40417: Artificial Intelligence

Inference in Predicate Logic

Instructor: Gholamreza Ghassem-Sani

Sharif University of Technology

Sentences vs. well-formed formulae

- A *free variable* is a variable that is not bound by a quantifier.
  - $\exists y$ Likes(x, y), here, x is free and y is bound.
  - $\forall x$ (P(x)) $\wedge$ Q(x), here, the first appearance of x is bound but the second one is free.
- A *well-formed formula* is any formula with the syntax of FOL.
- A *sentence* is a *w.f.f* that contains no free variable.
- Thus, in a sentence, every variable must be within the scope of a quantifier.

Rules of Inference

- In addition to the inference rules of the proposition logic, there are four new rules that can be used in FOL.

1. Universal instantiation (-∀)

   $\forall x\ P(x)$

   $\therefore\ P(c)$ if $c \in U$

   U, called the *universe of discourse* or *domain of discourse*, identifies the set of values that each variable can take

2. Existential generalization (+∃)

   $P(c)$ for some element $c \in U$

   $\therefore\ \exists x\ P(x)$

3. Universal generalization (+∀)

   $P(c)$ for an arbitrary $c \in U$

   $\therefore\ \forall x\ P(x)$

   c must not be free in the assumptions whose scope contain P(c)

4. Existential instantiation (-∃)

   $\exists x\ P(x)$

   $\therefore\ P(c)$ for some element $c \in U$

   c must not be free in any previous line or in -∃ conclusion

- 1 and 2 are straightforward; but usage of 3 and 4 have some preconditions.

[40-417, SUT]

Universal generalization - example

1. $\forall x(\text{Ontable}(x, T1) \Rightarrow \text{Upturned}(x, T2))$
2. $\forall x \forall y(\text{Upturned}(x, y) \Rightarrow \text{Empty}(x, y))$

