Artificial Intelligence

COMPUTER SCIENCE ENGINEERING DEPARTMENT

DCE,GURGAON

Knowledge representation in AI

Symbolic Logic

- Simbolic logic representation
- Formal system
- Propositional logic
- Predicate logic
- Theorem proving

1. Knowledge representation

- Why Symbolic logic
- Power of representation
- Formal language: syntax, semantics
- Conceptualization + representation in a language
- Inference rules

2. Formal systems

- O formal system is a quadruple $S = < A, \mathcal{F}, \mathcal{A}, \Re >$
- A ***rule of inference*** $R \in \Re$ of arity *n* is an association:

$$R \subseteq \mathcal{F}^n \times \mathcal{F},\ \overline{y} = \langle y_1, \ldots, y_n \rangle \xrightarrow{R} x,\ \ x, y_i \in \mathcal{F},\ \ \forall i = \overline{1, n}$$

- Immediate consequence
- Be the set of premises $\Gamma = \{y_1, \ldots, y_n\}$ $E_0 = \Gamma \cup \mathcal{A}$

$$E_1 = E_0 \bigcup_{n \geq 1} \{x | \exists \overline{y} \in E_0^n,\ \overline{y}\, \Re\, x\} \qquad E_2 = E_1 \bigcup_{n \geq 1} \{x | \exists \overline{y} \in E_1^n,\ \overline{y}\, \Re\, x\}$$

- An element $E_i\ (i \geq 0)$

is an immediate **consequence** of a set of premises Γ

Formal systems - cont

- If $E_0 = A$ $(\Gamma = \phi)$ then the elements of E_i are called theorems

- Be $x \in E_i$ a theorem; it can be obtained by successive applications of i.r on the formulas in E_i
- Sequence of rules - **<u>demonstration</u>** . $\vdash_S x \vdash_{\mathcal{R}} x$

- If $E_0 = \Gamma \cup A$ then $x \in E_i$ can be <u>deduced</u> from Γ

$$\Gamma \vdash_S x$$

3. Propositional logic

- Formal language
- **3.1 Syntax**
- Alphabet
- A **well-formed formula** (wff) in propositional logic is:

(1) An atom is a wff

(2) If P is a wff, then ~P is a wff.

(3) If P and Q are wffs then $P \wedge Q$, $P \vee Q$, $P \rightarrow Q$ si $P \leftrightarrow Q$ are wffs.

(4) The set of all wffs can be generated by repeatedly applying rules (1)..(3).

3.2 Semantics

- Interpretation
- *Evaluation function of a formula*
- Properties of wffs
  - Valid / tautulogy
  - Satisfiable
  - Contradiction
  - Equivalent formulas

Semantics - cont

- A formula F is a logical consequence of a formula P
- A formula F is a logical consequence of a set of formulas $P_1,\ldots P_n$
- Notation of logical consequence $P_1,\ldots P_n \Rightarrow F$.
- **Theorem**.Formula F is a logical consequence of a set of formulas $P_1,\ldots P_n$ if the formula $P_1,\ldots P_n \rightarrow F$ is valid.
- **Teorema**. Formula F is a logical consequence of a set of formulas $P_1,\ldots P_n$ if the formula $P_1 \wedge \ldots \wedge P_n \wedge \sim F$ is a contradiction.

Equivalence rules

Idempotenta	$P \vee P \equiv P$	$P \wedge P \equiv P$	
Asociativitate	$(P \vee Q) \vee R \equiv P \vee (Q \vee R)$	$(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$	
Comutativitate	$P \vee Q \equiv Q \vee P$	$P \wedge Q \equiv Q \wedge P$	$P \leftrightarrow Q \equiv Q \leftrightarrow P$
Distributivitate	$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$	$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$	
De Morgan	$\sim (P \vee Q) \equiv \sim P \wedge \sim Q$	$\sim (P \wedge Q) \equiv \sim P \vee \sim Q$	
Eliminarea implicatiei	$P \rightarrow Q \equiv \sim P \vee Q$		
Eliminarea implicatiei duble	$P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$		

