Topic 9.
The first-order predicate logic

1

How can we represent the following statements?

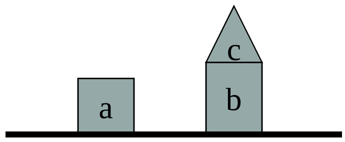
- Fish live in water.
- If he is sick, he needs a doctor.
- The SAT problem cannot be solved in polynomial time unless P is equal to NP.

2

Capabilities of an AI representation language

–Handle qualitative knowledge.

- – clear(c)
- – clear(a)
- – ontable(a)
- – ontable(b)
- – on(c,b)
- – cube(b)
- – cube(a)
- – pyramid(c)



– Allow new knowledge to be inferred from a set of rules and facts.

➢ $\forall X \neg \exists Y on(Y,X) \rightarrow clear(X)$

– Allow representation of general principles as well as specific situations.

➢ A useful AI representation needs variables.

– Capture complex semantic meaning.

➢ hassize(bluebird,small)

➢ hascolor(bluebird,blue)

➢ hascovering(bluebird,feathers)

➢ isa(bluebird,bird)

➢ isa(bluebird,vertebrate)

– Allow for meta-level reasoning.

- An intelligent system must be able to solve problems as well as explain how it solved the problem and why it made certain decisions

Examples of propositional(命題式) statements:

P: Charcoal(木炭) is black.
Q: Sugar is hydrocarbon(碳水化合物).
R: Smith is married.

The symbols, such as P, Q, and R, used to denote propositions are called propositional *symbols*.

5

First order predicate logic (FOPL)(1)

Can you represent the following statements by the propositional logic?

- Every man is mortal.
- Confucius is a man.
- Therefore, Confucius is mortal.

6

First order predicate logic (2)

- Every man is mortal.
  - $\forall X\ (man(X) \rightarrow mortal(X))$
- Confucius is a man.
  - man(confucius)
- Confucius is mortal.
  - mortal(confucius)
- We must prove the following formula is valid in order to prove that Confucius is mortal.
  - $(\forall X\ (man(X) \rightarrow mortal(X))) \wedge man(confucius) \rightarrow mortal(confucius)$

7

The syntax of the predicate calculus (1)

- Symbols begin with a letter and are followed by legal characters.
  - A..Z, a..z.,0..9,_.
- Example
  - Legal symbols

    George, fire3, tom_and_jerry, XXX, friends_of
  - Illegal symbols

    @#$, 3black, “no blank allowed”, hi!!

8

The syntax of the predicate calculus (2)

Predicate calculus symbols include

- **Truth symbols**: true and false.
- **Constant symbols**: the first character is lowercase.
  - Example: george, myMother.
- **Variable symbols**: the first character is uppercase.
  - Example: X, Man.
- **Function symbols**: the first character is lowercase. A function is a mapping that maps a list of constants to a constant.
  - function_name(t_1,t_2,…,t_n)
  - If plus is a function of arity 2 with domain the integer, plus(2,3) is a function expression whose value is the integer 5.

9

The syntax of the predicate calculus (3)

Predicate calculus terms are defined as follows.

- A constant is a term.
- A variable is a term.
- If f is an n-arity function symbol, and t_1, …, t_n are terms, then $f(t_1, \ldots, t_n)$ is a term.
- All terms are generated by applying the above rules.

A predicate calculus term is either a constant, variable or function expression.

Example.

cat, times(2,3), X, blue, mother(jane), kate.

10

The syntax of the predicate calculus (4)

- A predicate names a relationship between zero or more objects in the world.
  - Predicate symbols begin with a lowercase letter.
  - Example:
    like, equals, on, near, part_of
- An *atomic sentence* is a predicate of arity n followed by n terms.
  - Example:
    likes(george, kate)
    likes(X,george) friends(father_of(david),father_of(andrew))
- The truth values, **true** and **false**, are atomic sentences.

11

The syntax of the predicate calculus (5)

First order predicate calculus includes two variable qualifiers ∀ and ∃ that constrain the meaning of a sentence containing a variable.

12

The syntax of the predicate calculus (6)

- ***Universal qualifier*** $\forall$
  - $\forall$ indicates that the sentence is true for all values of the variable.
  - Example: $\forall$X like(X,ice_cream) is true for all values in the domain of the definition of X
- ***Existential qualifier*** $\exists$
  - $\exists$ indicates that the sentence is true for at least one value of the variable in the domain.
  - Example: $\exists$Y friends(Y,peter) is true if there is at least one object, indicated by Y, that is a friend of peter.

13

The syntax of the predicate calculus (7)

- Predicate calculus sentences are defined recursively as follows.

