Syntax and Semantics for FOPL

Prof. Dr. A K M Akhtar Hossain

Dept. of CSE , University of Rajshahi

FOPL (Fist Order Predicate Logic)

- In **first-order predicate logic**, a **predicate can** only refer to a single subject.
- **First-order logic** is also known as **first-order predicate calculus** or **first-order** functional **calculus**.
- A sentence in **first-order Predicate logic** is written in the form Px or P(x), where P is the **predicate** and x is the subject, represented as a variable.

FOPL (Fist Order Predicate Logic)

- FOPL was developed by logicians to extend the expressiveness of PL.
- The syntax for FOPL, like PL, is determined by the allowable symbols and rules of combination.
- **The semantics of FOPL are determined by interpretations assigned to predicates, rather than propositions.**

Syntax of FOPL

- **The symbols and rules of combination permitted in FOPL are defined as follows:**
- **Connectives:** There are five connective symbols:
    - ~(not or negation)
    - & (and or conjunction)
    - V (**or** or inclusive disjunction)
    - → (implication)
    - ↔ (equivalence or if and only if).

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Syntax of FOPL

- Quantifiers: The two quantifier symbols are ∃ (existential quantification) and ∀ (universal quantification).
- Where (∃**x**) means for some **x** or there is an **x** .

  and (∀**x**) means for **all x**.
- When more than one variable is being quantified by the same quantifier, such as, (∀**x**) (∀**y**) (∀**z**), we abbreviate with a single quantifier and drop the parentheses to get ∀**xyz**.

Syntax of FOPL

- **Constants:** Constants are **fixed-value** terms that belong to a given domain of discourse.
- They are denoted by **numbers**, **words**, and **small letters** near the beginning of the alphabet.
- **Examples:** a , b , c , 5.256, -67, -75.65 , flight-305, john, , Marina, etc.

Syntax of FOPL

- **Variables:** Variables are terms that can assume **different values** over a given domain.
- They are denoted by **words** and **small letters** near the end of the alphabet.
- **Examples:** aircraft-type, individuals, x, y, and z.

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Syntax of FOPL

- **Functions**: Function symbols denote **relations** defined on a domain **D**. They map ***n*** elements (***n≥0***) to a single element of the domain.
- Symbols *f* , *g*, & *h*, and words such as **father-of** , or **age-of**, represent functions.
- An ***n*** place (n-ary) function is written as $f(t_1, t_2, t_3, \ldots \;\ldots \ldots t_n)$ where the t_i are terms (constants, variables, or functions) defined over some domain. **A 0-ary function is a constant.**

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Syntax of FOPL

- **Predicates:** Predicate symbols denote relations or functional mappings from the elements of a domain **D** to the values **true** or **false**.
- Capital letters and capitalized words such as ***P***, ***Q***, ***R***, ***EQUAL***, and ***MARRIED*** are used to represent **predicates**.

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Syntax of FOPL

- Like functions, predicates may have ***n*** (***n* ≥ *0***) terms for arguments written as $P(t_1, t_2, t_3, \ldots \ldots \ldots, t_n)$
- Where the terms t_i , i =1 , 2, 3, ... , n are defined over some domain.
- **A *0-ary* predicate is a proposition**, that is, a **constant predicate**.

Syntax of FOPL

- **Constants**, **variables**, and **functions** are referred to as **terms**, and predicates are referred to as **atomic formulas** or **atoms** for short.

Syntax of FOPL

- **Examples:**
- E1: All employees earning Tk. 300000 or more per year pay taxes.
- E2: Some employees are sick today.
- E3: No employee earns more than the president.

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Syntax of FOPL

- To represent such **expressions FOPL**, we must define abbreviations for the predicates and functions.
- E(x) for x is an employee.
- P(y) for y is president.
- i(x) for the income of x (lower case denotes a function).
- GE(x, y) for x is greater than or equal to y.
- S(x) for x is sick today.
- T(x) for x pays taxes.

Syntax of FOPL

- Using the above abbreviations, we represent E1, E2, and E3 as:
- **E1**: $\forall x\ ((E(x)\ \&\ GE(i(x),300000))\ \rightarrow T(x))$
- **E2**: $\exists y\ (E(y)\ \rightarrow S(y))$
- **E3**: $\forall xy\ ((E(x)\ \&\ P(y))\ \rightarrow\ \sim GE(i(x),i(y)))$

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Syntax of FOPL

- An atomic formula is a **wffs** (**well-formed formulas**) .
- If P and Q are wffs, then ~P, P & Q, P V Q, P

  P↔Q, ∀x P(x), and ∃x P(x) are wffs.
- Wffs are formed only by applying the above rules a finite number of times.
- **The above rules state that all wffs are formed from atomic formulas and the proper application of quantifiers and logical connections.**

Syntax of FOPL

- **Some examples of valid wffs are**
- 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))

Syntax of FOPL

- **Some examples of statements that are not wffs are:**
- **$\forall$P P(x)→Q(x)**
- /* Universal quantification is applied to the predicate P(x). This is **invalid** in FOPL. */
- **MAN(~john)**
- /*The expression is **invalid** since the term John, a constant, is negated. */
- **father-of(Q(x))**
- /* The expression is **invalid** due to it is function with a predicate argument. */
- **MARRIED(MAN,WOMAN)**
- /* The expression **fails** since it is predicate with two predicate arguments.*/

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Semantics for FOPL

- When considering specific wffs, we always have in mind some **domain D**. If not stated explicitly, D will be understood from the context.
- The arguments predicates must be terms (**constant, variables or functions**). Therefore, the domain of each n-place predicate is also defined over D.

