CS440/ECE448: Intro to Artificial Intelligence

Lecture 9: More on predicate logic

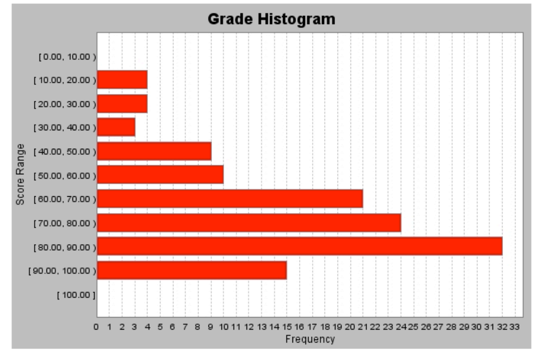
Prof. Julia Hockenmaier
juliahmr@illinois.edu

http://cs.illinois.edu/fa11/cs440

Quick upgrade on quizzes

Count:	122
Average:	69.6
Median:	76.0
Maximum:	97.8
Minimum:	10.7
Standard Deviation:	20.37

Hide Histogram



Review: syntax of predicate logic

The building blocks

A (finite) set of **variables** VAR:

$VAR = \{x, y, z, \ldots\}$

A (finite) set of **constants** $CONST$:

$CONST = \{john, mary, tom, \ldots\}$

For *n=1…N* :

A (finite) set of *n*-place **function** symbols $FUNC$

$FUNC_1 = \{fatherOf, \ successor, \ldots\}$

A (finite) set of *n*-place **predicate** symbols $PRED_n$:

$PRED_1 = \{student, \ blue, \ldots\}$

$PRED_2 = \{friend, \ sisterOf, \ldots\}$

Putting everything together

Terms: constants (*john*); variables (*x*); n-ary function symbols applied to *n* terms (*fatherOf(x)*)

– *Ground* terms contain no variables

Formulas: *n*-ary predicate symbols applied to *n* terms (*likes(x,y)*); negated formulas ($\neg$*fatherOf(x)*); conjunctions, disjunctions or implications of two formulas; quantified formulas

– *Ground* formulas (= sentences; propositions) contain no free variables
– *Open* formulas contain at least one free variable

Semantics of predicate logic

Model $M=(D,I)$

The **domain** D is a nonempty set of objects:
$D = \{a1, b4, c8, \ldots\}$

The **interpretation function** I maps:

- each constant c to an element c^I of D: $John^I = a1$
- each n-place function symbol f to an (total) n-ary function $f^I\ D^n \rightarrow D$: $fatherOf^I(a1) = b4$
- each n-place predicate symbol p to an n-ary relation $p^I \subseteq D^n$: $child^I = \{a1, c8\}$ $likes^I = \{\langle a1, b4\rangle, \langle b4, a1\rangle\}$

Interpretation of variables

A variable assignment v over a domain D is a (partial) function from variables to D.

The assignment $v = [a21/x, b13/y]$ assigns object $a21$ to the variable x, and object $b13$ to variable y.

We recursively manipulate variable assignments when interpreting quantified formulas.
Notation: $v[b/z]$ is just like v, but it also maps z to b.
We will make sure that v is undefined for z.

Interpretation of terms

Variables: $[\![x]\!]^{M,g} = g(x)$
defined by the variable assignment

Constants: $[\![c]\!]^{M,g} = c^I$
defined by the interpretation function

Functions: defined by the interpretation function and recursion on the arguments
$[\![f(t_1, \ldots. t_n)]\!]^{M,g} = f^{I}([\![t_1]\!]^{M,g}, \ldots, [\![t_n]\!]^{M,g})$

Interpretation of formulas

Atomic formulas:
$[\![P(t_1,\ldots t_n)]\!]^{M,g} = true$ iff $\langle [\![t_1]\!]^{M,g}, \ldots [\![t_n]\!]^{M,g} \rangle \in P^I$

Complex formulas (connectives):

$[\![\neg \varphi]\!]^{M,g} = true$ iff $[\![\varphi]\!]^{M,g} = false$
$[\![\varphi \wedge \psi]\!]^{M,g} = true$ iff $[\![\varphi]\!]^{M,g} = true$ and $[\![\psi]\!]^{M,g} = true$
$[\![\varphi \vee \psi]\!]^{M,g} = true$ iff $[\![\varphi]\!]^{M,g} = true$ or $[\![\psi]\!]^{M,g} = true$
$[\![\varphi \rightarrow \psi]\!]^{M,g} = true$ iff $[\![\varphi]\!]^{M,g} = false$ or $[\![\psi]\!]^{M,g} = true$

