# First Order Predicate and Inference Rules

Module 4

# Well-formed formulas

- wffs in FOPL are constructed using the following rules:

1. True and False are wffs.
2. Each propositional constant (i.e. specific proposition) is a wff
3. Each atomic formula (i.e. a specific predicate with variables) is a wff.
4. If A and B are wffs, then so are $\neg A$, $(A \vee B)$, $(A \wedge B)$, $(A \Rightarrow B)$, and $A \Leftrightarrow B$.
5. If x is a variable (representing objects of the universe of discourse), and A is a wff, then so are $\forall x\ A$ and $\exists x\ A$.

# Clausal forms (also called conjuctive normal forms (CNFs))

- A literal is either an atomic formula or the negation of an atomic formula.
- A clause is the disjunction of literals.
- A formula in clausal form (or, CNF) consists of a conjunction of clauses.

# Algorithm to convert a wff to the clausal form

1. **Eliminate the two-way implication** (biconditional) $\Leftrightarrow$ : Replace $a \Leftrightarrow b$ by $(a \Rightarrow b) \wedge (b \Rightarrow a)$.
2. **Eliminate implications** : Replace $a \Rightarrow b$ by $\neg a \vee b$.
3. **Move $\neg$ inwards**

   $\neg(a \vee b)$ by $\neg a \wedge \neg b$

   $\neg(a \wedge b)$ by $\neg a \vee \neg b$

   $\neg \forall x\, P$ by $\exists x \neg P$

   $\neg \exists x\, P$ by $\forall x \neg P$

**4. Standardize variables** Change variables so that each quantifier binds a unique variable. For example, for sentences like $(\exists\, x\, P(x)) \vee (\exists\, x\, Q(x))$ which use the same variable name twice, change the name of one of the variables.

$$(\exists\, x\, P(x)) \vee (\exists\, y\, Q(y))$$

**5. Skolemize**

- Skolemization is the process of removing existential quantifiers by elimination.
- a) Existentially quantified variables which are not inside the scope of a universal quantifier are replaced by creating new constants. For example, $\exists x\, P(x)$ is changed to $P(c)$, where c is a new constant.
- (b) Replace every existentially quantified variable y with a term $f(x1, \ldots, xn)$ whose function symbol f is new. The variables of this term are the variables that are universally quantified and whose quantifiers precede that of y.
- For example, in the formula $\forall x\, \exists y\, \forall z\, P(x, y, z)$ is replaced by $\forall x\, \forall z\, P(x, f(x), z)$.
- $\forall x\, \forall y\, \exists z\, \forall u\, \exists v\, P(x, y, z, u, v)$ will be replaced by $\forall x\, \forall y\, \forall u\, P(x, y, F(x, y), u, G(x, y, u))$ where $F(x, y)$ and $G(x, y, u)$ are Skolem functions.

**6. Remove universal quantifiers**

At this point, all remaining variables will be universally quantified. We now drop the universal quantifiers.

**7. Distribute ∨ over ∧**

Convert the resulting expression into a conjunction of disjunctions by using distributivity.

Replace $a \vee (b \wedge b)$ by $(a \vee b) \wedge (a \vee b)$.

**8. CNF of the given wff** The sentence is now in CNF

- Convert the following wff to the clausal form:

$$\forall x \left[ [E(x)] \Leftrightarrow [\forall y \; E(T(x, y))] \right]$$

Step 1. Remove biconditional

$$\forall x\ ([E(x)] \Rightarrow [\forall y\ E(T(x,y))]) \land ([\forall y\ E(T(x,y))] \Rightarrow [E(x)])$$

Step 2. Remove implications

$$\forall x\ (\neg[E(x)] \lor [\forall y\ E(T(x,y))]) \land (\neg[\forall y\ E(T(x,y))] \lor [E(x)])$$

Step 3. Move $\neg$ inwards

$$\forall x\ (\neg[E(x)] \lor [\forall y\ E(T(x,y))]) \land ([\exists y\ \neg E(T(x,y))] \lor [E(x)])$$

Step 4. Standardize variables

$$\forall x\ (\neg[E(x)] \lor [\forall y\ E(T(x,y))]) \land ([\exists z\ \neg E(T(x,z))] \lor [E(x)])$$

Step 5. Skolemize

$$\forall x\ (\neg[E(x)] \lor [\forall y\ E(T(x,y))]) \land ([\neg E(T(x,F(x)))] \lor [E(x)])$$

where $F(x)$ is a Skolem function.

