Logical Languages
part 1

2020

Instructor: Odelia Schwartz

Programming paradigms

- Imperative
- Functional
- Logical

Logical programs: declarative rather than procedural
Only desired results (and collections of facts and rules) specified, rather than detailed procedure for producing Results

Syntax and semantics very different from imperative

Towards logical languages: applications

- Relational Database Management Systems
  e.g., Structured Query Database (SQL) is non procedural (tables of information; relations between tables)

- Expert systems
  Designed to emulate user expertise; lots of facts and relations in databases. Use inference rules to infer new facts. Example: with Prolog

Towards logical languages: applications

Fairly recent example: IBM Watson won jeopardy challenge

https://www.cs.miami.edu/home/odelia/teaching/csc419_spring20/syllabus/IBM_Watson_Prolog.pdf

Natural Language Processing With Prolog in the IBM *Watson* System

Adam Lally
IBM Thomas J. Watson Research Center
Paul Fodor Stony Brook University
24 May 2011

https://www.youtube.com/watch?v=P18EdAKuC1U

Formal logic and intro to predicate calculus

- Before we look at Prolog
- We will talk about formal logic…
  this class

Formal logic and intro to predicate calculus

- Proposition?

Formal logic and intro to predicate calculus

- Proposition: Logical statement that may or may not be true. Consists of objects and relationships amongst objects

Formal logic and intro to predicate calculus

- Proposition: Logical statement that may or may not be true. Consists of objects and relationships amongst objects

  We check validity of propositions through formal logic

Formal logic and intro to predicate calculus

- Symbolic logic used to:

➢ Express propositions

Formal logic and intro to predicate calculus

- Symbolic logic used to:

  - Express propositions
  - Relationships between propositions

Formal logic and intro to predicate calculus

- Symbolic logic used to:

- Express propositions
- Relationships between propositions
- How to infer new propositions from others assumed true

Formal logic and intro to predicate calculus

- Close relation between formal logic and mathematics

- Axioms of number and set theory are initial propositions, assumed true
- Theorems and additional propositions can be inferred from initial set

Formal logic and intro to predicate calculus

- Propositions

Examples:

class(csc419)
year(2020)
location(zoom)

Formal logic and intro to predicate calculus

- Propositions

Examples:

class(csc419)
year(2020)
location(zoom)

like(odelia, python)
like(michael, elixir)
like(David,haskell)

Formal logic and intro to predicate calculus

- Propositions can be stated in two forms:

➢ Fact: proposition assumed to be true

Formal logic and intro to predicate calculus

- Propositions can be stated in two forms:

- Fact: proposition assumed to be true
- Query: truth of proposition to be determined

Formal logic and intro to predicate calculus

- Compound proposition

- Two or more atomic propositions
- Propositions connected by operators

Formal logic and intro to predicate calculus

- Compound proposition

- Two or more atomic propositions
- Propositions connected by operators
- What logical operators?

Formal logic and intro to predicate calculus

- Logical operators

Name	*Symbol*	*Example*	*Meaning*
negation	$\neg$	$\neg$ a	not *a*

Formal logic and intro to predicate calculus

- Logical operators

Name	*Symbol*	*Example*	*Meaning*
negation	$\neg$	$\neg$ a	not *a*
conjunction	$\cap$	$a \cap b$	*a* and *b*
disjunction	$\cup$	$a \cup b$	*a* or *b*

Formal logic and intro to predicate calculus

- Logical operators

Name	*Symbol*	*Example*	*Meaning*
negation	$\neg$	$\neg$ a	not *a*
conjunction	$\cap$	$a \cap b$	*a* and *b*
disjunction	$\cup$	$a \cup b$	*a* or *b*
equivalence	$\equiv$	$a \equiv b$	*a* is equivalent to *b*

Identical truth table

Formal logic and intro to predicate calculus

- Logical operators

Name	*Symbol*	*Example*	*Meaning*
negation	$\neg$	$\neg$ a	not *a*
conjunction	$\cap$	$a \cap b$	*a* and *b*
disjunction	$\cup$	$a \cup b$	*a* or *b*
equivalence	$\equiv$	$a \equiv b$	*a* is equivalent to *b*
implication	$\supset$	$a \supset b$	*a* implies *b*
	$\subset$	$a \subset b$	*b* implies *a*

What does implication mean?

Formal logic and intro to predicate calculus

- Implication

implication	$\supset$	$a \supset b$	a implies b
	$\subset$	$a \subset b$	b implies a

a implies b:

- if a is true then b is true
- if a is false, that can imply anything

Example?