3.3 Obtaining new knowledge

- Conceptualization
- Reprezentation in a formal language
- Model theory

  $KB \models_x M$

- Proof theory

  $KB \vdash_{S\ x} M$

- Monotonic logics
- Non-monotonic logics

3.4 Inference rules

- *Modus Ponens*

$$\frac{\begin{array}{c}P \\ P \rightarrow Q\end{array}}{Q}$$

- *Substitution*
- *Chain rule*

$$\frac{\begin{array}{c}P \rightarrow Q \\ Q \rightarrow R\end{array}}{P \rightarrow R}$$

- *AND introduction*

$$\frac{\begin{array}{c}P \\ Q\end{array}}{P \wedge Q}$$

- *Transposition*

$$\frac{P \rightarrow Q}{\sim Q \rightarrow \sim P}$$

Example

- *Mihai has money*
- *The car is white*
- *The car is nice*
- *If the car is white or the car is nice and Mihai has money then Mihai goes to the mountain*
- B
- A
- F
- $(A \vee F) \wedge B \rightarrow C$

4. First order predicate logic

4.1 Syntax

Be *D* a domain of values. A *term* is defined as:

- (1) A constant is a term with a fixed value belonging to *D*.
- (2) A variable is a term which may take values in *D*.
- (3) If *f* is a function of n arguments and $t_1,..t_n$ are terms then $f(t_1,..t_n)$ is a term.
- (4) All terms are generated by the application of rules (1)…(3).

Syntax PL - cont

- Predicates of arity n
- Atom or atomic formula.
- Literal

A *well formed formula (wff)* in first order predicate logic is defined as:

(1) A atom is an wff

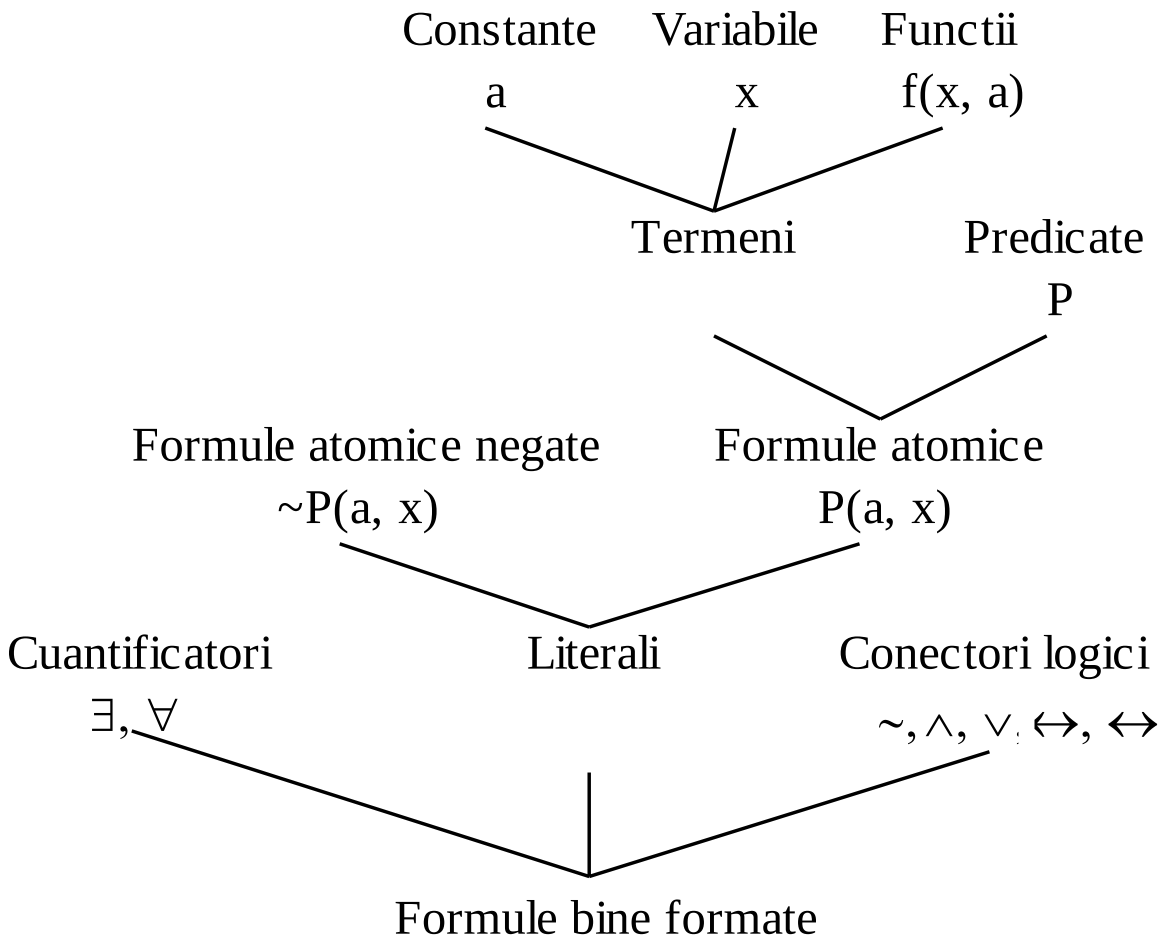
(2) If P[x] is a wff then ~P[x] is an wff.

