Introduction to Artificial Intelligence: Knowledge Representation

Dr. Tapas Kumar Mishra

Assistant Professor
Department of Computer Science and Engineering
SRM University
Amaravati-522240, Andhra Pradesh, India
Email: tapaskumar.m [at] srmap [dot] edu [dot] in

- **First order predicate logic** (FOPL) or **Predicate calculus** - One of the oldest and most important representation scheme
- **Logic** is a formal method for reasoning.
- In FOPL, statements from a **natural language** like English are translated into **symbolic structures** comprised of **predicates**, **functions**, **variables**, **constants**, **quantifiers** and **logical connectives**.
- **Inference rules** may then be applied to compare, combine and transform these **assumed** structures into new **deduced** structures.

Introduction to KR: Formalized Symbolic Logics

- **Example:** All employees of the AI-software company are programmers
- **FOPL:** $(\forall x)$ (AI-SOFTWARE-CO-EMPLOYEE(x) $\rightarrow$ PROGRAMMER(x))
- **Predicate:** AI-SOFTWARE-CO-EMPLOYEE(x), PROGRAMMER(x)
- "if x is an AI software company employee" and "x is a programmer"
- **Variable:** x
- If it is known that Jim is an employee of AI software company
  - AI-SOFTWARE-CO-EMPLOYEE(jim)
- One can draw the conclusion that Jim is a programmer
  - PROGRAMMER(jim)

Syntax and semantics for propositional logic

- **Propositions** are elementary atomic sentences/formulas/well-formed formulas.
- **Propositions** may be either **true** or **false** but may take on no other value.
- **Example:**
  - It is raining.
  - My car is painted silver.
  - John and Sue have five children.
  - Snow is white.
  - People live on the moon.

- **Compound propositions** are formed from atomic formulas using the logical connectives, **not**, **or**, **if** ... **then**, **and** & **if and only if**.
- **Example:**
    - It is raining and the wind is blowing.
    - The moon is made of green cheese or it is not.
    - If you study hard you will be rewarded.
    - The sum of 10 and 20 is not 50.

Syntax and semantics for propositional logic

- **Proposition** - capital letter, sometimes followed by digits
- T - true
- F - false
- $\sim$ - not or negation
- & - and or conjunction
- V - or or disjunction
- $\rightarrow$ - if . . . then or implication
- $\leftrightarrow$ - if and only if or double implication
- **Punctuation** - (,), {, }, period

Syntax and semantics for propositional logic

- It is raining and the wind is blowing.
  - $(R \& B)$
  - where R and B stand for the propositions "It is raining" and "the wind is blowing".
- It is raining or the wind is blowing or both.
  - $(R \vee B)$
  - V - Inclusive disjunction

Syntax and semantics for propositional logic - Syntax

- If P and Q are formulas, the following are formulas.
  - $(\sim P)$, $(P \;\&\; Q)$, $(P \vee Q)$, $(P \rightarrow Q)$, $(P \leftrightarrow Q)$
- **Compound formula:** $(P \;\&\; (\sim Q \vee R) \rightarrow (Q \rightarrow S))$
- **Precedence:** $\sim$, &, V, $\rightarrow$, $\leftrightarrow$
- Example:
  - $P \;\&\; \sim Q \vee R \rightarrow S \leftrightarrow U \vee W$
  - $((((P \;\&\; \sim(Q)) \vee R) \rightarrow S) \leftrightarrow (U \vee W))$

Syntax and semantics for propositional logic - Semantics

- The **semantics** or **meaning** of a sentence is just the value true or false; that is, it is an assignment of a truth value to the sentence.
- An **interpretation** for a sentence or group of sentences is an assignment of a truth value to each propositional symbol.
- **Example:**
  - $(P \ \& \ {\sim}Q)$
  - Interpretation (I_1): True to P and false to Q
  - Interpretation (I_2): True to P and true to Q
  - There are **four distinct interpretations** for this sentence.