Interpretation of formulas: quantifiers

Universal quantifier:
$[\![\forall x\ \varphi]\!]^{M,g} = true$ iff $[\![\varphi]\!]^{M,g[u/x]} = true$
for **all** $u \in D$

Existential quantifier:
$[\![\forall x\ \varphi]\!]^{M,g} = true$ iff $[\![\varphi]\!]^{M,g[u/x]} = true$
for at least **one** $u \in D$

Satisfaction and entailment

φ is satisfied in M ($M \vDash \varphi$) *iff* $[\![\varphi]\!]^{M,[]} = true$

φ is valid ($\vDash \varphi$) *iff* φ is satisfied in any model.

φ entails ψ *iff* $\varphi \rightarrow \psi$ is valid.

NB: We cannot interpret *open* formulas
(i.e. containing free variables)

Using predicate logic

From English to predicate logic

Birds fly.

Some birds fly.

There are three people in SC1404.

SC1404 is empty.

There are exactly three people in SC1404.

"Birds fly" vs *"Some birds fly"*

Birds fly:

$\forall x\ [\mathrm{Bird}(x) \Rightarrow \mathrm{Flies}(x)]$

$\equiv \neg\exists x\ [\mathrm{Bird}(x) \wedge \neg\mathrm{Flies}(x)]$

Some birds fly:

$\exists x\ [\mathrm{Bird}(x) \wedge \mathrm{Flies}(x)]$

There are three people in SC1404 SC1404 is empty.

There are three people in 1404:

$$\begin{aligned}\exists x \exists y \exists z\ [&\mathrm{person}(x) \wedge \mathrm{in}(x, \mathrm{SC1404}) \\ &\wedge \mathrm{person}(y) \wedge \mathrm{in}(x, \mathrm{SC1404}) \\ &\wedge \mathrm{person}(z) \wedge \mathrm{in}(x, \mathrm{SC1404}) \\ &\wedge x \neq y \wedge x \neq z \wedge y \neq z]\end{aligned}$$

SC1404 is empty.

$\neg\exists x\ [\mathrm{person}(x) \wedge \mathrm{in}(x, \mathrm{SC1404})]$

There are *exactly* three people in SC1404

$\exists x \exists y \exists z\ [person(x) \wedge in(x, SC1404)$
$\wedge\ person(y) \wedge in(x, SC1404)$
$\wedge\ person(z) \wedge in(x, SC1404)$
$\wedge\ x \neq y \wedge x \neq z \wedge y \neq z$
$\wedge\ \forall w\ [(person(w) \wedge in(w, SC1404))$
$\longrightarrow (w = x \vee w = y \vee w = z)]]$

What about….

"Most birds fly."

This cannot be expressed in FOL:
$|\ bird^I \cap fly^I\ | > 0.5\ |\ bird^I\ |$

Inference in predicate logic

Inference in predicate logic

All men are mortal.
Socrates is a man.
Socrates is mortal.

We need a new version of modus ponens:

$$\frac{\forall x\ P(x) \longrightarrow Q(x) \qquad P(s')}{Q(s')}$$

FOL is semi-decidable

Can we prove whether A ⊨ B?

Case 1: A does entail B.
Any sound & complete inference procedure will terminate (and confirm that A ⊨ B)

Case 2: A does not entail B.
No inference procedure is guaranteed to terminate.

How do we deal with quantifiers and variables?

Solution 1: **Propositionalization**
Ground all the variables.

Solution 2: **Lifted inference**
Ground (skolemize) all the existentially quantified variables. All remaining variables are universally quantified.
Use unification.

Inference in predicate logic: propositionalization

Grounding variables

$\theta = \{x_1/c_1, \ldots x_n/c_n\}$ is a list of substitutions: variable x_i is replaced by ground term c_i.

The function SUBST(θ, ψ) applies the substitutions in θ to the free variables in formula ψ and returns the result.

SUBST({x/a, y/b}, friend(x,y)) = friend(a,b)

Substitutions

A *substitution* θ is a set of pairings of variables v_i with terms t_i:

$$\theta = \{v_1/t_1, v_2/t_2, v_3/t_3, \ldots, v_n/ t_n\}$$

- Each variable v_i is distinct
- t_i can be any term (variable, constant, function), as long as it does not contain v_i directly or indirectly

NB: the order of variables in θ doesn't matter
$\{x/y, y/f(a)\} = \{y/f(a), x/y\} = \{x/f(a), y/f(a)\}$

Are these acceptable substitutions?

