CS5201: Advanced Artificial Intelligence

Predicate Logic

Arijit Mondal
Dept of Computer Science and Engineering
Indian Institute of Technology Patna
www.iitp.ac.in/~arijit/

Insufficiency of Propositional Logic

- Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.
- No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.
- All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.
- Every passenger is either in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.

Predicate logic

- Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.
- No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.
- All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.
- Every passenger is either in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.
- New addition in Proposition logic (First order logic)

Predicate logic

- Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.
- No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.
- All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.
- Every passenger is either in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.
- New addition in Proposition logic (First order logic)
  - Variables, Constants, Predicate symbols

Predicate logic

- Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.
- No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.
- All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.
- Every passenger is either in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.
- New addition in Proposition logic (First order logic)
  - Variables, Constants, Predicate symbols
  - New connectors: $\exists$ (there exists), $\forall$ (for all)

Formulating Predicate Logic Statement - 1

- **Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.**
  - **Predicate $goes(x, y)$ to denote x goes to y**

Formulating Predicate Logic Statement - 1

- **Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.**
  - **Predicate** $goes(x, y)$ **to denote** x **goes to** y
  - F_1 **:** $\forall x(goes(Mary, x) \rightarrow goes(Lamb, x))$

Formulating Predicate Logic Statement - 1

- **Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.**
  - **Predicate** $goes(x, y)$ **to denote** x **goes to** y
  - F_1 **:** $\forall x(goes(Mary, x) \rightarrow goes(Lamb, x))$
  - F_2 **:** $goes(Mary, School)$

Formulating Predicate Logic Statement - 1

- **Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.**
  - **Predicate** $goes(x, y)$ **to denote** x **goes to** y
  - F_1 **:** $\forall x(goes(Mary, x) \rightarrow goes(Lamb, x))$
  - F_2 **:** $goes(Mary, School)$
  - G **:** $goes(Lamb, School)$

Formulating Predicate Logic Statement - 1

- **Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.**
  - **Predicate** $goes(x, y)$ **to denote** x **goes to** y
  - F_1 **:** $\forall x(goes(Mary, x) \rightarrow goes(Lamb, x))$
  - F_2 **:** $goes(Mary, School)$
  - G **:** $goes(Lamb, School)$
- **To prove:** $(F_1 \wedge F_2) \rightarrow G$ **is always true**

Formulating Predicate Logic Statement - 2

- No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.

Formulating Predicate Logic Statement - 2

- **No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.**
  - **Predicate** $contractor(x), dependable(x), engineer(x)$

Formulating Predicate Logic Statement - 2

- **No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.**
  - **Predicate** $contractor(x), dependable(x), engineer(x)$
  - F_1 **:** $\forall x(contractor(x) \rightarrow \neg dependable(x))$
    Alternatively, $\neg\exists x(contractor(x) \wedge dependable(x))$

Formulating Predicate Logic Statement - 2

- **No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.**
  - **Predicate** $contractor(x), dependable(x), engineer(x)$
  - F_1 **:** $\forall x(contractor(x) \rightarrow \neg dependable(x))$
    Alternatively, $\neg\exists x(contractor(x) \wedge dependable(x))$
  - F_2 **:** $\exists x(engineer(x) \wedge contractor(x))$

Formulating Predicate Logic Statement - 2

- **No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.**
  - **Predicate** $contractor(x), dependable(x), engineer(x)$
  - F_1 **:** $\forall x(contractor(x) \rightarrow \neg dependable(x))$
    Alternatively, $\neg\exists x(contractor(x) \wedge dependable(x))$
  - F_2 **:** $\exists x(engineer(x) \wedge contractor(x))$
  - G **:** $\exists x(engineer(x) \wedge \neg dependable(x))$

Formulating Predicate Logic Statement - 2

- **No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.**
  - **Predicate** $contractor(x), dependable(x), engineer(x)$
  - F_1 **:** $\forall x(contractor(x) \rightarrow \neg dependable(x))$
    Alternatively, $\neg\exists x(contractor(x) \wedge dependable(x))$
  - F_2 **:** $\exists x(engineer(x) \wedge contractor(x))$
  - G **:** $\exists x(engineer(x) \wedge \neg dependable(x))$
- **To prove:** $(F_1 \wedge F_2) \rightarrow G$ **is always true**

Example - 3

- All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.

Example - 3

- **All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.**
  - **Predicate** $dancer(x), graceful(x), students(x)$
  - F_1: $\forall x(dancer(x) \rightarrow graceful(x))$
  - F_2: $student(Ayesha)$
  - F_3: $dancer(Ayesha)$
  - G: $\exists x\{student(x) \wedge graceful(x)\}$
- **To prove:** $(F_1 \wedge F_2 \wedge F_3) \rightarrow G$ **is always true**

Example - 4

- Every passenger is either in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.