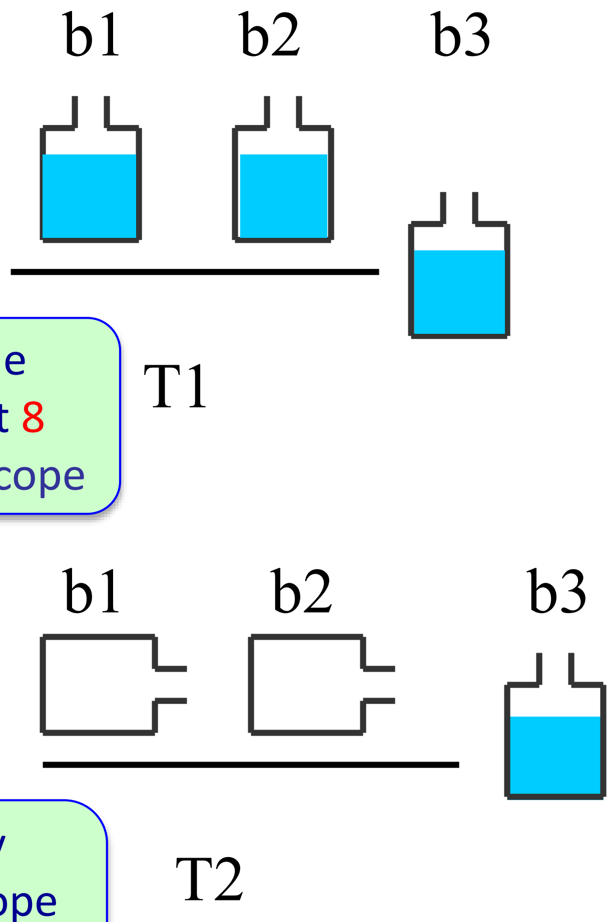
1 is a Premise
2 is an Axiom

3.	$\text{Ontable}(b2, T1)$	Assumption
4.	$\text{Ontable}(b2, T1) \Rightarrow \text{Upturned}(b2, T2)$	$- \forall$, 1
5.	$\text{Upturned}(b2, T2)$	$- \Rightarrow$, 3, 4
6.	$\text{Upturned}(b2, T2) \Rightarrow \text{Empty}(b2, T2)$	$- \forall$, 2
7.	$\text{Empty}(b2, T2)$	$- \Rightarrow$, 5, 6
8.	$\text{Ontable}(b2, T1) \Rightarrow \text{Empty}(b2, T2)$	$+ \Rightarrow$, 7
9.	$\forall x(\text{Ontable}(x, T1) \Rightarrow \text{Empty}(x, T2))$	$+ \forall$, 8

b2 is free in the assumption but 8 is not within its scope

Constants are regarded as free variables in checking the precondition of $+\forall$ and $-\exists$

b2 is not free in any assumption whose scope contains sentence 8; so, we can apply $+\forall$



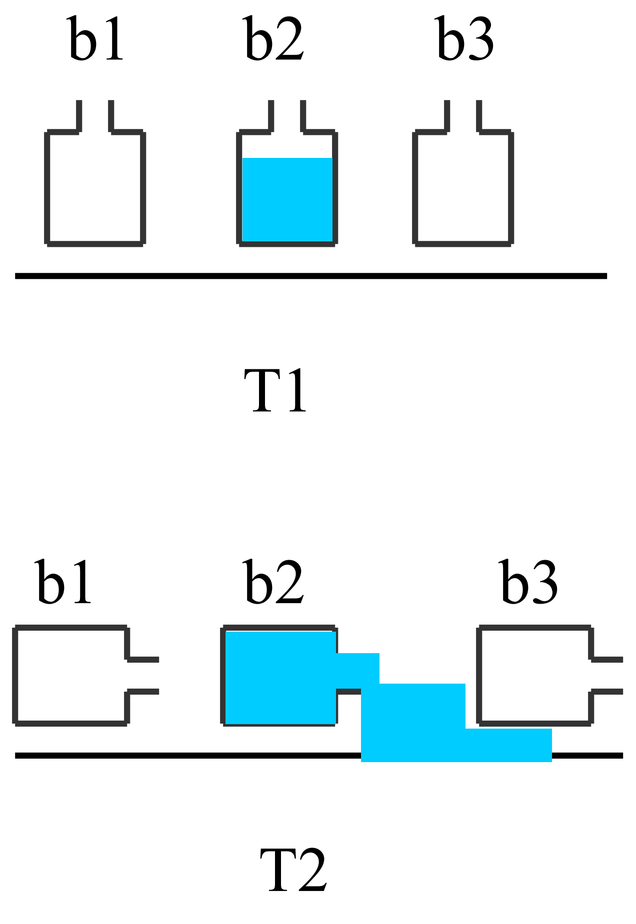
[40-417, SUT]

Existential instantiation - example

1. $\forall x(Bottle(x,T1) \Rightarrow Upturned(x, T2))$
2. $\forall x \forall y(Upturned(x, y) \Rightarrow Empty(x, y))$
3. $\forall x(Full(x, T1) \wedge Empty(x, T2) \Rightarrow Wet(Floor))$
4. $\exists x\ (Bottle(x, T1)\ \&\ Full(x, T1))$
5. $Bottle(b1, T1) \wedge Full(b1, T1))$ — $-\exists$ assumption, 4
6. $Bottle(b1, T1)$ — $-\wedge$, 5
7. $Full(b1, T1)$ — $-\wedge$, 5
8. $Upturned\ (b1, T2)$ — $-\forall$, 1 and $-\Rightarrow$, 6
9. $Empty(b1, T2)$ — $-\forall$, 2 and $-\Rightarrow$, 7
10. $Full(b1, T1) \wedge Empty(b1, T2)$ — $+\wedge$, 7, 9
11. $Wet(Floor)$ — $-\forall$, 3 and $-\Rightarrow$, 10
12. $Wet(Floor)$ — $-\exists$ conclusion, 11

