is False under all interpretations. The world is never like what it describes, as in “It’s raining and it’s not raining.”
- **P entails Q**, written P |= Q, means that whenever P is True, so is Q. In other words, all models of P are also models of Q.

Inference rules

- **Logical inference** is used to create new sentences that logically follow from a given set of predicate calculus sentences (KB).
- An inference rule is **sound** if every sentence X produced by an inference rule operating on a KB logically follows from the KB. (That is, the inference rule does not create any contradictions)
- An inference rule is **complete** if it is able to produce every expression that logically follows from (is entailed by) the KB. (Note the analogy to complete search algorithms.)

Sound rules of inference

- Here are some examples of sound rules of inference
  - *A rule is sound if its conclusion is true whenever the premise is true*
- Each can be shown to be sound using a truth table

RULE	**PREMISE**	**CONCLUSION**
Modus Ponens	$A, A \rightarrow B$	B
And Introduction	A, B	$A \wedge B$
And Elimination	$A \wedge B$	A
Double Negation	$\neg\neg A$	A
Unit Resolution	$A \vee B, \neg B$	A
Resolution	$\mathbf{A \vee B, \neg B \vee C}$	$\mathbf{A \vee C}$

Proving things

- A **proof** is a sequence of sentences, where each sentence is either a premise or a sentence derived from earlier sentences in the proof by one of the rules of inference.
- The last sentence is the **theorem** (also called goal or query) that we want to prove.
- Example for the "weather problem" given above.

1 Humid	Premise	"It is humid"
2 Humid→Hot	Premise	"If it is humid, it is hot"
3 Hot	Modus Ponens(1,2)	"It is hot"
4 (Hot ∧ Humid)→Rain	Premise	"If it's hot & humid, it's raining"
5 Hot ∧ Humid	And Introduction(1,2)	"It is hot and humid"
6 Rain	Modus Ponens(4,5)	"It is raining"

Horn sentences

- A **Horn sentence** or **Horn clause** has the form:

  $P1 \wedge P2 \wedge P3 \ldots \wedge Pn \rightarrow Q$

or alternatively

$(P \rightarrow Q) = (\neg P \vee Q)$

  $\neg P1 \vee \neg P2 \vee \neg P3 \ldots \vee \neg Pn \vee Q$

where Ps and Q are non-negated atoms

- To get a proof for Horn sentences, apply Modus Ponens repeatedly until nothing can be done
- We will use the Horn clause form later

Entailment and derivation

- **Entailment: KB |= Q**
  - Q is entailed by KB (a set of premises or assumptions) if and only if there is no logically possible world in which Q is false while all the premises in KB are true.
  - Or, stated positively, Q is entailed by KB if and only if the conclusion is true in every logically possible world in which all the premises in KB are true.
- **Derivation: KB |- Q**
  - We can derive Q from KB if there is a proof consisting of a sequence of valid inference steps starting from the premises in KB and resulting in Q

Two important properties for inference

Soundness: If KB |- Q then KB |= Q

- If Q is derived from a set of sentences KB using a given set of rules of inference, then Q is entailed by KB.
- Hence, inference produces only real entailments, or any sentence that follows deductively from the premises is valid.

Completeness: If KB |= Q then KB |- Q

- If Q is entailed by a set of sentences KB, then Q can be derived from KB using the rules of inference.
- Hence, inference produces all entailments, or all valid sentences can be proved from the premises.

Propositional logic is a weak language

- Hard to identify "individuals" (e.g., Mary, 3)
- Can't directly talk about properties of individuals or relations between individuals (e.g., "Bill is tall")
- Generalizations, patterns, regularities can't easily be represented (e.g., "all triangles have 3 sides")
- First-Order Logic (abbreviated FOL or FOPC) is expressive enough to concisely represent this kind of information

  FOL adds relations, variables, and quantifiers, e.g.,

  - *"Every elephant is gray":* $\forall$ x (elephant(x) $\rightarrow$ gray(x))
  - *"There is a white alligator":* $\exists$ x (alligator(X) ^ white(X))

Example

- Consider the problem of representing the following information:
  - Every person is mortal.
  - Confucius is a person.
  - Confucius is mortal.
- How can these sentences be represented so that we can infer the third sentence from the first two?