Formal logic and intro to predicate calculus

- Implication

implication	$\supset$	$a \supset b$	a implies b
	$\subset$	$a \subset b$	b implies a

a implies b:

- if a is true then b is true
- if a is false, that can imply anything

Example?

- if you know your functional languages, you will easily land an internship
- if you don't know your functional languages, you may either easily land an internship or you may not

Formal logic and intro to predicate calculus

- Implication

implication	$\supset$	$a \supset b$	*a* implies *b*
	$\subset$	$a \subset b$	*b* implies *a*

a implies b:

- if a is true then b is true
- if a is false, that can imply anything

Example?
also, collie implies dog…
dog implies mammal…

Formal logic and intro to predicate calculus

- Precedence order

$$\neg$$

$$\cap \quad \cup \quad \equiv$$

$$\supset \quad \subset$$

Formal logic and intro to predicate calculus

- Compound statements

$$a \cap \neg b \supset d$$

Formal logic and intro to predicate calculus

- Compound statements

$$a \cap \neg b \supset d$$

Based on precedence:

$$(a \cap (\neg b)) \supset d$$

Formal logic and intro to predicate calculus

- For all, there exists…

Name	*Example*	*Meaning*
universal	$\forall X.P$	For all X, P is true.
existential	$\exists X.P$	There exists a value of X such that P is true.

Formal logic and intro to predicate calculus

- For all, there exists…

Name	*Example*	*Meaning*
universal	$\forall X.P$	For all X, P is true.
existential	$\exists X.P$	There exists a value of X such that P is true.

Variables appear in propositions only as quantifiers

Formal logic and intro to predicate calculus

- Examples:

$\forall X.$ (dog(x) $\supset$ mammal(x))

Formal logic and intro to predicate calculus

- Examples:

$\forall X.$ (dog(x) $\supset$ mammal(x))

For any value of x, if x is a dog then x is a mammal

Formal logic and intro to predicate calculus

- Examples:

$$\exists X.\ (\text{mother}(\text{mary}, X) \cap \text{male}(X))$$

Formal logic and intro to predicate calculus

- Examples:

$$\exists X.\ (\text{mother}\,(\text{mary}, X)\ \cap\ \text{male}\,(X)\ )$$

There exists a value of X such that mary is the mother of X and X is a male

Formal logic and intro to predicate calculus

- Examples:

$$\exists X.\ (\mathrm{mother}\,(\mathrm{mary}, X)\ \cap\ \mathrm{male}\,(X)\ )$$

There exists a value of X such that mary is the mother of X and X is a male

In other words, mary has a son…

Formal logic and intro to predicate calculus

- Clausal form

➢ There are too many ways to state the same thing (e.g., propositions with the same meaning)

Formal logic and intro to predicate calculus

- Clausal form

- There are too many ways to state the same thing (e.g., propositions with the same meaning)

- Not so much a problem for humans, but for computers a serious problem

Formal logic and intro to predicate calculus

- Clausal form

- There are too many ways to state the same thing (e.g., propositions with the same meaning)

- Not so much a problem for humans, but for computers a serious problem

- A standard form is desirable

Formal logic and intro to predicate calculus

- Clausal form

- There are too many ways to state the same thing (e.g., propositions with the same meaning)

- Not so much a problem for humans, but for computers a serious problem

- A standard form is desirable

- Clausal form is a relatively simple form of propositions and is one such standard form

Formal logic and intro to predicate calculus

- All propositions can be expressed in **clausal form**

$$B_1 \cup B_2 \cup \ldots \cup B_n \subset A_1 \cap A_2 \cap \ldots \cap A_m$$

Meaning??

Formal logic and intro to predicate calculus

- All propositions can be expressed in **clausal form**

$$B_1 \cup B_2 \cup \ldots \cup B_n \subset A_1 \cap A_2 \cap \ldots \cap A_m$$

Meaning??

Right side implies left side

Formal logic and intro to predicate calculus

- All propositions can be expressed in **clausal form**

$$B_1 \cup B_2 \cup \ . \ . \ . \ \cup B_n \subset A_1 \cap A_2 \cap \ . \ . \ . \ \cap A_m$$

Meaning??

Right side implies left side
If all of the A are true, at least one B is true

Formal logic and intro to predicate calculus

- All propositions can be expressed in **clausal form**

$$B_1 \cup B_2 \cup \ldots \cup B_n \subset A_1 \cap A_2 \cap \ldots \cap A_m$$

All predicate calculus propositions can be converted into clausal form (proof: Nilsson 1971)

Formal logic and intro to predicate calculus

- All propositions can be expressed in **clausal form**

$$B_1 \cup B_2 \cup \ldots \cup B_n \subset A_1 \cap A_2 \cap \ldots \cap A_m$$

Examples??