1, 3 and 4 are Premises
2 is an Axioms

b1 is not free in any previous line or in $-\exists$ conclusion



[40-417, SUT]

Existential instantiation - example 2

- Prove $(\forall x(P(x) \Rightarrow Q)) \Rightarrow (\exists x P(x) \Rightarrow Q))$

1.	$\forall x(P(x) \Rightarrow Q)$	Assumption
2.	$\exists x P(x)$	Assumption
3.	$P(x)$	-∃ Assumption, 2
4.	$P(x) \Rightarrow Q$	-∀, 1
5.	Q	-⇒ 3, 4
6.	Q	-∃ Conclusion, 5
7.	$\exists x P(x) \Rightarrow Q$	+⇒ 2, 6
8.	$(\forall x(P(x) \Rightarrow Q)) \Rightarrow (\exists x P(x) \Rightarrow Q)$	+⇒ 1, 7

x is not free in any previous lines or in the rule's conclusion

[40-417, SUT]

Universal generalization - example 2

- Prove $(\exists x P(x) \Rightarrow Q) \Rightarrow (\forall x(P(x) \Rightarrow Q))$

1.	$(\exists x P(x) \Rightarrow Q)$	Assumption
2.	$P(x)$	Assumption
3.	$\exists x P(x)$	$+\exists$
4.	Q	$-\Rightarrow$ 1, 3
5.	$P(x) \Rightarrow Q$	$-\Rightarrow$ 2, 4
6.	$\forall x(P(x) \Rightarrow Q)$	$+\forall$, 5
7.	$(\exists x P(x) \Rightarrow Q) \Rightarrow (\forall x(P(x) \Rightarrow Q))$	$+\Rightarrow$ 1, 6

x is free in line 2, but line 5 is not within the scope of the assumption in line 2

x is not free in the assumptions whose scope include sentence in line 5; so we can apply $+\forall$

Resolution in FOL

- Resolution inference rule:

$$\frac{l_1 \vee \ldots \vee l_k \qquad\qquad m_1 \vee \ldots \vee m_n}{\sigma(l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n)}$$

where Unify(l_i, m_j) = σ, and $\sigma(l_i)$ is the negation of $\sigma(m_j)$.

- E.g.,

$$\frac{Animal(F(x)) \vee Loves(G(x), x) \qquad\qquad \neg Loves(u, v) \vee \neg Kills(u, v)}{Animal(F(x)) \vee \neg Kills(G(x), x)}$$

$\sigma = \{u/G(x), v/x\}$

- All variables assumed universally quantified.