  Every atomic sentence is a sentence.
  1. If **s** is a sentence, then $\neg$ **s** is a sentence.
  2. If $\mathbf{s}_1$ and $\mathbf{s}_2$ are sentences, then $\mathbf{s}_1 \wedge \mathbf{s}_2$ is a sentence.
  3. If $\mathbf{s}_1$ and $\mathbf{s}_2$ are sentences, then $\mathbf{s}_1 \vee \mathbf{s}_2$ is a sentence.
  4. If $\mathbf{s}_1$ and $\mathbf{s}_2$ are sentences, then $\mathbf{s}_1 \rightarrow \mathbf{s}_2$ is a sentence.
  5. If $\mathbf{s}_1$ and $\mathbf{s}_2$ are sentences, then $\mathbf{s}_1 \equiv \mathbf{s}_2$ is a sentence.
  6. If X is a variable and **s** is a sentence, then $\forall$ X **s** is a sentence.
  7. If X is a variable and **s** is a sentence, then $\exists$ X **s** is a sentence.
- **Example.**
  - $\exists$ X foo(X,two,plus(two,three)) $\wedge$ equal(plus(two,three),five) is a sentence.

14

$\neg$weather(rain, monday) $\rightarrow$ go(tom,mountains)
If it does not rain on Monday, Tom will go to the mountains.

$\forall$X(basketball_player(X) $\rightarrow$ tall(X))
All basketball player are tall.

$\exists$X (person(X) $\wedge$ likes(X,anchovies))
Some people like anchovies.

$\neg \exists$ X likes(X,taxes)
Nobody likes taxes.

$\forall$X ($\neg \exists$ Y on(Y,X) $\rightarrow$ clear(X))
For all X, X is clear if there does not exist a Y such that Y is on X.

No unique mapping of sentences into predicate calculus expressions.

15

meaning	definition
either p or q	$(P \vee Q) \wedge \neg (P \wedge Q)$
p unless q	$\neg Q \Rightarrow P$
p because q	$(P \wedge Q) \wedge (Q \Rightarrow P)$
no p is q	$P \Rightarrow \neg Q$

16

A semantics for the predicate calculus (1)

The database of predicate calculus expressions, **each having truth value T**, describes the state of the world.

- Example.
  mother(eve,abel), mother(eve,cain), father(adam,abel), father(adam, cain).

17

A semantics for the predicate calculus (2)

The truth value of an expression E with respect to an interpretation I over a nonempty domain D is determined by:

- The value of a constant is the element of D it is assigned to by I.
- The value of a variable is the set of elements of D it is assigned to by I.
- The value of a function expression is the element of D obtained by evaluating the function for the parameter values assigned by I.
- The value of truth symbol "true" is T and "false" is F.
- The value of an atomic sentence is either T or F, as determined by I.
- The truth value of expressions using $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\equiv$ is determined from the values of their operands as defined previously.
- The value of $\forall$ X **s** is T if **s** is T for all assignments to X under I; otherwise, it is F.
- The value of $\exists$ X **s** is T if there is an assignment to X in I under which **s** is T; otherwise, it is F.

18

Example.
Consider the formula $\forall X \exists Y\ p(X,Y)$.
Let us define an interpretation as follows.

D={1,2} (the domain of X and Y)

Assignment for predicate p.

p(1,1)	p(1,2)	p(2,1)	p(2,2)
T	F	F	T

In this interpretation, for every X, there always exists a Y such that p(X,Y) is true. Therefore, $\forall X \exists Y\ p(X,Y)$ is **T** in this interpretation.

19

Example.
Consider the formula $\forall X(p(X) \rightarrow q(f(X),a))$.
Let us define an interpretation as follows. D={1,2}

Assignment for function f

f(1)	f(2)
2	1

Assignment for constant a

a
1

Assignment for predicates p and q

p(1)	p(2)	q(1,1)	q(1,2)	q(2,1)	q(2,2)
F	T	T	T	F	T

If X=1, $p(1) \rightarrow q(f(1),1)) = p(1) \rightarrow q(2,1) = F \rightarrow F = T$.
If X=2, $p(2) \rightarrow q(f(2),1)) = p(2) \rightarrow q(1,1) = T \rightarrow T = T$.

Therefore, $\forall X(p(X) \rightarrow q(f(X),a))$ is T in this interpretation.

- For a predicate calculus expression **s** and an interpretation I:
  - If **s** has a value of **T** under I and a particular variable assignment, then I is said to satisfy **s** .
  - **s** is satisfiable if and only if there exist an interpretation and variable assignment that satisfy it; otherwise, it is unsatisfiable.
  - If **s** has a value **T** for all possible interpretations, **s** is said to be valid.

A semantics for the predicate calculus (3)

Some relationships between negation and the universal and existential quantifiers.

- $\neg \exists X\ p(X) \equiv \forall X \neg p(X)$
- $\neg \forall X\ p(X) \equiv \exists X \neg p(X)$
- $\exists X\ q(X) \equiv \exists Y\ q(Y)$
- $\forall X\ q(X) \equiv \forall Y\ q(Y)$
- $\mathbf{\forall X\ (p(X) \wedge q(X)) \equiv \forall X\ \ p(X) \wedge \forall Y\ q(Y)}$
- $\mathbf{\exists X\ (p(X) \vee q(X)) \equiv \exists X\ \ p(X) \vee \exists Y\ q(Y)}$

First-order predicate calculus allows quantified variables to refer to objects in the domain of discourse and not to predicates or functions.