Formal logic and intro to predicate calculus

- All propositions can be expressed in **clausal form**

$$B_1 \cup B_2 \cup \ldots \cup B_n \subset A_1 \cap A_2 \cap \ldots \cap A_m$$

Examples??

$$\text{likes(bob, trout)} \subset \text{likes(bob, fish)} \cap \text{fish(trout)}$$

Formal logic and intro to predicate calculus

- All propositions can be expressed in **clausal form**

$$B_1 \cup B_2 \cup \ldots \cup B_n \subset A_1 \cap A_2 \cap \ldots \cap A_m$$

Examples??

$$\text{likes}(\text{bob}, \text{trout}) \subset \text{likes}(\text{bob}, \text{fish}) \cap \text{fish}(\text{trout})$$

if Bob likes fish and trout is a fish,
then bob likes trout

Formal logic and intro to predicate calculus

- All propositions can be expressed in **clausal form**

Examples??

$$\text{father}(\text{louis}, \text{al}) \cup \text{father}(\text{louis}, \text{violet}) \subset \text{father}(\text{al}, \text{bob}) \cap \text{mother}(\text{violet}, \text{bob}) \cap \text{grandfather}(\text{louis}, \text{bob})$$

Meaning??

Formal logic and intro to predicate calculus

- All propositions can be expressed in **clausal form**

Examples??

$$\text{father}(\text{louis}, \text{al}) \cup \text{father}(\text{louis}, \text{violet}) \subset \text{father}(\text{al}, \text{bob}) \cap \text{mother}(\text{violet}, \text{bob}) \cap \text{grandfather}(\text{louis}, \text{bob})$$

Meaning??

If al is bob's father, and violet is bob's mother, and louis is bob's grandfather

Then this implies that either louis is violet's father or louis is al's father

Predicate calculus and proving theorems

- Predicate calculus: method for expressing collections of propositions
- One use: determine whether any interesting/useful facts can be inferred…

Predicate calculus and proving theorems

- Predicate calculus: method for expressing collections of propositions
- One use: determine whether any interesting/useful facts can be inferred…

(analogous to mathematics; discover new theorems that can be inferred from known axioms and theorems)

Predicate calculus and proving theorems

- 1950s, 1060s: lots of interest in automating theorem-proving process
- Significant breakthrough: Alan Robinson 1965

Predicate calculus and proving theorems

- 1950s, 1060s: lots of interest in automating theorem-proving process
- Significant breakthrough: Alan Robinson 1965

Resolution: inference rule that allows inferred propositions to be computed from given propositions

Predicate calculus and proving theorems

- 1950s, 1060s: lots of interest in automating theorem-proving process
- Significant breakthrough: Alan Robinson 1965

Resolution: inference rule that allows inferred propositions to be computed from given propositions, therefore providing method with potential for theorem proving

Predicate calculus and proving theorems

Example of resolution process:

Suppose two propositions with the form:

$P_1 \subset P_2$ (P_2 implies P_1)

$Q_1 \subset Q_2$ (Q_2 implies Q_1)

Predicate calculus and proving theorems

Example of resolution process:

Suppose two propositions with the form:

$P_1 \subset P_2$ (P_2 implies P_1)

$Q_1 \subset Q_2$ (Q_2 implies Q_1)

Suppose further:

P_1 is identical to Q_2

Predicate calculus and proving theorems

Example of resolution process:

Suppose further:

P_1 is identical to Q_2

So we can rename P_1 and Q_2 to T

So now:

$$T \subset P_2$$
$$Q_1 \subset T$$

Predicate calculus and proving theorems

Example of resolution process:

So now:

$$T \subset P_2$$
$$Q_1 \subset T$$

So we can infer that??

Predicate calculus and proving theorems

Example of resolution process:

So now:

$$T \subset P_2$$
$$Q_1 \subset T$$

We can therefore infer that:

$$Q_1 \subset P_2$$

Now, because P_2 implies T and T implies Q_1, it is logically obvious that P_2 implies Q_1, which we could write as

$$Q_1 \subset P_2$$

Predicate calculus and proving theorems

Another example of resolution process:

$older(joanne, jake) \subset mother(joanne, jake)$
$wiser(joanne, jake) \subset older(joanne, jake)$

Predicate calculus and proving theorems

Another example of resolution process:

older (joanne, jake) $\subset$ mother (joanne, jake)
wiser (joanne, jake) $\subset$ older (joanne, jake)

Then using resolution we can infer that?