$\{x/y\}$	Yes
$\{x/y, z/F(y)\}$	Yes
$\{x/y, z/F(y), x/A\}$	NO: x appears twice
$\{x/y, z/F(y), y/A\}$	Yes
$\{x/y, y/F(z), z/G(x)\}$	NO: z/F(z) not allowed

Universal instantiation

We can replace a universally quantified variable with *any* ground term:

$$\frac{\forall x\ \psi(x)}{\textsc{Subst}(\{x/a\}, \psi)}\ (\text{UI})$$

Hence: $$\frac{\forall x\ (man(x) \longrightarrow mortal(x))}{\textsc{Subst}(\{x/s'\}, man(x) \longrightarrow mortal(x))}\ (\text{UI})$$
$$= man(s') \longrightarrow mortal(s')$$

Existential instantiation

We can replace a existentially quantified variable with a *new* ground term:

$$\frac{\exists x\ \psi(x)}{\textsc{Subst}(\{x/t\}, \psi)}\ (\text{EI})$$

Condition:
t doesn't appear in ψ or in any other formula in our KB

Hence: $$\frac{\exists x\ man(x)}{man(c')}\ (\text{EI})$$

Propositionalization

If we have a finite domain, we can use UI and EI to eliminate all variables.

KB: $\forall x\ [(student(x) \wedge inClass(x)) \longrightarrow sleep(x)]$
$student(Mary)$ $professor(Julia)$

Propositionalized KB:
$(student(Mary) \wedge inClass(Mary)) \longrightarrow sleep(Mary)$
$(student(Julia) \wedge inClass(Julia)) \longrightarrow sleep(Julia)$
etc.

…this is not very efficient....

Propositionalization

If we can reduce A $\models$ B to propositional resolution, A $\models$ B is decidable.

We cannot reduce A $\models$ B to propositional resolution if they contain functions or equality.

Direct (lifted) inference in predicate logic: will it work?

Some fragments of FOL are decidable

Can we prove whether A $\models$ B?

If both A and B belong to the same decidable fragment of FOL,
any sound and complete inference procedure will terminate
(and tell us whether A $\models$ B)

Which fragment of FOL does φ belong to?

Every WFF in FOL can be translated to prenex conjunctive normal form.

Prenex form:
All quantifiers are at the front of the formula

Conjunctive Normal Form:
A conjunction of clauses.

Depending on the form of the clauses in the CNF of A, B, $A \vDash B$ may be decidable.

Prenex normal form

Every well-formed formula in FOL has an equivalent **prenex normal form,** in which all the quantifiers are at the front.

$\forall x \exists y \exists z \forall w\ \varphi(x,y,z,w)$

$\varphi(x,y,z,w)$ (the 'matrix') does not contain any quantifiers.

We can move quantifiers outside of connectives

We already saw:

$\forall x P(x) \wedge \forall y Q(y) \equiv \forall x \forall y\ [P(x) \wedge Q(y)]$
$\forall x P(x) \vee \forall y Q(y) \equiv \forall x \forall y\ [P(x) \vee Q(y)]$

$\exists x P(x) \wedge \exists y Q(y) \equiv \exists x \exists y\ [P(x) \wedge Q(y)]$
$\exists x P(x) \vee \exists y Q(y) \equiv \exists x \exists y\ [P(x) \vee Q(y)]$

Equivalences: alphabetic variants

We can move quantifiers to the front, e.g.:
$\exists x P(x) \wedge \exists y Q(y) \equiv \exists x \exists y\ [P(x) \wedge Q(y)]$
but we cannot 'merge' them:
$\exists x P(x) \wedge \exists x Q(x) \not\equiv \exists x\ [P(x) \wedge Q(y)]$

To avoid clashes, we first rename bound variables to any other new (unbound) variable:
$\exists x P(x) \wedge \exists x Q(x) \equiv \exists x P(x) \wedge \exists y Q(y)$

$\exists y Q(y)$ is an alphabetic variant of $\exists x Q(x)$

We can move quantifiers outside of connectives

If x is not free in φ:

$\varphi \wedge \forall x\, \psi(x) \equiv \forall x[\varphi \wedge \psi(x)]$
$\varphi \wedge \exists x\, \psi(x) \equiv \exists x[\varphi \wedge \psi(x)]$