Example. $\forall$ (L) L(george,kate) is not a well-formed expression in the first-order predicate calculus.

23

Inference rules

An inference rule is a way to produce additional wffs from other ones.

- Inference rules provide a computationally feasible way to determine whether an expression, a component of an interpretation, logically follows for that interpretation.
- **When every sentence X produced by a set of inference rules, *R*, operating on a set S of expressions logically follows from S, *R* is said to be sound.**
  - Example: modus ponens is a sound inference rule.
- **If *R* can produce every expression that logically follows from S, then *R* is said to be complete.**

24

Useful inference rules

- *Modus ponens*: If P and P $\Rightarrow$ Q are true, then we can infer that Q is true.
- *Modus tollens*: If P $\Rightarrow$ Q is true and Q is false, we can infer $\neg$ P.
- *And elimination*: If P $\wedge$ Q is true, we can infer that P and Q are true.
- *And introduction*: If P and Q are true, then P $\wedge$ Q is true.
- *Universal instantiation*: If $\forall$X p(X) is true and a is from the domain of X, we can infer p(a).

25

If it is raining then the ground will be wet.

–P $\Rightarrow$ Q

It is raining.

–P.

We want to prove that the ground is wet. $(P \wedge (P \Rightarrow Q)) \Rightarrow Q$

–Applying modus ponens, we can infer that Q is true.

Every man is mortal.

–$\forall$X (man(X) $\Rightarrow$ mortal(X))

Confucius is a man.

–man(confucius)

Confucius is mortal.

–Applying universal instantiation, we can infer that man(confucius) $\rightarrow$ mortal(confucius).

–Applying modus ponens, we can infer that mortal(confucius).

26

All dogs are animals.
All animals will die.
Fido is a dog.
We want to prove that "Fido will die" from above statements.

All dogs are animals.
$\forall X\ (dog(X) \Rightarrow animal(X))$
Fido is a dog.
dog(fido).
Modus ponens and universal instantiation ({fido/X}) give animal(fido).
All animals will die.
$\forall Y\ (animal(Y) \Rightarrow die(Y))$.
Modus ponens and universal instantiation ({fido/Y}) give die(fido).

27

$\forall X$ likes(george,X)
$\exists X$ likes(george,X)

If the domain of X is infinite, for example, X ranges over the set of all humans, the test of the sentences may never halt.

28

Decidability

- A logical system is decidable if it is possible to produce an algorithm that will determine whether any wff is a theorem.
  - The propositional logic is decidable.
  - FOPL is not decidable.

29

Monotonicity

- A logical system is monotonic if a valid proof in the system cannot be made invalid by adding additional premises or assumptions.
  - If we can prove

$$\{A, B\} \vdash C$$

    then we can also prove

$$\{A, B, A', B'\} \vdash C$$

    - A' and B' can be anything, including ¬A and ¬B.
- **The propositional logic and FOPL are monotonic.**

30

Deductive, Abduction and Inductive reasoning

- Deductive reasoning $\dfrac{A \quad A \Rightarrow B}{B}$
  - Based on the use of modus pones and the other deductive rules of reasoning.
  - Dealing with certainties.
- Abductive reasoning $\dfrac{B \quad A \Rightarrow B}{A}$
  - Not logically sound
  - Reasoning from observed facts to the best explanation.
- Inductive reasoning
  - Inducing the information in its knowledge base by examples.
    - Induction is the process of inferring the general case from the specific.
  - Useful for dealing with uncertainties.

31

Predicate Calculus Resolution

Example

wffs

Change symbols (Unification)

1. p(f(Y),a) ∨ q(b,c)
2. ¬ p(X,a) ∨ r(X,c) ∨ s(a,b)

Convert arbitrary wffs to clause forms

3. ¬ p(f(Y),a) ∨ r(f(Y),c) ∨ s(a,b) (substituting f(Y) for X in 2)

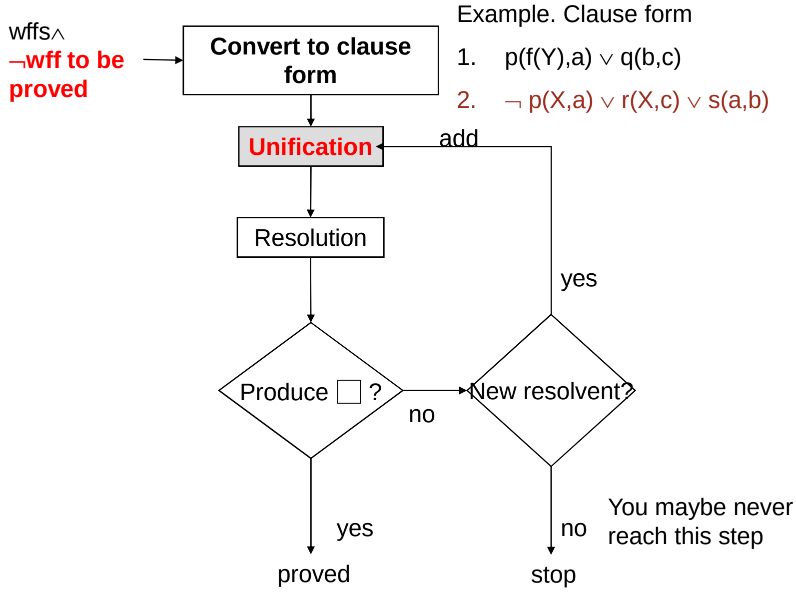
Resolution

4. q(b,c) ∨ r (f(Y),c) ∨ s(a,b) (from resolving 1 and 3)

- The appropriate substitution is computed by a process called *unification*.