Step 6. Drop universal quantifiers

$$(\neg[E(x)] \lor [E(T(x,y))]) \land ([\neg E(T(x,F(x)))] \lor [E(x)])$$

Step 7. This is the CNF of the given sentence.

$$\forall x\ [Romans(x) \wedge know(x, \text{Marcus})] \Rightarrow$$
$$[hate(x, \text{Caesar}) \vee (\forall y\ \exists z\ hate(y, z) \Rightarrow thinkcrazy(x, y))]$$

Step 1. Remove implications

$$\forall x \ \neg[R(x) \wedge K(x, M)] \vee [H(x, C) \vee (\forall y \ \neg(\exists z \ H(y, z)) \vee T(x, y))]$$

Step 2. Move $\neg$ inwards

$$\forall x \ [\neg R(x) \vee \neg K(x, M)] \vee [H(x, C) \vee (\forall y \ (\forall z \ \neg H(y, z)) \vee T(x, y))]$$

Step 3. Standardise variables

The variables are already standardised.

Step 4. Skolemize

No change.

Step 5. Remove universal quantifiers

$$[\neg R(x) \vee \neg K(x, M)] \vee [H(x, C) \vee ((\neg H(y, z)) \vee T(x, y))]$$

Step 6. This is the CNF of the given sentence.

# Argument

- An argument in propositional logic is a sequence of propositions.
- All propositions are called premises and the final proposition is called the conclusion.
- An argument is valid if the truth of all its premises implies that the conclusion is true.

$$\begin{array}{lll} & & P \vee Q \\ \text{Premises} & & P \Rightarrow R \\ & & Q \Rightarrow R \\ \hline \text{Conclusion} & \therefore & R \end{array}$$

Figure 7.1: An argument in propositional logic

It means that if the statements PvQ, P->R and Q->R are true then R is also true

# Proof of validity of an argument

- A proof of validity of a given argument is a sequence of statements, each of which is either a premise of that argument or follows from preceding statements of the sequence by an elementary valid argument, such that the last statement in the sequence is the conclusion of the argument whose validity is being proved

- Proof of validity of above argument

| Step no. | Formula | Derivation |
| --- | --- | --- |
| 1 | $P \vee Q$ | Premise |
| 2 | $P \Rightarrow R$ | Premise |
| 3 | $Q \Rightarrow R$ | Premise |
| 4 | $\neg P \vee R$ | Step 2, logical identity |
| 5 | $Q \vee R$ | Steps 1,4 (Resolution) |
| 6 | $\neg Q \vee R$ | Step 3, logical identity |
| 7 | $R \vee R$ | Steps 5, 6 (Resolution) |
| 8 | $R$ | Step 7, logical identity |

# Rules of inference

- The elementary valid arguments that are used in constructing proofs of validity of arguments are called the rules of inference.
- Rules of inference are templates for building valid arguments.

# Rules of inference

- Constructing truth tables, one can prove that the following is a tautology: (P^(P->Q)) ->Q
  This means that if P and P -> Q are true then Q is also true

$$\begin{array}{rl} & P \\ & P \Rightarrow Q \\ \hline \therefore & Q \end{array}$$

- This is a well known rule of inference is known as “modus-ponens”

# Inference rules in FOPL

1 **Rules borrowed from propositional logic**

- All the inference rules for propositional logic like modus ponens, modus tollens, resolution, etc. can be applied in FOPL as well

| Sl. No. | Inference rule | Name |
| --- | --- | --- |
| 1 | $$\begin{array}{rl} & P \\ & P \Rightarrow Q \\ \hline \therefore & Q \end{array}$$ | Modus ponens |
| 2 | $$\begin{array}{rl} & \neg Q \\ & P \Rightarrow Q \\ \hline \therefore & \neg P \end{array}$$ | Modus tollens |
| 3 | $$\begin{array}{rl} & P \Rightarrow Q \\ & Q \Rightarrow R \\ \hline \therefore & P \Rightarrow R \end{array}$$ | Hypothetical syllogism |
| 4 | $$\begin{array}{rl} & P \vee Q \\ & \neg P \\ \hline \therefore & Q \end{array}$$ | Disjunctive syllogism |

| | | |
|---|---|---|
| 5 | $\begin{array}{rl} & P \\ \hline \therefore & P \lor Q \end{array}$ | Addition |
| 6 | $\begin{array}{rl} & P \land Q \\ \hline \therefore & P \end{array}$ | Simplification |
| 7 | $\begin{array}{rl} & P \\ & Q \\ \hline \therefore & P \land Q \end{array}$ | Conjunction |
| 8 | $\begin{array}{rl} & P \lor Q \\ & \neg P \lor R \\ \hline \therefore & Q \lor R \end{array}$ | Resolution |

