UCS1524 – Logic Programming

Resolution in First Order Logic

Session Meta Data

Author	**Dr. D. Thenmozhi**
Reviewer	
Version Number	1.2
Release Date	20 July 2022

Session Objectives

- Understanding the concept of resolution in first order logic (FOL)
- Learning the resolution algorithm using unification and substitution

Session Outcomes

- At the end of this session, participants will be able to
  - apply resolution in FOL.

Agenda

- Unification
- Substitution
- Resolution

Ground clauses

- A ground term is a term without occurrences of variables.

$$s(0),\ s(s(0)),\ s(s(s(0)))$$

- A ground formula is a formula in which only ground terms occur.

$$Q(a) \vee P(b)$$

- A predicate clause is a disjunction of atomic formulas.
- A ground clause is a disjunction of ground atomic formulas.
- A ground instance of a predicate clause K is the result of substituting ground terms for the variables of K.

Unification

- Two formulas are said to unify if there are legal instantiations (assignments of terms to variables) that make the formulas in question *identical.*

- The act of unifying is called **unification. The** instantiation that unifies the formulas in question is called a **unifier.**

- There is a simple algorithm called the ***unification algorithm that does this.***

Unification algorithm

- Scan the literals *from left to right, until the first* position is found where in at least two literals, the corresponding symbols are different ;

- If none of these symbols is a variable then
  - output "non-unifiable" and halt
- Else
  - *sub := sub[x/t];*

Unification

- **Example: Unify the formulas *Q(a, y, z) and Q(y, b, c)***
- **Solution:**
  - Since *y in Q(a, y, z) is a different variable than y in Q(y, b, c), rename y in the second formula to* become *y1.*
  - This means that one must unify *Q(a, y, z) with Q(y1, b, c).*
  - An instance of *Q(a, y, z) is Q(a, b, c) and an* instance of *Q(y1, b, c) is Q(a, b, c).*
  - Since these two instances are identical, *Q(a, y, z)* and *Q(y, b, c) unify.*
  - The unifier is *y1* = *a, y* = *b, z* = *c.*

Unification

- **Unification: matching literals and doing** substitutions that resolution can be applied.

- **Substitution: when a variable name is** replaced by another variable or element of the domain.

  – Notation [x/a] means replacing all occurrences of x with a in the formula

  – Example: substitution [x/5] in $p(x) \lor Q(x,y)$ results in $p(5) \lor Q(5,y)$

Unification

- It is an algorithm for determining the substitutions needed to make two predicate logic expressions match.
- A variable cannot be unified with a term containing that variable. The test for it is called the occurs check.
- Example: cannot substitute x for x+y in p(x+y)
- Most applicable when rather than having variables we have whole expressions (terms) evaluating to elements of the domain.
- Example: $x + y$ is a term; when $x,y \in \mathbb{Z}$ and $x + y \in \mathbb{Z}$, with terms we can write formulas such as $p(x+y) \vee Q(y - 2)$

Steps to convert to CNF - Example

(1) Eliminate conditionals $\rightarrow$, using the equivalence

$$p \rightarrow q \equiv \neg p \vee q$$

e.g. $(\exists x)(p(x) \wedge (\forall y)(f(y) \rightarrow h(x,y)))$ becomes

$$(\exists x)(p(x) \wedge (\forall y)(\neg f(y) \vee h(x,y)))$$

(2) Eliminate negations or reduce the scope of negation to one atom.

e.g. $\neg\neg p \equiv p$

$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

$$\neg(\forall x \in S, F(x)) \equiv \exists x \in S, \ \neg F(x)$$

$$\neg(\exists x \in S, F(x)) \equiv \forall x \in S, \ \neg F(x)$$

(3) Standardize variables within a well-formed formula so that the bound or free variables of each quantifier have unique names. e.g. $(\exists x)\neg p(x) \vee (\forall x)p(x)$ is replaced by $(\exists x)\neg p(x) \vee (\forall y)p(y)$