Example. Clause form

1. p(f(Y),a) ∨ q(b,c)
2. ¬ p(X,a) ∨ r(X,c) ∨ s(a,b)



Flowchart of resolution refutation in the predicate calculus

Example.

All dogs are animals.
All animals will die.
Fido is a dog.
We want to prove that "Fido will die" from above statements.

All dogs are animals. — CLAUSE FORM

$\forall X\ (dog(X) \Rightarrow animal(X))$ — $\neg dog(X) \vee animal(X)$

All animals will die.

$\forall Y\ (animal(Y) \Rightarrow die(Y))$ — $\neg\ animal(Y) \vee die(Y)$

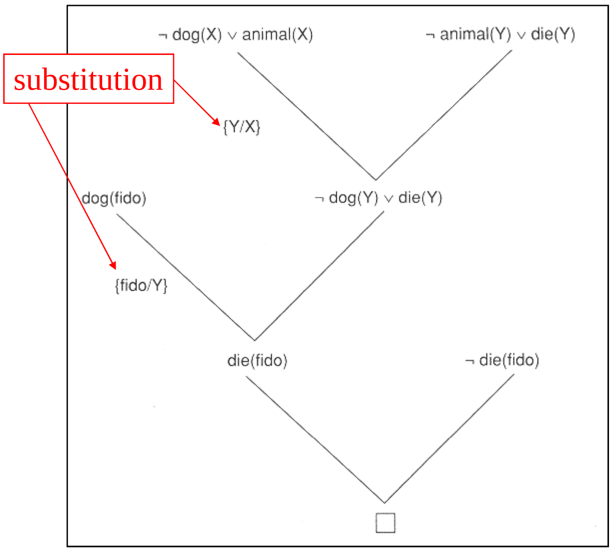
Fido is a dog.

dog(fido). — dog(fido)

Negate the conclusion that Fido will die.

$\neg die(fido)$

35



Resolution proof for the "dead dog" problem

Convert to clause form (1/2)

1. Eliminate implication signs.
   - $P \Rightarrow Q \quad \Rightarrow \neg P \vee Q$
2. Reduce scope of negation signs.
   - $\neg(P \vee Q) \quad \Rightarrow \neg P \wedge \neg Q$
3. Standardize variables.
   - Each quantifier is renamed so that each has its own variable symbol.
   - $(\forall X \neg P(X) \vee (\exists X Q(X))) \quad \Rightarrow (\forall X \neg P(X) \vee (\exists Y Q(Y)))$
4. Move all qualifiers to the left without changing their order.
   - This form is called the *prenex normal* form that all the qualifiers are in front as a prefix and the expression or *matrix* after.
   - $\underbrace{(\forall X)(\exists Y)}_{\text{prefix}} \underbrace{(a(X) \wedge b(Y))}_{\text{matrix}}$

37

Convert to clause form (2/2)

5. Eliminate existential quantifiers.
   - $\forall X \exists Y$ mother(X,Y) skolen function $\forall X$ mother(X,f(X))
     - Y depends on X.
   - $\exists Y$ g(Y) skolen constant g(sk)
     - The existential quantifier being eliminated is not within the scope of any universal quantifiers.
   - Skolen functions must be "new". Put the matrix in conjunctive form.
6. Eliminate universal quantifiers.
7. Convert the expression to the conjunct of disjuncts forms.
8. Call each conjunct a separate clause.
9. Rename variables.

38

Example

$(\forall X)((a(X) \wedge b(X)) \Rightarrow (c(X,I) \wedge (\exists Y)((\exists Z)(c(Y,Z)) \Rightarrow d(X,Y)))) \vee (\forall X)(e(X))$