# Inference Rules

**2. Universal Instantiation (UI)**

UI says that if $\forall$ x P(x) is true then P(c) is true, where c is a constant in the domain

# Universal Instantiation

Consider the following predicate formula:

$\forall$ x King(x) ^ Greedy(x) ->Evil(x)

If this is true then we can infer that the following are all true.

- King(John)^Greedy(John) ->Evil(John)
- King(Richard)^Greedy(Richard) ->Evil(Richard)
- King(Father(John)) ^Greedy(Father(John)) -> Evil(Father(John))

# Inference Rules

**3. Existential Instantiation (EI)**

- EI says that in a predicate formula involving existential quantifiers, the variable may be replaced by a single new constant symbol.
- In short, EI says that if $\exists\, x\, P(x)$ is true then $P(k)$ is true, where k is a new constant

# Existential Instantiation

For example, from the sentence

$$\exists\, x\ Crown(x) \wedge OnHead(x, John)$$

we can infer the sentence

$$Crown(C) \wedge OnHead(C, John)$$

as long as C does not appear elsewhere in the knowledge base.

# Unification

- To state the resolution inference rule in FOPL. we require the concept of unification.
- Unification is the process of finding substitutions that make different logical expressions look identical.
- For example, consider the logical expressions P(x,y) and P(a,f(z)).
- If we substitute x by a and y by f(z) in the first expression, the first expression will be identical to the second expression. We say that the unifier of the expressions is the set = {a/x,f(z) /y}

- Knows(John, x), Knows(x, Elizabeth)
- two sentences are unified by the substitutions $\theta$ = {x/Elizabeth, y/John}.
- For unification of two sentences, the predicate symbol must be same and number of arguments must also be the same

# Resolution-refutation method in FOP

Let be required to produce a proof of a statement $P$ from a set of statements $F$.

Step 1. Convert all statements in $F$ to clause form.

Step 2. Negate $P$ and convert the result to clause form. Add it to the set of clauses obtained in Step 1.

Step 3. Repeat until either a contradiction is found, no progress can be made, or a predetermined effort has been expended.

(a) Select two clauses. Call these the parent clauses.

(b) Resolve them together. The resulting clause is called the resolvent.

(The resolvent is the disjunction of all literals of both the parent clauses with the following exception: If there are any pairs of literals $T_1$ and $\not{T}_2$ such that one of the parent clauses contains $T_1$ and the other contains $T_2$ and if $T_1$ and $T_2$ are unifiable, then neither $T_1$ $T_2$ should appear in the resolvent. We call $T_1$ and $T_2$ complementary literals. Use the substitution produced by unification to create the resolvent. If there is more than one pair of complementary literals, only one pair should be omitted from the resolvent.)

(c) If the resolvent is the empty clause then a contradiction has been found.

(d) If it is not, then add it to the set of clauses available to the procedure.

- If a contradiction has been found, we conclude that P is a valid conclusion from the premises F. Otherwise, that is if the algorithm terminates in such a status that no progress can be made, we conclude that P is not a valid conclusion from F.

# Resolution Refutation method

1. Convert the facts into statements in FOPL.
2. Convert FOPL statements into CNF.
3. Negate the statement which needs to be proved and to the premises.
4. Draw the resolution graph showing the various unifications.