Example - 4

- **Every passenger is either in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.**
  - **Predicate** $p(x), f(x), s(x), w(x)$
  - F_1**:** $\forall x\{p(x) \rightarrow (f(x) \oplus s(x))\}$
  - F_2**:** $\forall x\{p(x) \rightarrow (s(x) \leftrightarrow \neg w(x))\}$
  - F_3**:** $\exists x\{p(x) \wedge w(x)\}$
  - F_4**:** $\neg(\forall x(p(x) \rightarrow w(x)))$ **also** $\exists x(p(x) \wedge \neg w(x))$
  - G**:** $\exists x(p(x) \wedge s(x))$

Use of quantifiers

- Someone likes everyone
- Everyone likes someone
- There is someone whom everyone likes
- Everyone likes everyone
- If everyone likes everyone then someone likes everyone
- If there is a person whom everyone likes then that person likes himself
- Laws of negation

Use of function symbols

- If x is greater than y and y is greater than z then x is greater than z
  - g(x,y) – x is greater than y
  - $\forall x \forall y \forall z((g(x, y) \wedge g(y, z)) \rightarrow g(x, z))$
- The age of a person is greater than the age of his child
  - Age(x) – Function symbol; child(x,y) – x is child of y
  - $\forall x \forall y(child(x, y) \rightarrow g(Age(y), Age(x))$
  - Age(x) returns value
  - child(x,y) returns TRUE or FALSE
- Therefore the age of a person is greater than the age of his grandchild
- The sum of ages of two children are never more than the sum of ages of their parents

Variables and Predicate / Function Symbols

- **Variables, Free variables, Bound variables**
  - $\forall x(p(x, y))$
  - $\forall x\{p(x, y) \wedge \exists z\ q(x, y, z, w)\}$
  - $\forall x\{p(x, y) \wedge \exists z \exists y\ q(x, y, z, w)\}$
- **Symbols – proposition symbols, constant symbols, function symbols, predicate symbols**
- **Variables can be quantified in first order predicate logic**
- **Symbols cannot be quantified in first order predicate logic**
- **Interpretations are mapping of symbols to relevant aspects of a domain**

Terminology for Predicate Logic

- **Domain: D**
- **Constant symbols: M, N, O, P, ...**
- **Variable symbols: x, y, z, ...**
- **Function symbols: F(x), G(x,y), ...**
- **Predicate symbols: p(x), q(x,y), ...**
- **Connectors:** $\sim, \wedge, \vee, \rightarrow, \exists, \forall$
- **Terms:**
- **Well-formed formula:**
- **Free and bound variables**
- **Interpretation, valid, non-valid, satisfiable, unsatisfiable**

Validity, Satisfiability, Structure

- F_1 : $\forall x(goes(Mary, x) \rightarrow goes(Lamb, x))$
- F_2 : $goes(Mary, School)$
- G : $goes(Lamb, School)$
- **To prove:** $(F_1 \wedge F_2) \rightarrow G$ **is always true**
- **The above is the same as follows**
- F_1 : $\forall x(www(M, x) \rightarrow www(L, x))$
- F_2 : $www(M, S)$
- G : $www(L, S)$
- **To prove:** $(F_1 \wedge F_2) \rightarrow G$ **is always true**

Interpretations

- **What is an interpretation? Assign a domain set D, map constants, functions, predicates suitably**
- **The formula will now have a truth value**
- **Example:** $F_1 : \forall x(g(M, x) \rightarrow g(L, x))$, $F_2 : g(M, S)$, $G : g(L, S)$
- **Interpretation 1: D={Akash, Baby, Home Play, Ratan, Swim}, etc.**
- **Interpretation 2: D = set of integers**
- **How many interpretations can there be?**
- **To prove Validity, means** $((F_1 \wedge F_2) \rightarrow G)$ **is true under all interpretations**
- **To prove Satisfiability, means** $((F_1 \wedge F_2) \rightarrow G)$ **is true under at least one interpretation**

Russell's Paradox

- There is a single barber in town.
- Those and only those who do not shave themselves are shaved by the barber
- Who shaves the barber?

Resolution Refutation for Propositional Logic

- **To prove validity of $M = ((F_1 \wedge F_2 \wedge \ldots \wedge F_n) \rightarrow G)$ we shall attempt to prove that $\neg M = (F_1 \wedge F_2 \wedge \ldots \wedge F_n \wedge \neg G)$ is unsatisfiable**
- **Steps for proof by resolution refutation**
  - **Convert to clausal form / Conjunctive Normal Form (CNF, Product of sums)**
  - **Generate new clauses using resolution rule**
  - **At the end, either false will be derived if the formula $\neg M$ is unsatisfiable implying M is valid**

Resolution Refutation for Propositional Logic

- If Asha is elected VP then Rajat is chosen as G-Sec and Bharati is chosen as Treasurer. Rajat is not chosen as G-Sec. Therefore Asha is not elected VP.