1. Eliminate implication signs.
   - $(\forall X)(\neg(a(X) \wedge b(X)) \vee (c(X,I) \wedge (\exists Y)((\exists Z)(\neg c(Y,Z)) \vee d(X,Y)))) \vee (\forall X)(e(X))$
2. Reduce the scopes of negation signs.
   - $(\forall \mathbf{X})((\neg a(X) \vee \neg b(X)) \vee (c(X,I) \wedge (\exists \mathbf{Y})((\exists \mathbf{Z})(\neg c(Y,Z)) \vee d(X,Y)))) \vee (\forall \mathbf{X})(e(X))$
3. Standardize the variables; that is, rename variables so that variables bound by different quantifiers have unique names.
   - $\mathbf{(\forall X)}((\neg a(X) \vee \neg b(X)) \vee (c(X,I) \wedge \mathbf{(\exists Y)((\exists Z)}(\neg c(Y,Z)) \vee d(X,Y)))) \vee \mathbf{(\forall W)}(e(W))$
4. Convert to prenex normal form.
   - $(\forall X)(\exists \mathbf{Y})(\exists \mathbf{Z})(\forall W)((\neg a(X) \vee \neg b(X)) \vee \mathbf{(c(X,I) \wedge (\neg c(Y,Z)) \vee d(X,Y)))}) \vee (e(W))$
5. Eliminate existential qualifiers.
   - $(\forall \mathbf{X}) (\forall \mathbf{W})((\neg a(X) \vee \neg b(X)) \vee (c(X,I) \wedge (\neg c(f(X),g(X))) \vee d(X, f(X))))) \vee (e(W))$
6. Remove all universal qualifiers.
   - $((\neg a(X) \vee \neg b(X)) \vee (c(X,I) \wedge (\neg c(f(X),g(X))) \vee d(X, f(X))))) \vee (e(W))$
7. Transform to conjunctive normal form.
   - $(\neg a(X) \vee \neg b(X) \vee c(X,I) \vee e(W)) \wedge (\neg a(X) \vee \neg b(X) \vee \neg c(f(X),g(X)) \vee d(X, f(X)) \vee e(W))$
8. Call each conjunct a separate clause.
   a) $\neg a(X) \vee \neg b(X) \vee c(X,I) \vee e(W)$
   b) $\neg a(X) \vee \neg b(X) \vee \neg c(f(X),g(X)) \vee d(X, f(X)) \vee e(W)$
9. Standardize the variables apart again.
   a) $\neg a(X) \vee \neg b(X) \vee c(X,I) \vee e(W)$
   b) $\neg a(U) \vee \neg b(U) \vee \neg c(f(U),g(U)) \vee d(U, f(U)) \vee e(V)$

Unification

Unification is an algorithm for determining the substitutions needed to make two predicate calculus expressions match.

40

Substitution (1)

Definition.

- A **substitution** is a finite set of the form $\{t_1/v_1,\ldots,t_n/v_n\}$, where every v_i is a variable, every t_i is a term different from v_i, and no two elements in the set have the same variable after the stroke symbol.
  - When $t_1,\ldots,t_n$ are ground terms, the substitution is called a ground substitution.
- ***X/Y indicates that X is substituted for Y in the original expression.***

41

Examples.

- Some instance of the expression foo(X,a,goo(Y)) generated by legal substitutes are as follows.
  1. foo(fred,a,good(Z)) foo(X,a,goo(Y)) {fred/X,Z/Y}
  2. foo(W,a,goo(jack)) foo(X,a,goo(Y)) {W/X,jack/Y}
  3. foo(Z,a,goo(moo(Z))) foo(X,a,goo(Y)) {Z/X,moo(Z)/Y}
- ***X/Y indicates that X is substituted for Y in the original expression.***

42

Substitution (2)

Definition.

- Let $\theta=\{t_1/v_1,\ldots,t_n/v_n\}$ be a substitution and E be an expression. Then $E\theta$ is an expression obtained from E by **replacing simultaneously each occurrence of the variable v_i, $i=1,2,\ldots n$, in E by the term t_i.**
- $E\theta$ is called an instance of E.
- Example

  Let $\theta=\{a/X,f(b)/Y,c/Z\}$ and $E=p(X,Y,Z)$. Then $E\theta = p(a,f(b),c)$.

43

Composition of unification substitutions

Definition.

- Let $\theta=\{t_1/x_1,\ldots,t_n/x_n\}$ and $\lambda=\{u_1/y_1,\ldots,u_m/y_m\}$ be two substitutions. Then the *composition* of θ and λ is the substitution, denoted by $\theta\lambda$, that is obtained from the set

  $$\{t_1\lambda/x_1,\ldots, t_n\lambda/x_n, u_1/y_1,\ldots,u_m/y_m\}$$

  by deleting any element $t_j\lambda/x_j$ for which $t_j\lambda=x_j$, and any element u_i/y_i such that y_i is among $\{x_1,\ldots,x_n\}$.

- The composition of substitutions is associative.
  - $(\theta\lambda)\mu= \theta(\lambda\mu)$

44

Example.

- {f(Y)/X,Z/Y} {a/X,b/Y,Y/Z}

  {f(b)/X,~~Y/Y~~,~~a/X~~,b/Y,Y/Z}

  ={f(b)/X ,b/Y,Y/Z}
- {g(X,Y)/Z} {a/X,b/Y,c/W,d/Z}={g(a,b)/Z, a/X,b/Y,c/W,~~d/Z~~}

45

Unifier

Definition.

- A substitution θ is called a **unifier** for a set $\{E_1, E_2, \ldots, E_k\}$ if and only if $E_1\theta = E_2\theta = \ldots = E_k\theta$. The set $\{E_1, E_2, \ldots, E_k\}$ is said to be **unifiable** if there is a unifier for it.

Example.

The set {p(a,Y),p(X,f(b)} is unifiable because the substitution θ={a/X,f(b)/Y} is a unifier for the set.