Steps to convert to CNF - Example

(4) Advanced step: if there are existential quantifiers, eliminate them by using Skolem functions

e.g. $(\exists x)p(x)$ is replaced by $p(a)$

$(\forall x)(\exists y)k(x,y)$ is replaced by $(\forall x)\, k(x, f(x))$

(5) Convert the formula to prenex form

e.g. $(\exists x)(p(x) \wedge (\forall y)\,(\neg f(y) \vee h(x,y)))$ becomes

$(\forall y)\,(p(a) \wedge (\neg f(y) \vee h(a,y)))$

(6) Convert the formulas to CNF, which is a conjunctive of clauses. Each clause is a disjunction.

e.g. $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

(7) Drop the universal quantifiers

e.g. the formula in (5) becomes $p(a) \wedge (\neg f(y) \vee h(a,y))$

Steps to convert to CNF - Example

(8) Eliminate the conjunctive signs by writing the formula as a set of clauses

e.g. $p(a) \wedge (\neg f(y) \vee h(a, y))$ becomes $p(a)$, $(\neg f(y) \vee h(a, y))$

(9) Rename variables in clauses, if necessary, so that the same variable name is only used in one clause.

e.g. $p(x) \vee q(y) \vee k(x, y)$ and $\neg p(x) \vee q(y)$ becomes $p(x) \vee q(y) \vee k(x, y)$ and $\neg p(x1) \vee q(y1)$

Algorithm for resolution in FOL

- (i) Convert all the statements of F to clause form.
- (ii) Negate P and convert the result to clause form. Add it to the set of clauses obtained in step (i).
- (iii) Repeat until either a contradiction is found, no progress can be made, or a predetermined amount of effort has been expended.
  - (a) Select two clauses. Call these the parent clauses.

Algorithm for resolution in FOL

– (b) Resolve them together. The resolvent will be the disjunction of all the literals of both parent clauses with appropriate substitutions performed and with the following exception: If there is one pair of literals T1 and T2 such that one of the parent clauses contains T1 and the other contains T2 and if T1 and T2 are unifiable, then neither T1 nor T2 should appear in the resolvent. We call T1 and T2 complementary literals. Use the substitution produced by the 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. If it is not, then add it to the set of clauses available to the procedure.

Example for resolution in FOL

Anyone passing his history exams and winning the lottery is happy.

Anyone who studies or is lucky can pass all his exams.

John did not study but he is lucky.

Anyone who is lucky wins the lottery.

Conclusion: John is happy

Example for resolution in FOL

Anyone passing his history exams and winning the lottery is happy.

$\forall$ **X (pass (X,history) $\wedge$ win (X,lottery) $\rightarrow$ happy (X))**

Anyone who studies or is lucky can pass all his exams.

$\forall$ **X** $\forall$ **Y (study (X) $\vee$ lucky (X) $\rightarrow$ pass (X,Y))**

John did not study but he is lucky.

$\neg$ **study (john) $\wedge$ lucky (john)**

Anyone who is lucky wins the lottery.