Predicate calculus and proving theorems

Another example of resolution process:

older (joanne, jake) $\subset$ mother (joanne, jake)
wiser (joanne, jake) $\subset$ older (joanne, jake)

Then using resolution we can infer that:

wiser (joanne, jake) $\subset$ mother (joanne, jake)

Predicate calculus and proving theorems

Process of resolution:

1. Terms on left side of the two clausal propositions are OR’d together, to make left side of new proposition

Predicate calculus and proving theorems

Process of resolution:

1. Terms on left side of the two clausal propositions are OR'd together, to make left side of new proposition
2. Terms on right side of two clausal propositions are And'd together, to make right side of new proposition

Predicate calculus and proving theorems

Process of resolution:

1. Terms on left side of the two clausal propositions are OR'd together, to make left side of new proposition
2. Terms on right side of two clausal propositions are And'd together, to make right side of new proposition
3. Any term that appears on both sides of new proposition removed from both sides

Predicate calculus and proving theorems

Example process of resolution:

$$\text{father(bob, jake)} \cup \text{mother(bob, jake)} \subset \text{parent(bob, jake)}$$
$$\text{grandfather(bob, fred)} \subset \text{father(bob, jake)} \cap \text{father(jake, fred)}$$

First, what does this read as?

Next, what is the resolution?

Predicate calculus and proving theorems

Example process of resolution:

$\text{father}(\text{bob}, \text{jake}) \cup \text{mother}(\text{bob}, \text{jake}) \subset \text{parent}(\text{bob}, \text{jake})$
$\text{grandfather}(\text{bob}, \text{fred}) \subset \text{father}(\text{bob}, \text{jake}) \cap \text{father}(\text{jake}, \text{fred})$

1, 2: $\text{father}(\text{bob}, \text{jake}) \cup \text{mother}(\text{bob}, \text{jake}) \cup \text{grandfather}(\text{bob}, \text{fred})$

$\subset$

$\text{parent}(\text{bob}, \text{jake}) \cap$
$\text{father}(\text{bob}, \text{jake}) \cap \text{father}(\text{jake}, \text{fred})$

OR the left sides

AND the right sides

Predicate calculus and proving theorems

Example process of resolution:

father (bob, jake) $\cup$ mother (bob, jake) $\subset$ parent (bob, jake)
grandfather (bob, fred) $\subset$ father (bob, jake) $\cap$ father (jake, fred)

1, 2: father (bob, jake) $\cup$ mother (bob, jake) $\cup$ grandfather (bob, fred)

OR the left sides

$\subset$

parent (bob, jake) $\cap$
father (bob, jake) $\cap$ father (jake, fred)

AND the right sides

Now what??

Predicate calculus and proving theorems

Example process of resolution:

$father(bob, jake) \cup mother(bob, jake) \subset parent(bob, jake)$
$grandfather(bob, fred) \subset father(bob, jake) \cap father(jake, fred)$

1, 2: $father(bob, jake) \cup mother(bob, jake) \cup grandfather(bob, fred)$

OR the left sides

$\subset$

$parent(bob, jake) \cap father(bob, jake) \cap father(jake, fred)$

AND the right sides

3: remove any term that appears on both sides

Predicate calculus and proving theorems

Example process of resolution:

1, 2: father (bob, jake) ∪ mother (bob, jake) ∪ grandfather (bob, fred)

⊂

parent (bob, jake) ∩ father (bob, jake) ∩ father (jake, fred)

OR the left sides

AND the right sides

3: remove any term that appears on both sides

So we obtain by resolution:

mother (bob, jake) ∪ grandfather (bob, fred) ⊂ parent (bob, jake) ∩ father (jake, fred)

Predicate calculus and proving theorems

Example process of resolution:

We started with:

$father(bob, jake) \cup mother(bob, jake) \subset parent(bob, jake)$
$grandfather(bob, fred) \subset father(bob, jake) \cap father(jake, fred)$

We obtained from resolution:

$$mother(bob, jake) \cup grandfather(bob, fred) \subset parent(bob, jake) \cap father(jake, fred)$$

Predicate calculus and proving theorems

Example process of resolution:

We started with original propositions:

$$\text{father}(\text{bob}, \text{jake}) \cup \text{mother}(\text{bob}, \text{jake}) \subset \text{parent}(\text{bob}, \text{jake})$$
$$\text{grandfather}(\text{bob}, \text{fred}) \subset \text{father}(\text{bob}, \text{jake}) \cap \text{father}(\text{jake}, \text{fred})$$

We obtained from resolution:

$$\text{mother}(\text{bob}, \text{jake}) \cup \text{grandfather}(\text{bob}, \text{fred}) \subset \text{parent}(\text{bob}, \text{jake}) \cap \text{father}(\text{jake}, \text{fred})$$

In English…?