{p(a,Y)}θ={p(a,f(b))}

{p(X,f(b)}θ={p(a,f(b))}

46

Most General Unifier (MGU)

Definition.

A unifier σ for a set $\{E_1, E_2, \ldots, E_k\}$ of expressions is a **most general unifier** (**mgu**) if and only if for each unifier θ for the set there is a substitution λ such that $\theta=\sigma\lambda$.

47

Unification algorithm

The unification algorithm can

- find the most general unifier for a finite unifiable set of nonempty expressions.
- determine whether or not a set of expressions is unifiable.

48

Predicate calculus syntax	List syntax
p(a,b)	**(p a b)**
p(f(a),g(X,Y))	**(p (f a) (g X Y))**

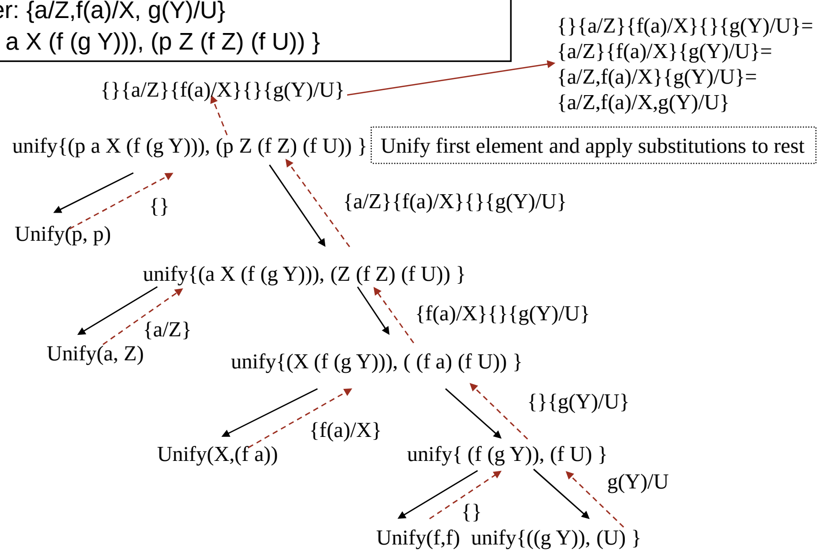
```
function unify(E1,E2)
begin // E1 and E2 are in the list representation
        case
        E1 and E2 are constants or the empty list:
                if E1=E2 then return {} else return FAIL;
        either E1 or E2 is empty then return FAIL.
        E1 is a variable:
                if E1 occurs in E2 then return FAIL else return {E2/E1}.
        E2 is a variable:
                if E2 occurs in E1 then return FAIL else return {E1/E2}.
        otherwise:
                begin
                        HE1 := first element of E1;
                        HE2 := first element of E2;
                        SUBS1 := unify(HE1,HE2);
                        if SUBS1== FAIL then return FAIL.
                        TE1 := apply(SUBS1, rest of E1); // substitution: (rest of E1)SUBS1
                        TE2 := apply(SUBS1, rest of E2); // substitution:(rest of E2)SUBS1
                        SUBS2 := unify(TE1,TE2);
                        if SUBS2==FAIL then return FAIL.
                        else return composition(SUBS1,SUBS2); // composition
                end
        end
end
```

Example

Find an mgu for W={p(a,X,f(g(Y))),p(Z,f(Z),f(U))}.

Answer: {a/Z,f(a)/X, g(Y)/U}

W={(p a X (f (g Y))), (p Z (f Z) (f U)) }



50

Example
Consider the following set of formulas:
F_1: $\forall X\ (c(X) \Rightarrow (w(X) \wedge r(X)))$
F_2: $\exists\ X\ (c(X) \wedge o(X))$
G: $\exists\ X\ (o(X) \wedge r(X))$.
Show that G is a logical consequence of F_1 and F_2.

1.	$\neg c(X) \vee w(X)$	from F_1
2.	$\neg c(Y) \vee r(Y)$	
3.	$c(a)$	from F_2
4.	$o(a)$	
5.	$\neg o(Z) \vee \neg\, r(Z)$	from $\neg G$
6.	$\neg r(a)$	(5) and (4)
7.	$r(a)$	(3) and (2)
8.	$\square$	(6) and (7)

Therefore, G is a logical consequence of F_1 and F_2.

Example:
Premises:
1. The custom officials searched everyone who entered this country who was not a VIP.
2. Some of the drug pushers entered this country and they were only searched by drug pushers.
3. No drug pusher was a VIP.

Conclusions: Some of the officials were drug pushers.

Let
e(X) mean “X entered this country,”
v(X) mean “X was a VIP,”
s(X,Y) mean “Y searched X,”
c(X) mean “X was a custom official,” and
p(X) mean “X was a drug pusher.”

Premises
$\forall X(e(X) \wedge \neg v(X) \Rightarrow \exists Y\ (s(X,Y) \wedge c(Y)))$
$\exists\ X(e(X) \wedge p(X) \wedge \forall Y\ (s(X,Y) \Rightarrow p(Y)))$
$\forall X(p(X) \Rightarrow \neg v(X))$
Conclusions
$\exists X(c(X) \wedge p(X))$.