Factoring

- *Factoring* is required to make *Resolution* a complete procedure

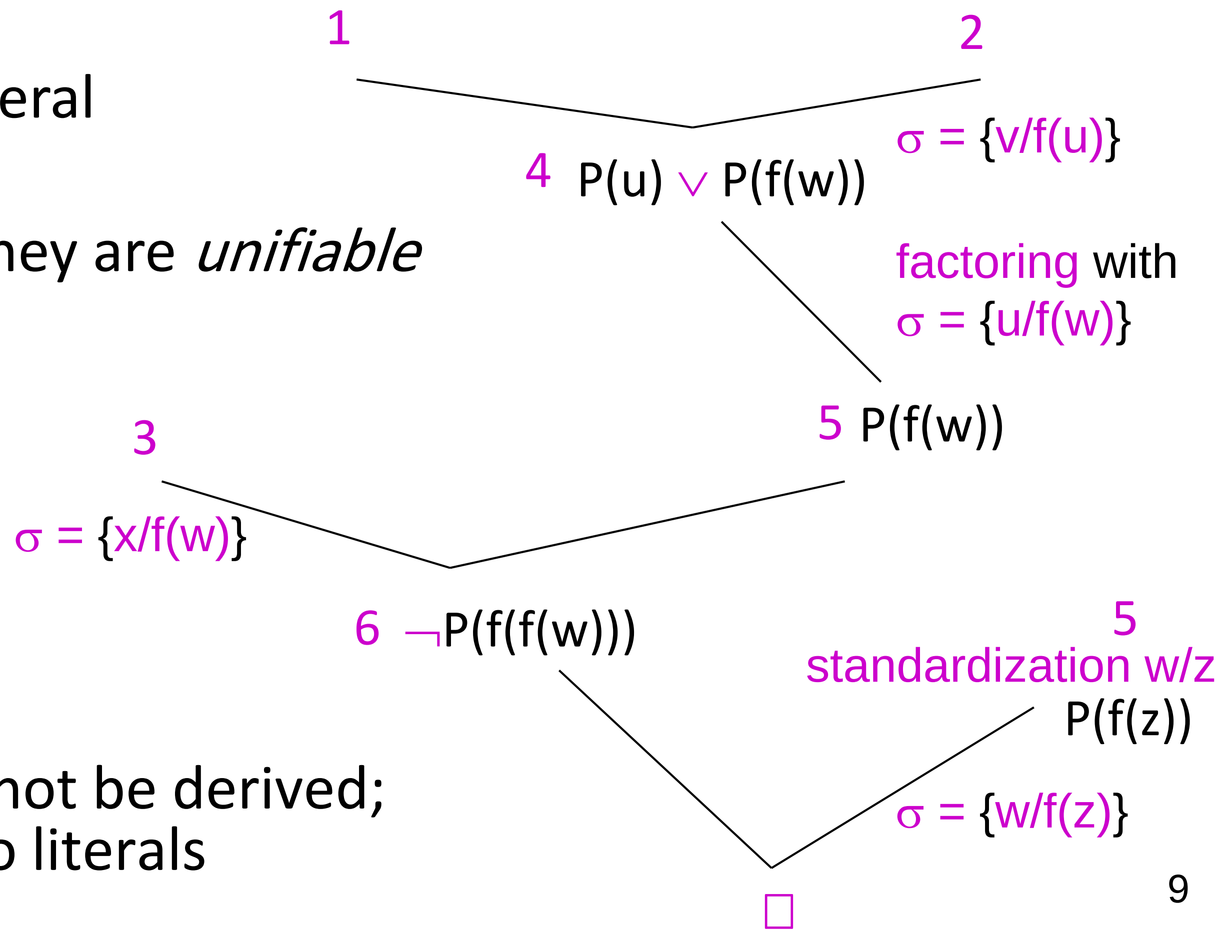
  - Factoring is the removal of redundant literal

  - Factoring reduces two literals to one if they are *unifiable*

    E.g., KB (in CNF):

    1. P(u) ∨ P(f(u))
    2. ¬P(v) ∨ P(f(w))
    3. ¬P(x) ∨ ¬P(f(x))

  - without factoring, the empty clause cannot be derived; because every resolvent would have two literals



[40-417, SUT]

Conversion to Clause Form

- $\forall x\, [Roman(x) \wedge know(x, Mark)] \Rightarrow [hate(x, Caesar) \vee (\forall y\, (\exists z\, hate(y, z)) \Rightarrow thinkcrazy(x, y))]$

1) Replace $\Leftrightarrow$ by two $\Rightarrow$ and Replace $\alpha \Rightarrow \beta$ by $\neg\alpha \vee \beta$

- $\forall x \neg [Roman(x) \wedge know(x, Mark)] \vee [hate(x, Caesar) \vee (\forall y \neg(\exists z\, hate(y, z)) \vee thinkcrazy(x, y))]$

$\neg\forall x\, P(x) \equiv \exists x \neg P(x)$
$\neg\exists x\, P(x) \equiv \forall x \neg P(x)$

2) Move $\neg$ inwards using de Morgan's rules and double-negation

- $\forall x\, [\neg Roman(x) \vee \neg know(x, Mark)] \vee [hate(x, Caesar) \vee (\forall y\, \forall z \neg hate(y, z) \vee thinkcrazy(x, y))]$