(3) If P[x] and Q [x] are wffs then $P[x] \wedge Q[x]$, $P[x] \vee Q[x]$, $P \rightarrow Q$ and $P \leftrightarrow Q$ are wffs.

(4) If P[x] is an wff then $\forall x\ P[x]$, $\exists x\ P[x]$ are wffs.

(5) The set of all wffs can be generated by repeatedly applying rules (1)..(4).

Syntax - schematically



CNF, DNF

- Conjunctive normal form (CNF)

$$F_1 \wedge \ldots \wedge F_n,$$

$$F_i \text{ , } i=1,n$$

$$(L_{i1} \vee \ldots \vee L_{im}).$$

- Disjunctive normal form (DNF)

$$F_1 \vee \ldots \vee F_n,$$

$$F_i \text{ , } i=1,n$$

$$(L_{i1} \wedge \ldots \wedge L_{im})$$

4.2 Semantics of PL

- *The interpretation of a formula* F in first order predicate logic consists of fixing a domain of values (non empty) D and of an association of values for every constant, function and predicate in the formula F as follows:
- (1) Every constant has an associated value in D.
- (2) Every function f, of arity n, is defined by the correspondence $D^n \to D$ where

$$D^n = \{(x_1, \ldots, x_n) | x_1 \in D, \ldots, x_n \in D\}$$

- (3) Every predicate of arity n, is defined by the correspondence $P : D^n \to \{t, \mathbf{f}\}$

Interpretation - example

$$(\forall x)(((A(a,x) \vee B(f(x))) \wedge C(x)) \rightarrow D(x))$$

D={1,2}

a
2

f(1)	f(2)
2	1

A(2,1)	A(2,2)	B(1)	B(2)	C(1)	C(2)	D(1)	D(2)
a	**f**	**a**	**f**	**a**	**f**	**f**	**a**

X=1 $((\mathbf{a} \vee \mathbf{f}) \wedge \mathbf{a}) \rightarrow \mathbf{f}$

X=2 $((\mathbf{f} \vee \mathbf{a}) \wedge \mathbf{f}) \rightarrow \mathbf{a}$

4.3 Properties of wffs in PL

- Valid / tautulogy
- Satisfiable
- Contradiction
- Equivalent formulas

- A formula F is a logical consequence of a formula P
- A formula F is a logical consequence of a set of formulas $P_1,\ldots P_n$
- Notation of logical consequence $P_1,\ldots P_n \Rightarrow F$.
- **Theorem**. Formula F is a logical consequence of a set of formulas $P_1,\ldots P_n$ if the formula $P_1 \wedge\ldots \wedge P_n \rightarrow F$ is valid.
- **Teorema**. Formula F is a logical consequence of a set of formulas $P_1,\ldots P_n$ if the formula $P_1\wedge\ldots \wedge P_n \wedge \sim F$ is a contradiction.