$\varphi \vee \forall x\, \psi(x) \equiv \forall x[\varphi \vee \psi(x)]$
$\varphi \vee \exists x\, \psi(x) \equiv \exists x[\varphi \vee \psi(x)]$

$\varphi \rightarrow \forall x\, \psi(x) \equiv \forall x[\varphi \rightarrow \psi(x)]$
$\varphi \rightarrow \exists x\, \psi(x) \equiv \exists x[\varphi \rightarrow \psi(x)]$

We can move quantifiers outside of negation

Not all x are ψ = (At least) one x is not ψ

$$\neg [\forall x\, \psi(x)] \equiv \exists x[\neg \psi(x)]$$

No x is ψ = All x are not ψ

$$\neg [\exists x\, \psi(x)] \equiv \forall x\, [\neg \psi(x)]$$

Gold medal winners have not played against anybody who beat all other players.

$\forall x[g(x) \rightarrow \neg \exists y[p(y,x) \wedge \forall z\, b(y,z)]]$

$\equiv \forall x[g(x) \rightarrow \forall y\, \neg\, [p(y,x) \wedge \forall z\, b(y,z)]]$
$\equiv \forall x[g(x) \rightarrow \forall y[\neg\, p(y,x) \vee \neg \forall z\, b(y,z)]]$
$\equiv \forall x[g(x) \rightarrow \forall y[\neg\, p(y,x) \vee \exists z\, \neg\, b(y,z)]]$
$\equiv \forall x \forall y\, [g(x) \rightarrow [\neg\, p(y,x) \vee \exists z\, \neg\, b(y,z)]]$
$\equiv \forall x \forall y\, [g(x) \rightarrow \exists z\, [\neg\, p(y,x) \vee \neg\, b(y,z)]]$

$\equiv \forall x \forall y \exists z\, [g(x) \rightarrow [\neg\, p(y,x) \vee \neg\, b(y,z)]]$
$\equiv \forall x \forall y \exists z\, [g(x) \rightarrow [p(y,x) \rightarrow \neg\, b(y,z)]]$

= Everybody that a gold medal winner played against has not beaten (=has drawn or lost against) somebody.

Normal forms for FOL

Conjunctive normal form: a conjunction of clauses.

$(\varphi_1 \vee \ldots \vee \varphi_n) \wedge \ldots \wedge (\varphi'_1 \vee \ldots \vee \varphi'_m)$

Clause: a disjunction of an arbitrary number of positive or negative literals.

$(\varphi_1 \vee \neg\varphi_2 \vee \varphi_3 \vee \neg\varphi_4)$

Literal: a (negated) predicate and its arguments

likes(x, John')
¬likes(John', motherOf(Mary))
¬student(x)

Fragments of FOL

Definite clause: exactly one positive literal
Facts: ψ
Implications: $\neg\varphi_1 \vee \ldots \vee \neg\varphi_n \vee \psi \equiv [(\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow \psi]$

Horn clause: at most one positive literal.
All definite clauses, plus (resolution) queries: $\neg\psi$

Clauses: arbitrary number of positive literals
All Horn clauses, plus any disjunction, e.g $\varphi \vee \psi$

Inference algorithms for FOL

Forward chaining:
Complete for definite clauses.

Resolution:
Refutation-complete for CNF.
(Resolution will derive a contradiction if a set of sentences is not satsifiable).

The resolution of two Horn clauses yields another Horn clause.

Prerequisites for lifted inference: Skolemization and Unification

Prerequisites for lifted inference

We cannot interpret open formulas, but we need to deal with quantified variables.

Existentially quantified variables:
replace by ground terms (Skolemization)

Universally quantified variables:
match with other universally quantified variables or ground terms (unification)

Existentially quantified variables

If $\forall x\ \exists y\ R(x,y)$ is valid in our model, then:

For *any* way of instantiating x there is a way to instantiate y such that "R" holds between them.

Since y can depend on x, we replace all occurrences of y by a new function of x, $F(x)$. We assume $F(x)$ evaluates to the value which makes $R(x,F(x))$ true.
$\forall x\ \exists y\ R(x,y)$ is equivalent to $\forall x\ R(x,F(x))$

Skolemization: remove existentially quantified variables

Replace any existentially quantified variable $\exists x$ that is in the scope of universally quantified variables $\forall y_1 \ldots \forall y_n$ with a new function $F(y_1,\ldots,y_n)$ (a **Skolem function**)

Replace any existentially quantified variable $\exists x$ that is not in the scope of any universally quantified variables with a new constant c (a **Skolem term**)

The effect of Skolemization

$\forall x\ \forall y\ \exists w\ \forall z\ Q(x, y, w, z, G(w, x))$

is equivalent to

$\forall x\ \forall y\ \forall z\ Q(x, y, P(x, y), z, G(P(x, y), x))$

where P is the Skolem function for w.