Example II

- In PL we have to create propositional symbols to stand for all or part of each sentence. For example, we might have:
  P = “person”; Q = “mortal”; R = “Confucius”
- so the above 3 sentences are represented as:
  $P \rightarrow Q$; $R \rightarrow P$; $R \rightarrow Q$
- Although the third sentence is entailed by the first two, we needed an explicit symbol, R, to represent an individual, Confucius, who is a member of the classes “person” and “mortal”
- To represent other individuals we must introduce separate symbols for each one, with some way to represent the fact that all individuals who are “people” are also “mortal”

First-order logic

- Whereas propositional logic assumes the world contains facts,
- first-order logic (like natural language) assumes the world contains
  - **Objects**: people, houses, numbers, colors, baseball games, wars, …
  - **Relations**: red, round, prime, brother of, bigger than, part of, comes between, …
  - **Functions**: father of, best friend, one more than, plus, …

Syntax of FOL: Basic elements

• Constants	KingJohn, 2, IOE,...
• Predicates	Brother, >,...
• Functions	Sqrt, LeftLegOf,...
• Variables	x, y, a, b,...
• Connectives	$\neg$, $\Rightarrow$, $\wedge$, $\vee$, $\Leftrightarrow$
• Equality	=
• Quantifiers	$\forall$, $\exists$

Atomic sentences

Atomic sentence = *predicate* ($term_1$,...,$term_n$)
or $term_1 = term_2$

Term = *function* ($term_1$,...,$term_n$)
or *constant* or *variable*

- E.g., *Brother(KingJohn,RichardTheLionheart)*
- *> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))*

Complex sentences

- Complex sentences are made from atomic sentences using connectives

  $\neg S$, $S_1 \wedge S_2$, $S_1 \vee S_2$, $S_1 \Rightarrow S_2$, $S_1 \Leftrightarrow S_2$,

E.g. *Sibling(KingJohn,Richard)* $\Rightarrow$ *Sibling(Richard,KingJohn)*

$>(1,2) \vee \leq(1,2)$

$>(1,2) \wedge \neg >(1,2)$

Truth in first-order logic

- Sentences are true with respect to a model and an interpretation
- Model contains objects (domain elements) and relations among them
- Interpretation specifies referents for