Equivalence of quantifiers

$(Qx)F[x] \vee G \equiv (Qx)(F[x] \vee G)$	$(Qx)F[x] \wedge G \equiv (Qx)(F[x] \wedge G)$
$\sim ((\forall x)F[x]) \equiv (\exists x)(\sim F[x])$	$\sim ((\exists x)F[x]) \equiv (\forall x)(\sim F[x])$
$(\forall x)F[x] \wedge (\forall x)H[x] \equiv (\forall x)(F[x] \wedge H[x])$	$(\exists x)F[x] \vee (\exists x)H[x] \equiv (\exists x)(F[x] \vee H[x])$
$(Q_1x)F[x] \wedge (Q_2x)H[x] \equiv (Q_1x)(Q_2z)(F[x] \wedge H[z])$	$(Q_1x)F[x] \vee (Q_2x)H[x] \equiv (Q_1x)(Q_2z)(F[x] \vee H[z])$

Examples

- All apples are red
- All objects are red apples
- There is a red apple
- All packages in room 27 are smaller than any package in room 28
  - All purple mushrooms are poisonous
  - $\forall x\ (Purple(x) \wedge Mushroom(x)) \Rightarrow Poisonous(x)$
  - $\forall x\ Purple(x) \Rightarrow (Mushroom(x) \Rightarrow Poisonous(x))$
  - $\forall x\ Mushroom\ (x) \Rightarrow (Purple\ (x) \Rightarrow Poisonous(x))$

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

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

4.4. Inference rules in PL

- Modus Ponens

- Substitution
- Chaining
- Transpozition
- AND elimination (AE)
- AND introduction (AI)
- Universal instantiation (UI)
- Existential instantiation (EI)
- Rezolution

Example

- Horses are faster than dogs and there is a greyhound that is faster than every rabbit. We know that Harry is a horse and that Ralph is a rabbit. Derive that Harry is faster than Ralph.

- Horse(x) Greyhound(y)
- Dog(y) Rabbit(z)
- Faster(y,z))

$\forall x\ \forall y\ \text{Horse}(x) \land \text{Dog}(y) \Rightarrow \text{Faster}(x,y)$

$\exists y\ \text{Greyhound}(y) \land (\forall z\ \text{Rabbit}(z) \Rightarrow \text{Faster}(y,z))$

Horse(Harry)

Rabbit(Ralph)

$\forall y\ \text{Greyhound}(y) \Rightarrow \text{Dog}(y)$

$\forall x\ \forall y\ \forall z\ \text{Faster}(x,y) \land \text{Faster}(y,z) \Rightarrow \text{Faster}(x,z)$

Proof example

- **Theorem**: Faster(Harry, Ralph) ?
- **Proof using inference rules**

1. $\forall x\ \forall y\ Horse(x) \wedge Dog(y) \Rightarrow Faster(x,y)$
2. $\exists y\ Greyhound(y) \wedge (\forall z\ Rabbit(z) \Rightarrow Faster(y,z))$
3. $\forall y\ Greyhound(y) \Rightarrow Dog(y)$
4. $\forall x \forall y \forall z\ Faster(x,y) \wedge Faster(y,z) \Rightarrow Faster(x,z)$
5. $Horse(Harry)$
6. $Rabbit(Ralph)$
7. $Greyhound(Greg) \wedge (\forall z\ Rabbit(z) \Rightarrow Faster(Greg,z))$ 2, EI
8. $Greyhound(Greg)$ 7, AE
9. $\forall z\ Rabbit(z) \Rightarrow Faster(Greg,z))$ 7, AE

Proof example - cont

10.	Rabbit(Ralph) ⇒ Faster(Greg,Ralph)	9, UI
11.	Faster(Greg,Ralph)	6,10, MP
12.	Greyhound(Greg) ⇒ Dog(Greg)	3, UI
13.	Dog(Greg)	12, 8, MP
14.	Horse(Harry) ∧ Dog(Greg) ⇒ Faster(Harry, Greg)	1, UI
15.	Horse(Harry) ∧ Dog(Greg)	5, 13, AI
16.	Faster(Harry, Greg)	14, 15, MP
17.	Faster(Harry, Greg) ∧ Faster(Greg, Ralph) ⇒ Faster(Harry,Ralph)	4, UI
18.	Faster(Harry, Greg) ∧ Faster(Greg, Ralph)	16, 11, AI
19.	Faster(Harry,Ralph)	17, 19, MP