NB: the Skolem function is a function, so this is not decidable anymore.

Modus ponens

With propositionalization:

$$\cfrac{\cfrac{\forall x\ human(x) \longrightarrow mortal(x)}{human(s') \longrightarrow mortal(s')}\ (UI) \qquad human(s')}{mortal(s')}\ (MP)$$

How can we match $human(s')$ and $\forall x\ human(x) \longrightarrow mortal(x)$ directly?

Substitutions

A *substitution* θ is a set of pairings of variables v_i with terms t_i:

$\theta = \{v_1/t_1, v_2/t_2, v_3/t_3, \ldots, v_n/ t_n\}$

- Each variable v_i is distinct
- t_i can be any term (variable, constant, function), as long as it does not contain v_i directly or indirectly

NB: the order of variables in θ doesn't matter
$\{x/y, y/f(a)\} = \{y/f(a), x/y\} = \{x/f(a), y/f(a)\}$

Unification

Two sentences φ and ψ **unify** to σ
($\text{UNIFY}(\varphi, \psi) = \sigma$)
if σ is a substitution such that
$\text{SUBST}(\sigma,\varphi) = \text{SUBST}(\sigma,\psi)$.

Example:
$\text{UNIFY}(like(x, M'), like(C',y)) =\{x/C', y/M'\}$

Unification

A set of sentences $\varphi_{1,}\ldots\varphi_n$ **unify** to σ
if for all $i\neq j$: $\text{SUBST}(\sigma,\varphi_i) = \text{SUBST}(\sigma, \varphi_j)$.

σ is the unifier of $\varphi_{1,}\ldots\varphi_n$
$\text{SUBST}(\sigma,\varphi_i)$ is a unification instance.

Standardizing apart

Unification is not well-behaved if φ and ψ contain the same variable:

$\text{UNIFY}(like(x, M'), like(C',x))$: fail.

We need to *standardize* φ and ψ *apart* (rename this variable in one term):

$\text{UNIFY}(like(x, M'), like(C',y)) = \{x/C', y/M'\}$
to yield $like(C',M')$

Do these unify?

(Single lower case letters are variables)

UNIFY(P(x,y,z), P(w, w, Fred))

σ = {x=Fred, y=Fred, z=Fred, w=Fred}

Equivalently: σ' = {x=Fred, w=y, z=Fred , y=x}

Both yield P(Fred,Fred,Fred)

Are there others?

UNIFY(P(x,y,z), P(w, w, Fred))

σ = {x=Mary, y=Mary, z=Fred, w=Mary}

Equivalently: σ' = {x=Mary, w=y, z=Fred , y=x}

Both yield P(Mary, Mary, Fred)

Most General Unifier (MGU)

σ is the most general unifier (MGU) of φ and ψ if it imposes the fewest constraints.

The MGU of φ and ψ is unique.
(modulo alphabetic variants, i.e. different variable names)

Applying the MGU to an expression yields a *most general unification instance*.

We often define UNIFY(φ, ψ) to return MGU(φ, ψ)

What is the MGU?

MGU(P(x,y,z), P(w,w,Fred))

σ ={x=w, y=w, z=Fred} yields P(w,w,Fred)

Equivalently, σ ={x=u, y=u, w=u, z=Fred} yields the alphabetic variant P(u,u,Fred)

What is the MGU?

MGU(m(Ann, x, Bob), m(Ann, x, Bob)):
m(Ann, x, Bob)
MGU(m(Ann, x, Bob), m(y, x, Chuck)):
fail.
MGU(m(Ann, x, Bob), m(y, x, Father-of(Chuck)):
fail.
MGU(p(w, w, Fred) , p(x, y, y)):
p(Fred, Fred, Fred)
MGU(q(r, r), q(x, F(x))):
fail
MGU(r(g(x,Bob),y, y), r(z,g(Fred,w),z)):
r(g(x,Bob), g(Fred,w), g(Fred(w)))

To conclude…

Today's key concepts

Semantics of first-order logic:
Models for FOL

Inference with first-order logic:
Propositionalization
(Universal elimination and existential elimination)

Dealing with variables:
prenex normal form, unification and skolemization

To do: read Ch.9.1-3