3) Standardize variables so that each quantifier binds a unique variable

4) Move all quantifiers to the left without changing their relative order

The result is called prenex normal form

- $\forall x\, \forall y\, \forall z\, [\neg Roman(x) \vee \neg know(x, Mark)] \vee [hate(x, Caesar) \vee (\neg hate(y, z) \vee thinkcrazy(x, y))]$

6) Drop the prefix

The result is called skolem normal form

5) Eliminate existential quantifiers through the use of Skolem functions and constants

9) Standardize apart the variables of clauses no two clauses share any variable

- $[\neg Roman(x) \vee \neg know(x, Mark)] \vee [hate(x, Caesar) \vee (\neg hate(y, z) \vee thinkcrazy(x, y))]$

7) Distribute $\vee$ over $\wedge$

The result is called conjunctive normal form

8) Create a separate clause for each conjunct

The result is called clausal form

[40-417, SUT]

Skolemization

- In Skolemization, we eliminate existential quantifiers through the use of Skolem functions and constants.

- Replacing every existentially quantified variable y with a term $f(x_1, \ldots, x_n)$ whose function symbol f is new.

- If the formula is in prenex normal form, then $x_1, \ldots, x_n$ are the variables that are universally quantified and whose quantifiers precede that of y.

E.G.,
  - Prenex normal form:
  - Skolemized normal form

In the 2nd order logic, ∀ and ∃ can be used to quantify functions, too.

In FOL, the prenex normal form does not necessarily entail the Skolemized one; but the reverse entailment relation holds.

- Skolemization does not necessarily preserve the logical equivalence!

$\forall x\, \exists z\, \text{father-of}(x, z) \;\not\Rightarrow\; \forall x\, \text{father-of}(x, f1(x))$

$\Leftarrow$

- But, in the Second Order Logic, it does!

$\forall x\, \exists z\, \text{father-of}(x, z) \;\Leftrightarrow\; \exists f1 \forall x\, \text{father-of}(x, f1(x))$

Importance of standardizing variables apart

Consider KB:

1. father(x, y) ⇒ ¬woman(x)
2. mother(x, y) ⇒ woman(x)
3. mother(Chris, Mary)
4. father(Chris, Bill)

There is a contradiction in this KB.

KB in CNF:

1. ¬father(x, y) ∨ ¬woman(x)
2. ¬mother(x, y) ∨ woman(x)
3. mother(Chris, Mary)
4. father(Chris, Bill)

The contradiction is not detected because variables have not been standardized apart.

KB in standardized CNF:

1. ¬father(x1, y1) ∨ ¬woman(x1)
2. ¬mother(x2, y2) ∨ woman(x2)
3. mother(Chris, Mary)
4. father(Chris, Bill)

Now, they have been standardized.

1 2

¬father(x, y) ∨ ¬mother(x, y)

3

x/Chris, y/Mary

¬father(Chris, Mary)

1 2

x2/x1

¬father(x1, y1) ∨ ¬mother(x1, y2)

3

x1/Chris, y2/Mary

¬father(Chris, y1)

4

y1/

□

This time, a more general (stronger) clause is inferred

Note that variables of clauses are implicitly universally quantified

[40-417, SUT]