# Resolution Refutation method - Example

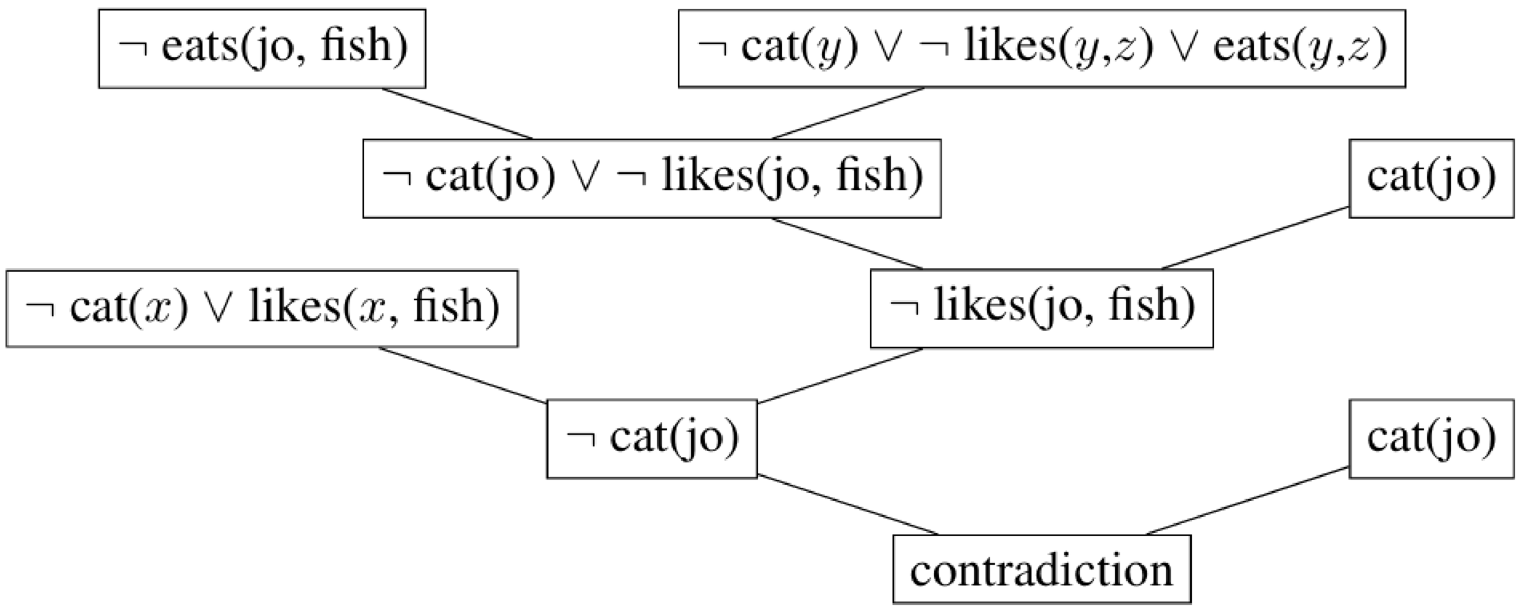
- Prove the validity of the following argument:
  1. Cats like fish.
  2. Cats eat everything they like.
  3. Josephine is a cat.
  4. Therefore, Josephine eats fish.

1. $\neg\, cat(x) \vee likes(fish)$

2. $\neg cat(y) \vee \neg likes(y, z) \vee eats(y, z)$

3. $cat(jo)$

4. $eats(jo, fish)$ (conclusion)