Semantics for FOPL

- **For Example**, our domain might be **all entities** that make up the Computer Science & Engineering Department at the University of Rajshahi.
- In this case, **constants** would be Professors (Dell, Cooke, Gelfond, and so on), Staff (Martha, Pat, Linda and so on), books, labs, offices and so forth.

Semantics for FOPL

- The **functions** we may choose might be advisor-of(x), lab-capacity(y), dept-grade-average(z), and the **predicates** MARRIED(x), TENURED(y), COLLABORATE(x,y) to name a few.
- **When an assignment of values is given to each term and to each predicate symbol in a wff, we say an interpretation is given to the wff.**

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Semantics for FOPL

- If the truth values for **two different wffs** are the **same under every interpretation**, they are said to be **equivalent**.
- A predicate (or wff) that has **no variables** is called a **ground atom**.

Semantics for FOPL

- **For example**, the predicate P(x) in $\forall x$ P(x), is true only if **it is true for every value** of x in the domain D.
- Likewise, the P(x) in $\exists x$ P(x) is **true** only if it is true for **at least one value** of x in the domain.
- If the **above conditions are not satisfied**, the predicate evaluates to **false**.

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Semantics for FOPL

- **For example**, we want to evaluate the truth value of **the expression E**, where
- **E:** $\forall x\ ((A(a,x) \vee B(f(x))) \,\&\, C(x)) \rightarrow D(x)$
- In this expression, there are **four predicates**: **A, B, C,** and **D**.
- The predicate **A** is a **two-place predicate**, the first argument being the **constant** **a**, and the second argument, a **variable** **x**.

Semantics for FOPL

- The predicates **B**, **C** and **D** are all **unary predicates** where the argument of **B is a function f(x),** and the argument of **C and D is the variable x**.
- Since the whole expression E is quantified with the universal quantifier ∀x, it will evaluate to **true** only if it **evaluates are to be true** for all x in the domain D.

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Semantics for FOPL

- Thus, to complete our example, suppose E is interpreted as follows: Define the domain **D = {1, 2}** and from **D** let the interpretation ***I*** assign the following values:
- a = 2
- f(1) = 2, f(2) = 1
- A(2,1) = true, A(2,2) = false
- B(1) = true, B(2) = false
- C(1) = true, C(2) = false
- D(1) = false, D(2) = true

Semantics for FOPL

- Using a table such as Table 4.3 we can evaluates E as follows:

a) . If **x = 1, A(2,1)** evaluates to **true**,

B(1) evaluates to True

, and

(A(2,1) V B(1)) evaluates to true.

C(1) evaluates to true.

Therefore, the expression in the outer parentheses **evaluates to true**.

Hence, since **D(1) evaluates to false**, **the expression E evaluates to false.**

Semantics for FOPL

(b) In a similar way, **if x =2**, the expression can be shown to evaluate to **true**.

Consequently, since E is not true for all x, the expression evaluates to false.

Properties of WFFS (well-formed formulas)

- As in the case of PL, **the evaluation of complex formulas in FOPL can often be facilitated** through the **substitution of equivalent formulas**.
- **Table 4.2 lists a number of equivalent expressions.**
- **Table 4.4 and Table 4.2 are similar**, but there are some notable differences, particularly in the wffs containing quantifiers.

Table 4.2 lists some of the important laws of PL (Some Equivalence Laws)

Name of Laws	Statements
Idempotency	P V P = P P & P = P
Associativity	(PVQ)VR = P V (Q V R) (P & Q) & R = P & (Q & R)
Commutativity	P V Q = Q V P P & Q = Q & P P ↔Q = Q↔P
Distributivity	P & (Q V R) = (P & Q)V (P &R) P V (Q & R) = (P V Q) & (P V R)
De Morgan's Laws	~(P V Q) = ~P & ~Q ~(P & Q) = ~P V ~Q
Conditional Elimination	P → Q = ~P V Q
Bi-conditional Elimination	P ↔Q = (p → Q) & (Q →P)

Table 4.4 Equivalent Logical Expressions

Name of the Rules	Expressions
Double negation	~(~F) = F
Commutativity	F & G = G & F, F V G = G V F
Associativity	(F & G) & H = F & (G & H) (F V G) V H = F V (G V H)
Distributivity	F V (G & H) = (F V G) & (F V H) F & (G V H) = (F & G) V (F & H)
De Morgan's Laws	~(F & G) = ~F V ~G ~(F V G) = ~F & ~G
Conditional Elimination	F → G = ~F V G

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Table 4.4 Equivalent Logical Expressions Cont..