Resolution - example

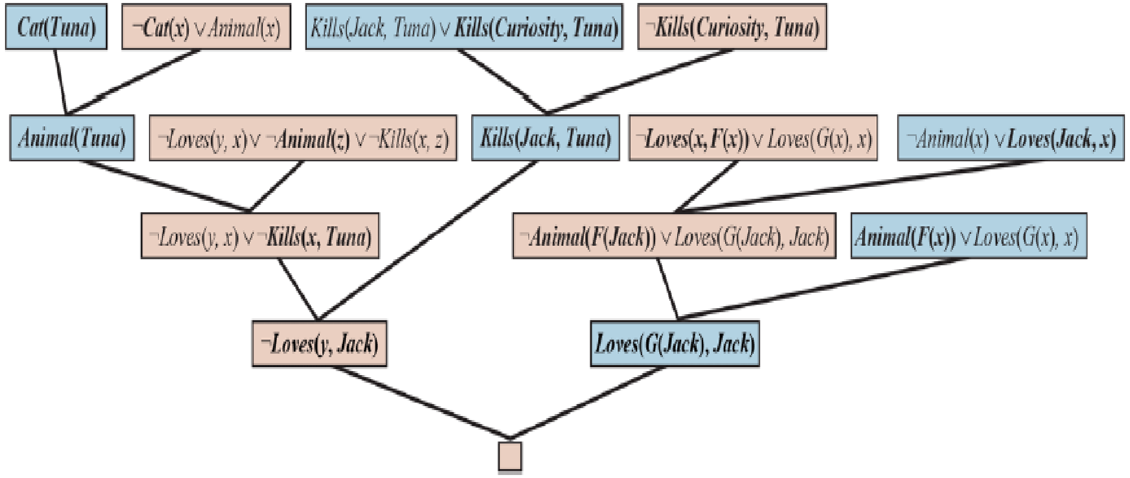
Everyone who loves all animals is loved by someone.

Anyone who kills an animal is loved by no one.

Jack loves all animals.

Either Jack or Curiosity killed the cat, who is named Tuna.

Did Curiosity kill the cat?



A. $\forall x\ [\forall y\ Animal(y) \Rightarrow Loves(x,y)] \Rightarrow [\exists y\ Loves(y,x)]$

B. $\forall x\ [\exists z\ Animal(z) \wedge Kills(x,z)] \Rightarrow [\forall y\ \neg Loves(y,x)]$

C. $\forall x\ Animal(x) \Rightarrow Loves(Jack,x)$

D. $Kills(Jack,Tuna) \vee Kills(Curiosity,Tuna)$

E. $Cat(Tuna)$

F. $\forall x\ Cat(x) \Rightarrow Animal(x)$

¬G. $\neg Kills(Curiosity,Tuna)$

A1. $Animal(F(x)) \vee Loves(G(x),x)$

A2. $\neg Loves(x,F(x)) \vee Loves(G(x),x)$

B. $\neg Loves(y,x) \vee \neg Animal(z) \vee \neg Kills(x,z)$

C. $\neg Animal(x) \vee Loves(Jack,x)$

D. $Kills(Jack,Tuna) \vee Kills(Curiosity,Tuna)$

E. $Cat(Tuna)$

F. $\neg Cat(x) \vee Animal(x)$

¬G. $\neg Kills(Curiosity,Tuna)$

Resolution strategies

- The main question in each step of the resolution is to select:
  - appropriate two clauses for being resolved?

- There are different techniques in this respect:
  - Unit preference, Unit Resolution, Set-of-Support (SoS), Input Resolution, Linear Resolution, etc.
  - Some of these strategies are complete (e.g. Unit Preference and Linear resolution)
  - Some are complete only if certain conditions hold (e.g., SoS)
  - Some others are complete only for the *Horn* clauses subsets (e.g. Unit Resolution and Input Resolution)
  - Horn clauses are slightly more general than definite clauses
  - There must be at most one positive literal in each clause
  - In Definite clauses there must be exactly one positive in each clause

Natural Deduction vs. Resolution

- CNF is simpler to work with;
- but, we loose readability and some useful control information.

Consider sentence "All judges who are not crooked, are educated."

1. $\forall x(\ judge(x) \wedge \neg\ crooked(x) \Rightarrow educated(x)\ )$ (standard form)
2. $\neg\ judge(x) \vee crooked(x) \vee educated(x)$ (Clausal form)

- In 1, we know that it only should be used to infer that someone is educated.
- But, with 2, there is no difference between the three predicates and it can be used to infer that someone is not a judge!