Rule No.	True Statements	False Statements
1	T	F
2	$\sim f$	$\sim t$
3	t & t' (t' is a true statement)	f & a
4	t V a	a & f
5	a V t	f V f' (f' is a false statement)
6	$a \rightarrow t$	$t \rightarrow f$
7	$f \rightarrow a$	$t \leftrightarrow f$
8	$t \leftrightarrow t'$	$f \leftrightarrow t$
9	$f \leftrightarrow f'$	$p \Leftrightarrow q$: $\mathbf{(}p \Rightarrow q\mathbf{)}$ **&** $\mathbf{(}q \Rightarrow p\mathbf{)}$

- $((P$ & $\sim Q) \rightarrow R)$ V Q
- $((P$ & $\sim Q) \rightarrow R)$ V $Q \Rightarrow$ F
- Interpretation I: Assign true to P, false to Q and false to R
- Rules: 2 ($\sim Q$ as true), 3 ((P & $\sim Q$) as true), 6 ((P & $\sim Q$) $\rightarrow R$ as false) and 5 (Q as false)

Properties of Statements

- **Satisfiable:** A statement is satisfiable if there is **some** interpretation for which it is true.
- **Contradiction:** A sentence is contradictory (unsatisfiable) if there is **no** interpretation for which it is true.
- **Valid/Tautologies:** A sentence is valid if it is true for **every** interpretation.
- **Equivalence:** Two sentences are equivalent if they have the same truth value under **every** interpretation.
- **Logical consequences:** A sentence is a logical consequence of another if it is satisfied by **all** interpretations which satisfy the first.
- A **valid** statement is **satisfiable**, but the **converse** is not necessarily true. b

Properties of Statements

- $P \vee \sim P$ is valid.
- $P \;\&\; \sim P$ is contradiction.
- P is satisfiable but not valid.
- P and $\sim(\sim P)$ are equivalent.
- P is a logical consequence of $(P \;\&\; Q)$ since any interpretation for which $(P \;\&\; Q)$ is true, P is also true.

Properties of Statements

Theorem 1

The sentence s is a logical consequence of $s_1, \ldots, s_n$ if and only if s_1 & s_2 & $\ldots$ $s_n \rightarrow s$ is valid.

Theorem 2

The sentence s is a logical consequence of $s_1, \ldots, s_n$ if and only if s_1 & s_2 & $\ldots$ & s_n **&** $\sim s$ is inconsistent.

Proof

$\sim(s_1 \ \& \ s_2 \ \& \ \ldots \ \& \ s_n \rightarrow s)$
$= \sim (\sim(s_1 \ \& \ s_2 \ \& \ \ldots \ s_n) \ \mathrm{V} \ s)$
$= \sim \sim (s_1 \ \& \ s_2 \ \& \ \ldots \ s_n) \ \mathrm{V} \ \sim s$
$= s_1 \ \& \ s_2 \ \& \ \ldots \ s_n \ \mathrm{V} \ \sim s$

Properties of Statements

- When s is a logical consequence of $s_1, \ldots, s_n$, the formula s_1 & s_2 & ... & $s_n \rightarrow s$ is called a **theorem**, with s the **conclusion**.
- Let $S = \{s_1, \ldots, s_n\}$
- S logically implies or logically entails s, written as $S \models s$

Some Equivalence Laws

Idempotency	$P \vee P = P$
	$P \,\&\, P = P$
Associativity	$(P \vee Q) \vee R = P \vee (Q \vee R)$
	$(P \,\&\, Q) \,\&\, R = P \,\&\, (Q \,\&\, R)$
Commutativity	$P \vee Q = Q \vee P$
	$P \,\&\, Q = Q \,\&\, P$
	$P \leftrightarrow Q = Q \leftrightarrow P$
Distributivity	$P \,\&\, (Q \vee R) = (P \,\&\, Q) \vee (P \,\&\, R)$
	$P \vee (Q \,\&\, R) = (P \vee Q) \,\&\, (P \vee R)$
De Morgan's Laws	$\sim(P \vee Q) = \sim P \,\&\, \sim Q$
	$\sim(P \,\&\, Q) = \sim P \vee \sim Q$
Conditional Elimination	$P \rightarrow Q = \sim P \vee Q$
Bi-conditional Elimination	$P \leftrightarrow Q = (P \rightarrow Q) \,\&\, (Q \rightarrow P)$

Truth Table for Equivalent Sentences

P	Q	$\sim P$	$(\sim P \vee Q)$	$P \rightarrow Q$	$Q \rightarrow P$	$(P \rightarrow Q)$ & $(Q \rightarrow P)$
true	true	false	true	true	true	true
true	false	false	false	false	true	false
false	true	true	true	true	false	false
false	false	true	true	true	true	true

Inference Rules

- The problem is given a set of sentences $S = \{s_1, \ldots, s_n\}$ (the premises), prove that the truth of s (the conclusion); that is, show that $S \models s$.

Modus Ponens

P
$P \to Q$

Q

Example

Joe is a father.
Joe is a father. $\to$ Joe has a child.

Joe has a child.