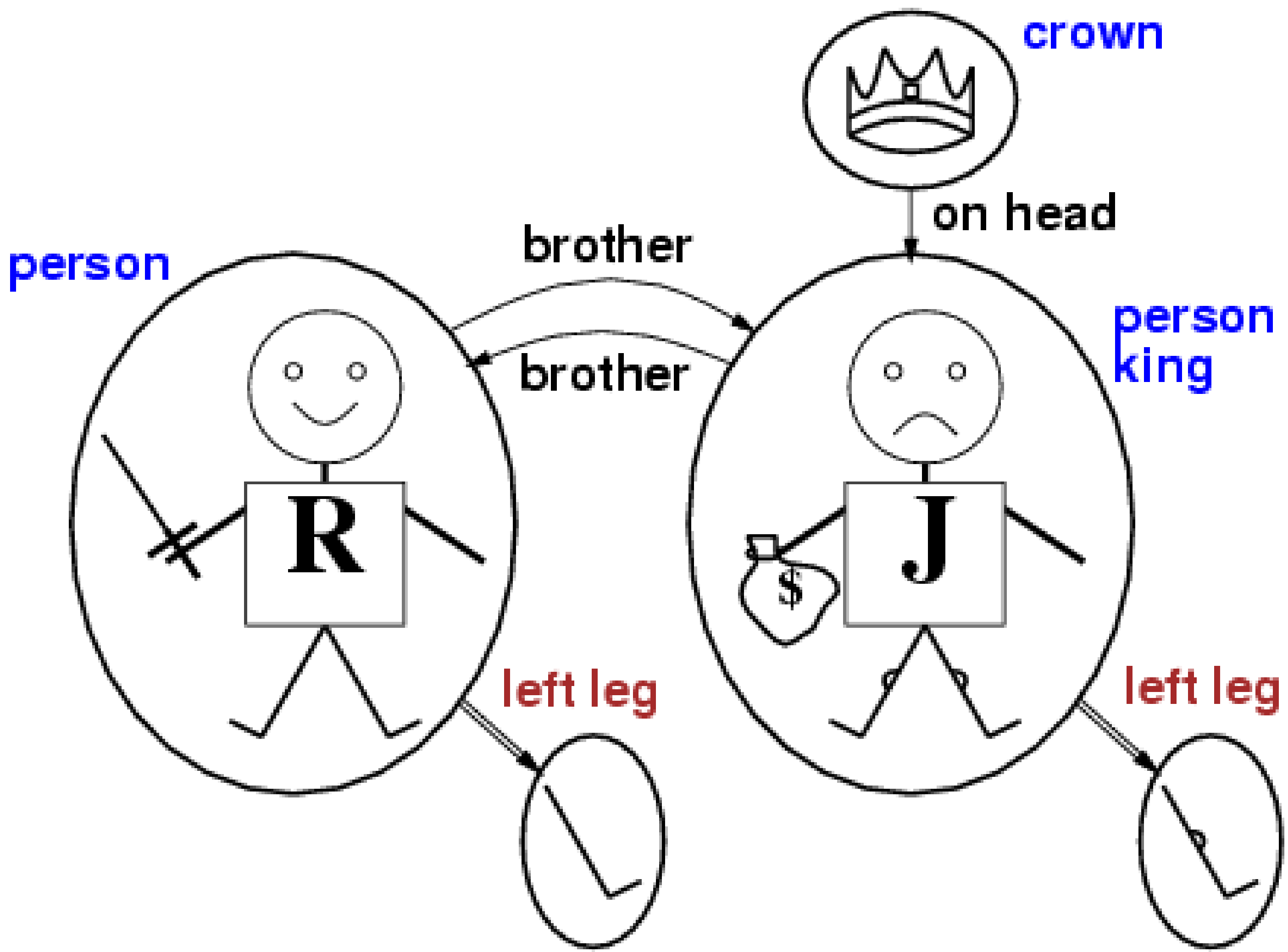
  constant symbols → objects

  predicate symbols → relations

  function symbols → functional relations

- An atomic sentence *predicate*($term_1,\ldots,term_n$) is true iff the objects referred to by $term_1,\ldots,term_n$ are in the relation referred to by *predicate*

Models for FOL: Example



Universal quantification

- $\forall$<*variables*> <*sentence*>

Everyone at IOE is smart:

$\forall x\ At(x, IOE) \Rightarrow Smart(x)$

- $\forall x\ P$ is true in a model m iff P is true with x being each possible object in the model

- Roughly speaking, equivalent to the conjunction of instantiations of P

$$
\begin{aligned}
& At(KingJohn,IOE) \Rightarrow Smart(KingJohn) \\
\wedge\ & At(Richard,IOE) \Rightarrow Smart(Richard) \\
\wedge\ & At(IOE,IOE) \Rightarrow Smart(IOE) \\
\wedge\ & \ldots
\end{aligned}
$$

A common mistake to avoid

- Typically, $\Rightarrow$ is the main connective with $\forall$
- Common mistake: using $\wedge$ as the main connective with $\forall$:
  $\forall x\ At(x, IOE) \wedge Smart(x)$
  means "Everyone is at IOE and everyone is smart"

Existential quantification

- $\exists$<*variables*> <*sentence*>
- Someone at IOE is smart:
  $\exists x$ At(x,IOE) $\wedge$ Smart(x)
- $\exists x$ *P* is true in a model *m* iff *P* is true with *x* being some possible object in the model
- Roughly speaking, equivalent to the disjunction of instantiations of *P*

  At(KingJohn,IOE) $\wedge$ Smart(KingJohn)
  $\vee$ At(Richard,IOE) $\wedge$ Smart(Richard)
  $\vee$ At(IOE,IOE) $\wedge$ Smart(IOE)
  $\vee$ ...