Bi-conditional Elimination	$F \leftrightarrow G = (\sim F \vee G) \,\&\, (\sim G \vee F)$
Quantifiers	$\forall x\, F[x] \vee G = \forall x\, (F[x] \vee G)$ $\exists x\, F[x] \vee G = \exists x\, (F[x] \vee G)$ $\forall x\, F[x] \,\&\, G = \forall x\, (F[x] \,\&\, G)$ $\exists x\, F[x] \,\&\, G = \exists x\, (F[x] \,\&\, G)$ $\sim(\forall x)\, F[x] = \exists x\, (\sim F[x])$ $\sim(\exists x)\, F[x] = \forall x\, (\sim F[x])$ $\forall x\, F[x] \,\&\, \forall x\, G[x] = \forall x\, (F[x] \,\&\, G[x])$ $\exists x\, F[x] \vee \exists x\, G[x] = \exists x\, (F[x] \vee G[x])$

Properties of WFFS

- For example, In Table 4.4 attention is called to the last four expressions which govern substitutions involving negated quantifiers and the movement of quantifiers across conjunctive and disjunctive connections.
- **A wff is said to be valid if it is true under every interpretation**.
- **A wff that is false under every interpretation is said to be inconsistent (or unsatisfiable)**.

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Properties of WFFS

- **If a wff is not valid, then it is called invalid**.
- Likewise, **a wff that is not inconsistent is satisfiable.**
- Again, this means that a valid wff is satisfiable and an inconsistent wff is invalid, but the respective converse statements do not hold.

Properties of WFFS

- **Finally,** we say that a wff Q is a logical consequence of the wffs $P_1, P_2, P_3, \ldots \ldots \ldots, P_n$ if and only if whenever P_1 & P_2 & P_3 & .. .. .. & P_n is **true** under an interpretation.
- Then, **Q is also true**.

Properties of WFFS

- **Example :**

(a). P &~P is inconsistent and P V ~P is valid since the first is false under every interpretation and the second is true under every interpretation.

(b). From the two wffs

CLEVER(bill) and

$\forall x$ CLEVER(x) → SUCCEED(x)

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

Properties of WFFS

- We can show that SUCCEED(bill) is a logical consequence. Thus, assume that both

  CLEVER(bill) and

  $\forall x$ CLEVER(x) → SUCCEED(x)

  are **true under an interpretation**.
- Then

  CLEVER(bill) → SUCCEED(bill)

Is certainly true since the wff was assumed to be true for all x, including x= bill.

Properties of WFFS

- But,

CLEVER(bill) → SUCCEED(bill) = ~ CLEVER(bill) V SUCCEED(bill)

are equivalent and, since CLEVER(bill) **is true**,

~CLEVER(bill) is false and, therefore, **SUCCEED(bill)** **must be true**. Thus, we conclude SUCCEED(bill) is a logical consequence of

CLEVER(bill) and ∀x CLEVER(x) → SUCCEED(x)

CONVERSION TO CLAUSAL FORM

- As noted earlier, we are interested in mechanical inference by programs using symbolic FOPL expressions.
- **One method we shall examine is called resolution.**
- It requires that all statements be converted into a normalized clausal form.
- **We define a clause as the disjunction of a number of literals.**
- **A ground clause is one in which no variables occur in the expression.**
- **A Horn clause is a clause with at most one positive literal.**

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

CONVERSION TO CLAUSAL FORM

- **<u>Clausal Conversion Procedure:</u>**
- **Step 1.:** Eliminate all implication and equivalency connectives (Use ~P V Q in place of P → Q and (~P V Q) & (~Q V P) in place of P↔Q).
- **Step 2:** Move all negations in to immediately precede an atom (use P in place of ~(~P), and DeMorgran's laws, ∃x ~F[x] in place of ~(∀x) F[x] and ∀x ~F[x] in place of ~(∃x) F[x]).

© Prof. Dr. A. K. M. Akhtar Hossain, Dept. of CSE, University of Rajshahi

CONVERSION TO CLAUSAL FORM

- **Step 3:** Rename variables, if necessary, so that all quantifiers have variable assignments; that is, rename variables so that variables bound by a different quantifier.
- **For example**, in the expression
- $\forall x\ (P(x) \to (\exists x\ Q(x)))$

rename the second "dummy" variable **x** which is bound by the existential quantifier to be a different variable, say **y**, to give

$$\forall x\ (P(x) \to (\exists y\ Q(y))).$$

CONVERSION TO CLAUSAL FORM

- **Step 4:** Skolemize by replacing all existentially quantified variables with Skolem functions as described above, and deleting the corresponding existential quantifiers.
- **Step 5:** Move all universal quantifiers to left of the expression and put the expression on the right into CNF (Conjunctive Normal Form).
- **[Skolemization: The process of removing all the existential quantifiers from formula.]**

CONVERSION TO CLAUSAL FORM

- **Step 6:** Eliminate all universal quantifiers and conjunctions since they are retained implicitly. **The resulting expressions are clauses and the set of such expressions is said to be in clausal form.**

- **Example: Page No. 64, Patterson**

- **THANKS**
- **THE END**