Inference Rules

Chain Rule

$P \rightarrow Q$
$Q \rightarrow R$

$P \rightarrow R$

Chain Rule

Programmer likes LISP $\rightarrow$ Programmer hates COBOL
Programmer hates COBOL $\rightarrow$ Programmer likes recursion

---

Programmer likes LISP $\rightarrow$ Programmer likes recursion

Inference Rules

Substitution

If s is a valid sentence, s' derived from s by consistent substitution of propositions in s, is also valid.

Example

The sentence $P \vee \sim P$ is valid; therefore $Q \vee \sim Q$ is also valid.

Simplification

From P & Q infer P

Conjunction

From P and from Q, infer P & Q

Transposition

From $P \rightarrow Q$, infer $\sim Q \rightarrow \sim P$

Syntax and Semantics for FOPL

- PL does **not** permit us to make **generalized statements** about classes of similar objects.
- There are **serious** limitations when reasoning about real world entities.

Example

All students in computer science must take Pascal.
John is a computer science major.

It is **not possible** to conclude in PL that **John must take Pascal** since the second statement does not occur as part of the first one.

- **FOPL** was developed by logicians to extend the **expressiveness** of **PL**.

Syntax of FOPL

- **Connectives** (five symbols): $\sim$, &, V, $\rightarrow$ and $\leftrightarrow$
- **Quantifiers** (two symbols): $\exists$ (Existential quantification) and $\forall$ (Universal quantification)
  - $(\forall x)$ $(\forall y)$ $(\forall z)$, we abbreviate with a single quantifier and drop the parentheses to get $\forall xyz$
- **Constants:** numbers, words and small letters near the beginning of the alphabet
  - a, b, c, 5.3, -21, flight-102 and john
- **Variables:** words and small letters near end of the alphabet
  - x, y and z

Syntax of FOPL

- **Function:** It denotes relations defined on a domain D.
- A 0-ary function is a constant.
  - **Example:** f, g, h, father-of and age-of
- **Predicates:** It denotes relations or functional mappings from the elements of a domain D to the values true or false.
- A 0-ary predicate is a proposition (a constant predicate).
  - **Example:** Capital letters and capitalized words such as P, Q, R, EQUAL and MARRIED
- **Term:** Constants, variables and functions
- **Atomic formulas/Atoms:** Predicates
- **Literal:** An atom or its negation
- **Punctuations:** (,), [,], {, } and period

Example

E1: All employees earning \$1400 or more per year pay taxes.
E2: Some employees are sick today.
E3: No employee earns more than the president.

$E(x)$ for x is an employee. (**NOTE:** upper case denotes a predicate)
$P(x)$ for x is president.
$i(x)$ for the income of x. (**NOTE:** lower case denotes a function)
$GE(u, v)$ for u is greater than or equal to v.
$S(x)$ for x is sick today.
$T(x)$ for x pays taxes.

E1': $\forall x \ ((E(x) \ \& \ GE(i(x), 1400)) \rightarrow T(x))$
E2': $\exists y \ (E(y) \rightarrow S(y))$
E3': $\forall xy \ ((E(x) \ \& \ P(y)) \rightarrow \sim GE(i(x), \ i(y)))$
The expressions E1', E2' and E3' are known as **well-formed formulas** (wffs).

Syntax of FOPL

- An atomic formula is a wff.
- If P and Q are wffs, then $\sim P$, P & Q, $P \vee Q$, $P \rightarrow Q$, $P \leftrightarrow Q$, $\forall x\ P(x)$ and $\exists x\ P(x)$ are wffs.

Valid wffs

MAN(john)
PILOT(father-of(bill))
$\exists xyz$ ((FATHER(x, y) & FATHER(y, z)) $\rightarrow$ GRANDFATHER(x, z))
$\forall x$ NUMBER(x) $\rightarrow$ ($\exists y$ GREATER-THAN(y, x))
$\forall x\ \exists y\ (P(x)$ & $Q(y)) \rightarrow (R(a) \vee Q(b))$

Not wffs

$\forall P\ P(x) \rightarrow Q(x)$ // Universal quantification is applied to the predicate $P(x)$
MAN($\sim$john) // Constant is negated
father-of($Q(x)$) // Function with a predicate argument
MARRIED(MAN, WOMAN) // Predicate with two predicate arguments

Semantics for FOPL

- A predicate (or wff) that has no variables is called a **ground atom**.

Example

E: $\forall x \; ((A(a, x) \vee B(f(x))) \; \& \; C(x)) \rightarrow D(x)$
Predicates: A, B, C and D
A is a **two-place predicate**, the first argument being the **constant** a and the second argument, a **variable** x.
B, C and D are all unary predicates.
The argument of B is a function $f(x)$ and the argument of C and D is the variable x.

Thank You!