Resolution Refutation for Propositional Logic

- **If Asha is elected VP then Rajat is chosen as G-Sec and Bharati is chosen as Treasurer. Rajat is not chosen as G-Sec. Therefore Asha is not elected VP.**
- **F_1:** $(a \rightarrow (b \wedge c)) = (\neg a \vee b) \wedge (\neg a \vee c)$
- **F_2:** $\neg b$, **G:** $\neg a$, $\neg$**G:** a
- **Let $C_1 = a \vee b$, $C_2 = \neg a \vee c$, then $C_3 = b \vee c$ can be derived**
- **To prove unsatisfiability use the resolution rule repeatedly to reach a situation where we have two contradictory clauses of the form $C_1 = a$ and $C_2 = \neg a$ from which false can be derived**
- **If the proposition formula is satisfiable then we will not reach a contradiction and eventually no new clauses will be derivable**
- **For propositional logic the procedure terminates**
- **Resolution rule is sound and complete**

Example

- Rajesh either took the bus or came by cycle. If he came by cycle or walked to class he arrived late. Rajesh did not arrive late. Therefore he took the bus to class.

Resolution Refutation for Predicate Logic

- Given a formula M which we wish to check for validity, we first check if there are any free variables. We then quantify all free variables universally
- Create $M' = \neg M$ and check for unsatisfiability of M'
- Steps:
  - Conversion to clausal form
    - Handling of variables and quantifiers, ground instances
  - Applying the resolution rule
    - Concept of unification
    - Principle of most general unifier (mgu)
    - Repeated application of resolution rule using mgu

Resolution Refutation for Predicate Logic

- **Conversion to clausal form in predicate logic**
  - **Remove implications and other Boolean symbols converting to equivalent forms using $\neg, \vee, \wedge$**
  - **Move negates ($\neg$) inwards as close as possible**
  - **Standardize (rename) variables to make them unambiguous**
  - **Remove existential quantifiers by an appropriate new function / constant symbol taking into account the variables dependent on the quantifier (Skolemization)**
  - **Drop universal quantifiers**
  - **Distribute $\vee$ over $\wedge$ and convert to CNF**

Conversion to clausal form

- Remove implications and other Boolean symbols converting to equivalent forms using $\neg, \vee, \wedge$
- Move negates ($\neg$) inwards as close as possible
- Standardize (rename) variables to make them unambiguous
- Remove existential quantifiers by an appropriate new function / constant symbol taking into account the variables dependent on the quantifier (Skolemization)
- Drop universal quantifiers
- Distribute $\vee$ over $\wedge$ and convert to CNF

- $\forall x\{(\forall y(student(y) \rightarrow likes(x, y))) \rightarrow (\exists z(likes(z, x)))\}$

Substitution, Unification, Resolution

- **Consider the following clauses:**
  - C_1: $\neg studies(x, y) \vee passes(x, y)$
  - C_2: $studies(Madan, z)$
  - C_3: $\neg passes(Chetan, Physics)$
  - C_4: $\neg passes(w, Mechanics)$
- **What new clauses can we derive by the resolution principle?**

Example

- F_1: $\forall x(contractor(x) \rightarrow \neg dependable(x))$
- F_2: $\exists x(engineer(x) \wedge contractor(x))$
- G: $\exists x(engineer(x) \wedge \neg dependable(x))$

Example

- F_1: $\forall x(dancer(x) \rightarrow graceful(x))$
- F_2: $student(Ayesha)$
- F_3: $dancer(Ayesha)$
- G: $\exists x(student(x) \wedge graceful(x))$

Example

- All hounds howl at night.
- Anyone who has any cats will not have any mice.
- Light sleepers do not have anything which howls at night.
- John has either a cat or a hound.
- Therefore, if John is a light sleeper, then John does not have any mice.

Example

- **All hounds howl at night.**
  $\forall x\ (hound(x) \rightarrow howl(x))$
- **Anyone who has any cats will not have any mice.**
  $\forall x \forall y\ (owns(x, y) \wedge cat(y) \rightarrow \neg \exists z\ (owns(x, z) \wedge mouse(z)))$
- **Light sleepers do not have anything which howls at night.**
  $\forall x\ (ls(x) \rightarrow \neg \exists y\ (howl(y) \wedge owns(x, y)))$
- **John has either a cat or a hound.**
  $\exists x\ (owns(J, x) \wedge (cat(x) \vee hound(x)))$
- **Therefore, if John is a light sleeper, then John does not have any mice.**
  $ls(J) \rightarrow \neg \exists x\ (owns(J, x) \wedge mouse(x))$

Thank you!