$\forall$ **X (lucky (X) $\rightarrow$ win (X,lottery))**

Example for resolution in FOL

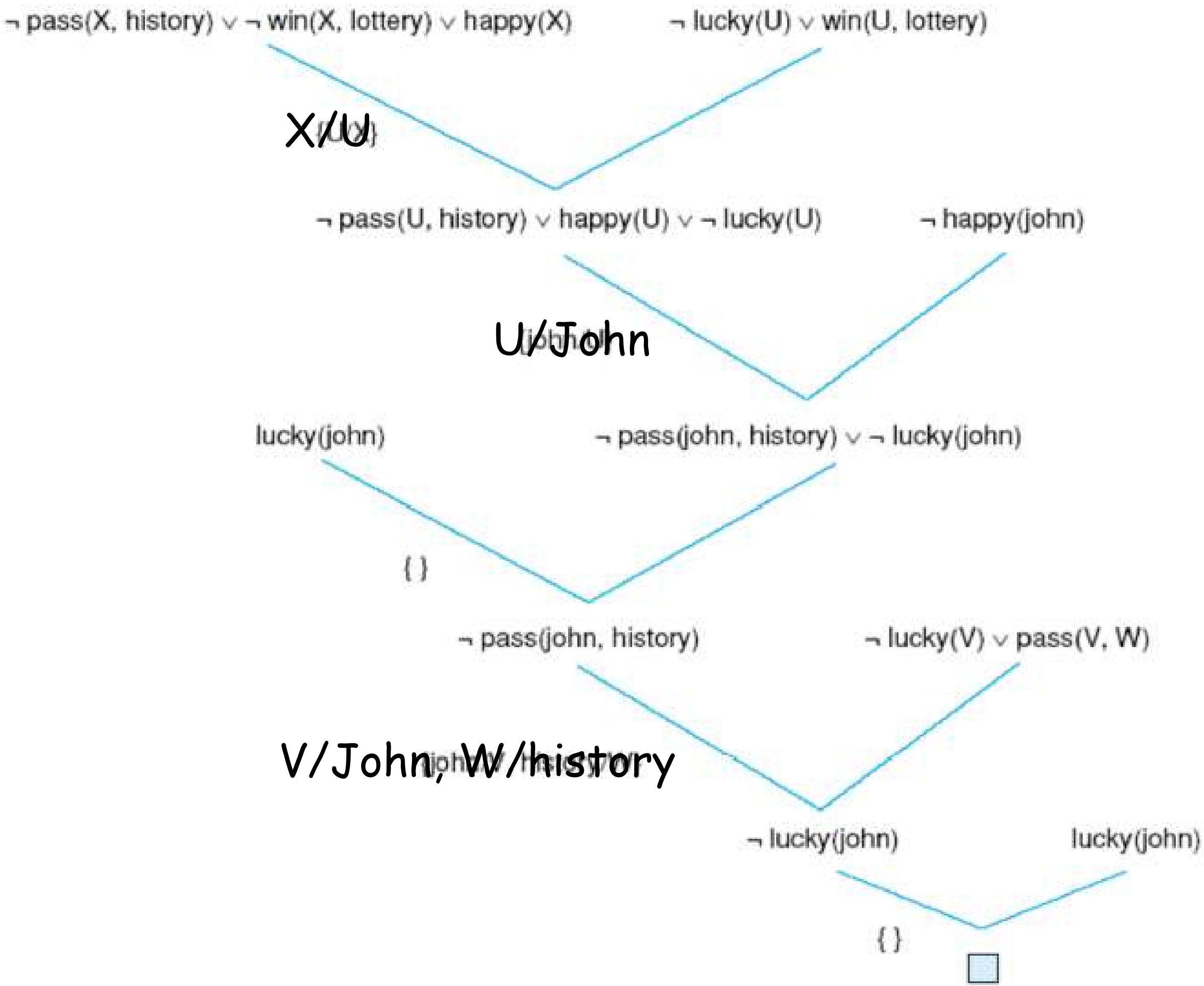
These four predicate statements are now changed to clause form

1. $\neg$ pass (X, history) $\vee$ $\neg$ win (X, lottery) $\vee$ happy (X)
2. $\neg$ study (Y) $\vee$ pass (Y, Z)
3. $\neg$ lucky (W) $\vee$ pass (W, V)
4. $\neg$ study (john)
5. lucky (john)
6. $\neg$ lucky (U) $\vee$ win (U, lottery)

Into these clauses is entered, in clause form, the negation of the conclusion:

7. $\neg$ happy (john)

Example for resolution in FOL



Summary

- Unification
- Substitution
- Resolution

Check your understanding

Set of statements given

1. Marcus was a man.
2. Marcus was a pompeian
3. All pompeians were romans
4. Caesar was a ruler.
5. All romans were either loyal to caesar or hated him.
6. Everyone is loyal to someone.
7. People only try to assassinate rulers they are not loyal to.
8. Marcus tried to assassinate Caesar.
9. All man are people.

a. Convert the given statements to CNF form

b. Solve the statement "Did Marcus hate Caesar?" using resolution

Check your understanding

- Prove that the following clause set is usatisfiable using resolution (Hint: deduce to empty clause)

$$\begin{aligned} F \quad = \quad & \{\{\neg P(x), Q(x), R(x, f(x))\}, \{\neg P(x), Q(x), S(f(x))\}, \{T(a)\}, \\ & \{P(a)\}, \{\neg R(a, z), T(z)\}, \{\neg T(x), \neg Q(x)\}, \{\neg T(y), \neg S(y)\}\} \end{aligned}$$