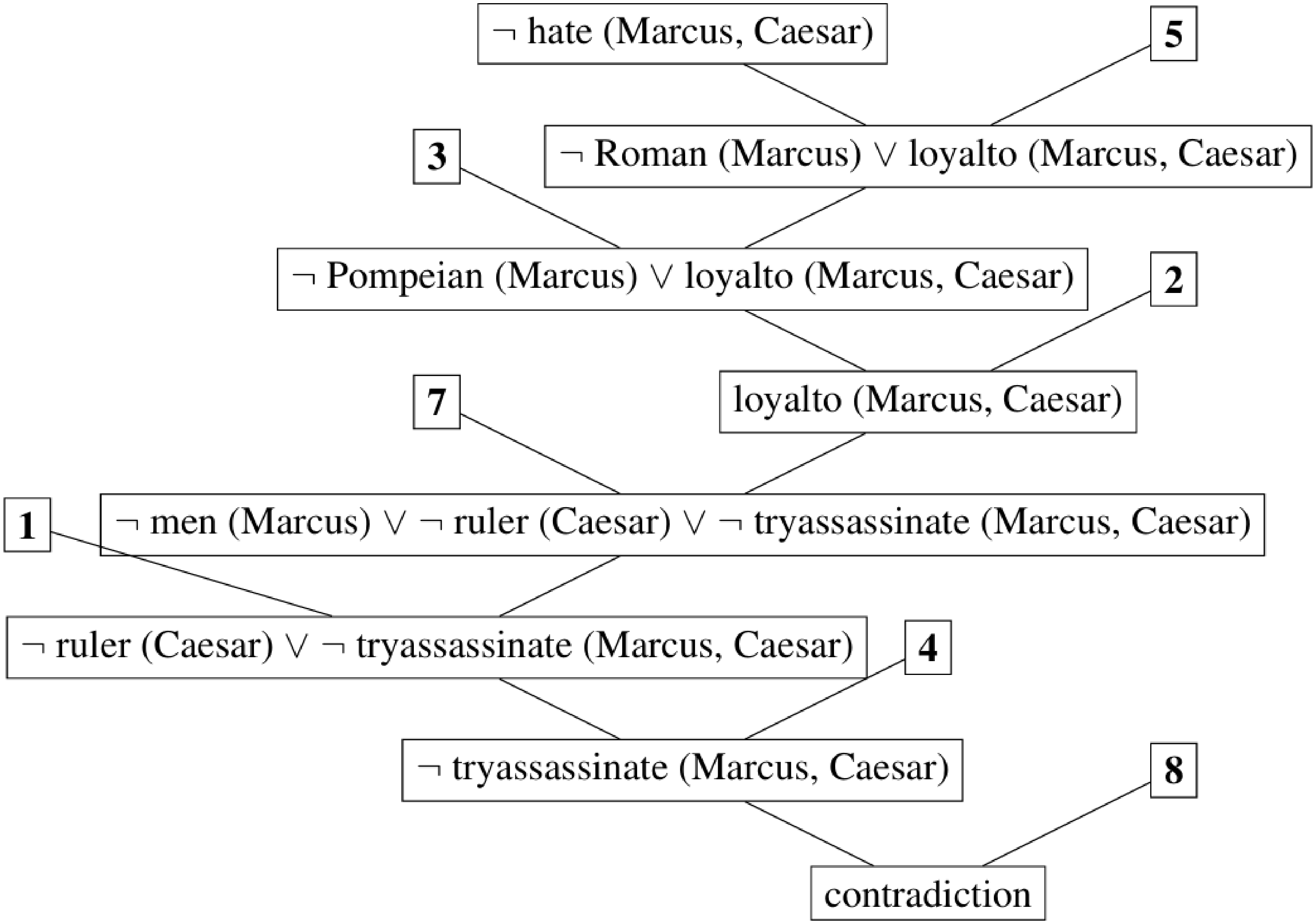
$\neg$ eats(jo, fish)
$\neg$ cat($y$) $\vee$ $\neg$ likes($y$,$z$) $\vee$ eats($y$,$z$)
$\neg$ cat(jo) $\vee$ $\neg$ likes(jo, fish)
cat(jo)
$\neg$ cat($x$) $\vee$ likes($x$, fish)
$\neg$ likes(jo, fish)
$\neg$ cat(jo)
cat(jo)
contradiction

Prove that the conclusion

$$hate(Marcus, Caesar)$$

can be derived from the following premises:

1. $man(Marcus)$
2. $Pompeian(Marcus)$
3. $\neg\, Pompeian(x_1) \vee Roman(x_1)$
4. $ruler(Caesar)$
5. $\neg Roman(x_2) \vee loyalto(x_2, Caesar) \vee hate(x_2, Caesar)$
6. $loyalto(x_3, f(x_3))$
7. $\neg man(x_4) \vee \neg ruler(y_1) \vee \neg tryassassinate(x_4, y_1) \vee loyalto(x_4, y_1)$
8. $tryassassinate(Marcus, Caesar)$


¬ hate (Marcus, Caesar)
5
¬ Roman (Marcus) ∨ loyalto (Marcus, Caesar)
3
¬ Pompeian (Marcus) ∨ loyalto (Marcus, Caesar)
2
loyalto (Marcus, Caesar)
7
1
¬ men (Marcus) ∨ ¬ ruler (Caesar) ∨ ¬ tryassassinate (Marcus, Caesar)
¬ ruler (Caesar) ∨ ¬ tryassassinate (Marcus, Caesar)
4
¬ tryassassinate (Marcus, Caesar)
8
contradiction

- Ram is a good student. All good students have high grades. All good students with high grades are bright. Show that Ram is bright using Resolution.