Another common mistake to avoid

- Typically, $\wedge$ is the main connective with $\exists$
- Common mistake: using $\Rightarrow$ as the main connective with $\exists$:

$$\exists x \; \mathrm{At}(x,\mathrm{IOE}) \Rightarrow \mathrm{Smart}(x)$$

is true if there is anyone who is not at IOE!

Translating English to FOL

Every gardener likes the sun.

$\forall x\ gardener(x) \rightarrow likes(x,Sun)$

You can fool some of the people all of the time.

$\exists x\ \forall t\ \ person(x)\ \wedge time(t) \rightarrow can\text{-}fool(x,t)$

You can fool all of the people some of the time.

$\forall x\ \exists t\ (person(x) \rightarrow time(t)\ \wedge can\text{-}fool(x,t))$

$\forall x\ (person(x) \rightarrow \exists t\ (time(t)\ \wedge can\text{-}fool(x,t))$

Equivalent

All purple mushrooms are poisonous.

$\forall x\ (mushroom(x)\ \wedge\ purple(x)) \rightarrow poisonous(x)$

No purple mushroom is poisonous.

$\neg\exists x\ purple(x)\ \wedge\ mushroom(x)\ \wedge\ poisonous(x)$

$\forall x\ \ (mushroom(x)\ \wedge\ purple(x)) \rightarrow \neg poisonous(x)$

Equivalent

There are exactly two purple mushrooms.

$\exists x\ \exists y\ mushroom(x)\ \wedge\ purple(x)\ \wedge\ mushroom(y)\ \wedge\ purple(y)\ \hat{}\ \neg(x=y)\ \wedge\ \forall z\ (mushroom(z)\ \wedge\ purple(z)) \rightarrow ((x=z)\ \vee\ (y=z))$

Clinton is not tall.

$\neg tall(Clinton)$

X is above Y iff X is on directly on top of Y or there is a pile of one or more other objects directly on top of one another starting with X and ending with Y.

$\forall x\ \forall y\ above(x,y) \leftrightarrow (on(x,y)\ \vee\ \ \exists z\ (on(x,z)\ \wedge\ above(z,y)))$

Properties of quantifiers

- $\forall x \; \forall y$ is the same as $\forall y \; \forall x$
- $\exists x \; \exists y$ is the same as $\exists y \; \exists x$
- $\exists x \; \forall y$ is not the same as $\forall y \; \exists x$

- $\exists x \; \forall y$ Loves(x,y)
  - "There is a person who loves everyone in the world"

- $\forall y \; \exists x$ Loves(x,y)
  - "Everyone in the world is loved by at least one person"

- Quantifier duality: each can be expressed using the other
- $\forall x$ Likes(x,IceCream) $\quad \neg\exists x \; \neg$Likes(x,IceCream)
- $\exists x$ Likes(x,Broccoli) $\quad \neg\forall x \; \neg$Likes(x,Broccoli)

Equality

- $term_1 = term_2$ is true under a given interpretation if and only if $term_1$ and $term_2$ refer to the same object

- E.g., definition of *Sibling* in terms of *Parent*:

  $\forall x,y\ Sibling(x,y) \Leftrightarrow [\neg(x = y) \land \exists m,f\ \neg (m = f) \land Parent(m,x) \land Parent(f,x) \land Parent(m,y) \land Parent(f,y)]$