Premises

∀X(e(X) ∧ ¬v(X) ⇒ ∃Y (s(X,Y) ∧c(Y)))

∃ X(e(X) ∧ p(X) ∧ ∀Y (s(X,Y) ⇒ p(Y)))

∀X(p(X) ⇒ ¬v(X))

Conclusions

∃X(c(X) ∧p(X)).

The resolution proof is as follows.

(1) ¬e(X) ∨v(X) ∨s(X,f(X))
(2) ¬e(Y) ∨v(Y) ∨c(f(Y))
(3) p(g(Y1))
(4) e(g(Y2))
(5) ¬s(g(Y3),Y3) ∨p(Z)
(6) ¬p(W) ∨¬v(W)
(7) ¬p(T) ∨¬c(T)
(8) ¬v(g(Y1)) from (3) and (6) {g(Y1)/W}
(9) v(g(Y2)) ∨c(f(g(Y2))) from (2) and (4) {g(Y2)/Y}
(10) c(f(g(Y2))) from (8) and (9) {Y2/Y1}
(11) v(g(Y2)) ∨s(g(Y2),f(g(Y2))) from (1) and (4) {g(Y2)/X}
(12) s(g(Y2),f(g(Y2))) from (8) and (11) {Y2/Y1}
(13) ¬e(g(Y2) ∨v(g(Y2)) from (1) and (12) {g(Y2)/X}
(14) ¬e(g(Y2) from (8) and (13) {Y2/Y1 }
(15) □ from (4) and (14)

Predicate-calculus resolution

- Predicate-calculus resolution is sound.
- Predicate-calculus resolution is refutation complete.

54

Example.

Anyone passing his history exams and winning the lottery is happy. But 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. Is John happy?

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.

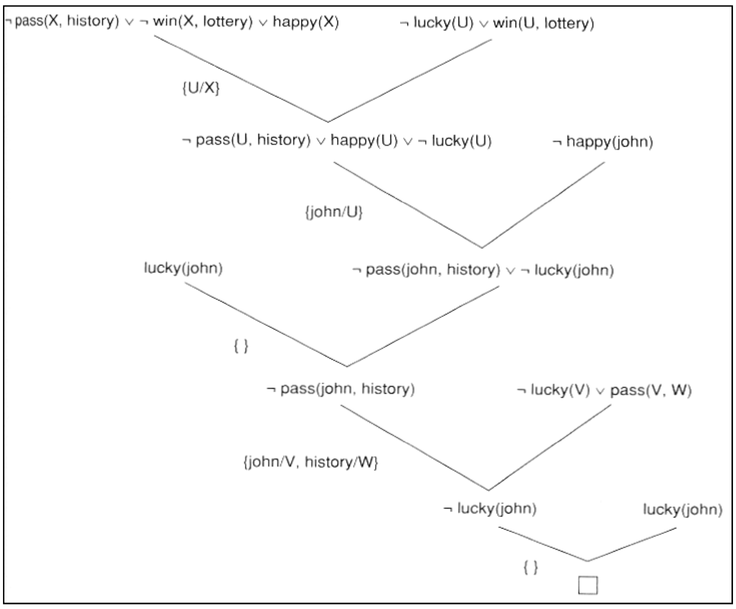
$\forall$ X (lucky (X) $\rightarrow$ win (X,lottery))

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)

THE CLAUSE FORM OF THE NEGATION OF CONCLUSION

7. $\neg$ happy (john)



One resolution refutation for the "happy student" problem

Example.

All people who are not poor and are smart are happy. Those people who read are not stupid. John can read and is wealthy. Happy people have exciting lives. Can anyone be found with an exciting life

We assume $\forall X$ (smart (X) $\equiv \neg$ stupid (X)) and $\forall Y$ (wealthy (Y) $\equiv \neg$ poor (Y)),

$\forall X$ ($\neg$ poor (X) $\wedge$ smart (X) $\rightarrow$ happy (X))

$\forall Y$ (read (Y) $\rightarrow$ smart (Y))

read (john) $\wedge \neg$ poor (john)

$\forall Z$ (happy (Z) $\rightarrow$ exciting (Z))

Negation of the conclusion

$\neg \exists$ W (exciting (W))

CLAUSE FORM

poor (X) $\vee \neg$ smart (X) $\vee$ happy (X)

$\neg$ read (Y) $\vee$ smart (Y)

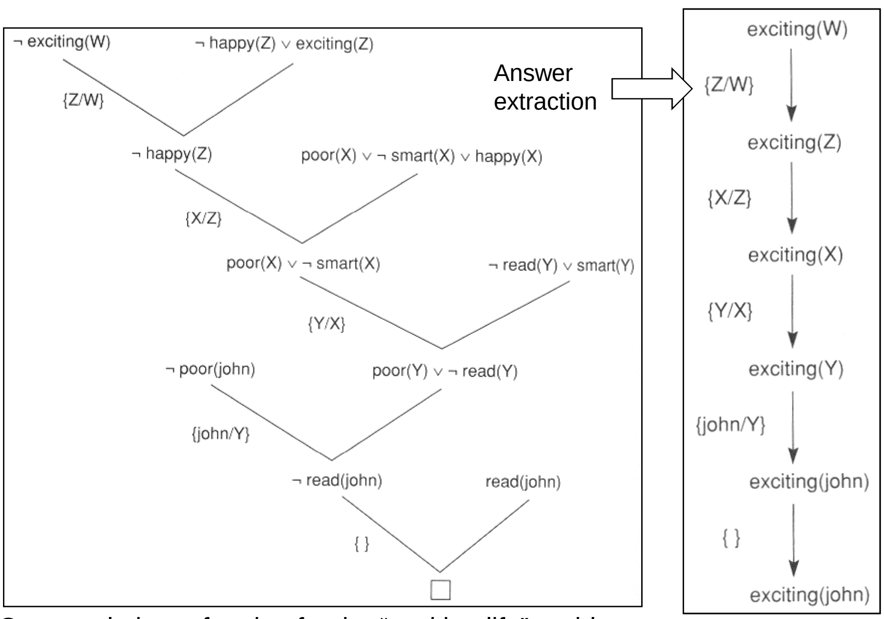
read (john)

$\neg$ poor (john)

$\neg$ happy (Z) $\vee$ exciting (Z)

$\neg$ exciting (W)

57



One resolution refutation for the "exciting life" problem

{Z/W}{X/Z}{Y/X}{john/Y}{}={john/W,john/Z,john/X,john/Y}

Example.

Fido the dog goes whenever John, his master, goes. John is at the library. Where is Fido?

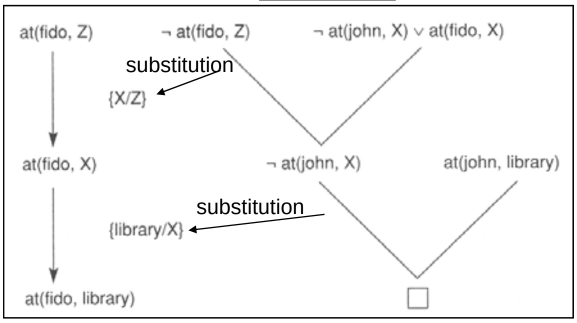
at (john,X) → at (fido,X)
at (john,library)

¬ at (john,Y) ∨ at (fido,Y)
at (john,library)

¬ at (fido,Z)

Negation of the conclusion

¬ at (fido,Z)



59

{X/Z}{library/X}={library/Z,library/X}

Example.

Everyone has a parent. The parent of a parent is a grandparent. Given the person John, prove that John has a grandparent.

Everyone has a parent:

(∀ X)(∃ Y) p(X,Y)

A parent of a parent is a grandparent.

(∀ X)(∀ Y)(∀ Z) p(X,Y) ∧ p(Y,Z) → gp(X,Z)

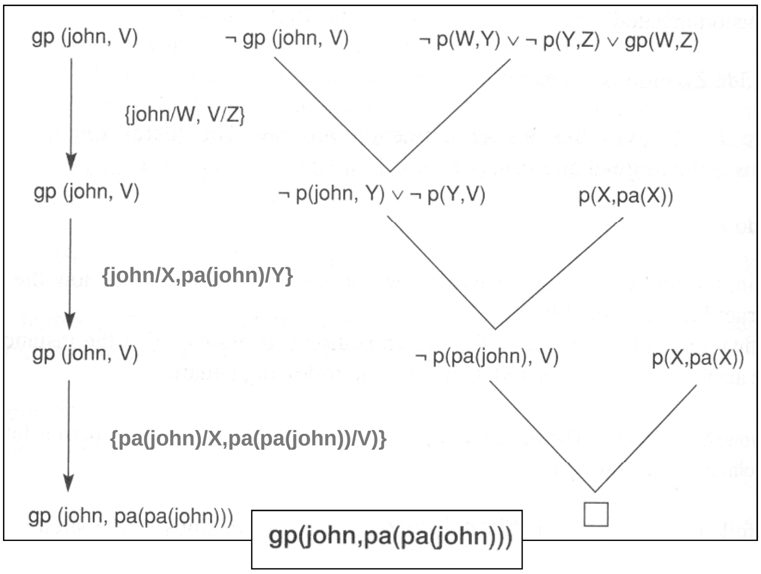
Conclusion

∃ (W)(gp(john,W))

CLAUSE FORM

p(X,pa(X))
¬ p(W,Y) ∨ ¬ p(Y,Z) ∨ gp(W,Z)
¬ gp(john,V)

60



{john/W,V/Z}{john/X,pa(john)/Y}{pa(john)/X,pa(pa(john))/V}=
{john/W,pa(X)/Z,john/X,pa(john)/Y}{pa(john)/X,pa(pa(john))/V)}=
{john/W, pa(pa(john)) /Z,john/X,pa(pa(john))/Y, pa(pa(john))/V}