Predicate calculus and proving theorems

Example process of resolution:

We started with original propositions:

father (bob, jake) $\cup$ mother (bob, jake) $\subset$ parent (bob, jake)
grandfather (bob, fred) $\subset$ father (bob, jake) $\cap$ father (jake, fred)

If bob is the parent of jake implies that bob is either the father or mother of jake
And bob is the father of jake and jake is the father of fred implies that bob is the grandfather of fred

We obtained from resolution:

mother (bob, jake) $\cup$ grandfather (bob, fred) $\subset$
parent (bob, jake) $\cap$ father (jake, fred)

Then bob is the parent of jake and jake is the father of fred implies either bob is jake's mother or bob is fred's grandfather

Predicate calculus and proving theorems

Resolution is actually more complex than these simple examples…

We'll later discuss in terms of Prolog

Predicate calculus and proving theorems

Resolution is actually more complex than these simple examples…

We'll later discuss in terms of Prolog

- Main idea: Presence of variables in propositions requires resolution to find values for the variables that allows matching to succeed

Predicate calculus and proving theorems

- Main idea: Presence of variables in propositions requires resolution to find values for the variables that allows matching to succeed

- **Unification:** Finding values for variables in propositions that allows matching to succeed

Predicate calculus and proving theorems

- Main idea: Presence of variables in propositions requires resolution to find values for the variables that allows matching to succeed

- **Unification:** Finding values for variables in propositions that allows matching to succeed

- **Instantiation:** temporary assigning of values to variables to allow unification

Predicate calculus and proving theorems

- Main idea: Presence of variables in propositions requires resolution to find values for the variables that allows matching to succeed

- **Unification:** Finding values for variables in propositions that allows matching to succeed

- **Instantiation:** temporary assigning of values to variables to allow unification

- **Backtracking:** if resolution process to instantiate a variable with a value fails to complete required matching, then we backtrack and instantiate variable with different value

Predicate calculus and proving theorems

- Main idea: Presence of variables in propositions requires resolution to find values for the variables that allows matching to succeed

- **Unification:** Finding values for variables in propositions that allows matching to succeed

- **Instantiation:** temporary assigning of values to variables to allow unification

- **Backtracking:** if resolution process to instantiate a variable with a value fails to complete required matching, then we backtrack and instantiate variable with different value

We'll discuss these more in Prolog

Predicate calculus and proving theorems

- One way to simplify resolution process: restrict to simpler forms of propositions

Predicate calculus and proving theorems

- One way to simplify resolution process: restrict to simpler forms of propositions

Horn Clause

Predicate calculus and proving theorems

- One way to simplify resolution process: restrict to simpler forms of propositions

Horn Clause

Either
(1) single atomic proposition on left side

Predicate calculus and proving theorems

- One way to simplify resolution process:
  restrict to simpler forms of propositions

Horn Clause

Either
(1) single atomic proposition on left side
Or (2) empty left side

Predicate calculus and proving theorems

- One way to simplify resolution process:
  restrict to simpler forms of propositions

Horn Clause

Either
(1) single atomic proposition on left side
(2) empty left side

Also called
(1) Headed horn clause
(2) Headless Horn clause

Predicate calculus and proving theorems

- One way to simplify resolution process: restrict to simpler forms of propositions

Horn Clause example:

(1) Headed horn clause

$$\text{likes}(\text{bob}, \text{trout}) \subset \text{likes}(\text{bob}, \text{fish}) \cap \text{fish}(\text{trout})$$

Predicate calculus and proving theorems

- One way to simplify resolution process:
  restrict to simpler forms of propositions

Horn Clause example:

(1) Headed horn clause

$$\text{likes}(\text{bob}, \text{trout}) \subset \text{likes}(\text{bob}, \text{fish}) \cap \text{fish}(\text{trout})$$

(2) Headless Horn clause

$$\text{father}(\text{bob}, \text{jake})$$

Often used to state fact

Predicate calculus and proving theorems

- One way to simplify resolution process: restrict to simpler forms of propositions

Horn Clause example:

(1) Headed horn clause

$$likes(bob, trout) \subset likes(bob, fish) \cap fish(trout)$$

(2) Headless Horn clause

$$father(bob, jake)$$

Often used to state fact

Most, but not all, propositions can be stated as Horn clauses. The restriction to Horn clauses makes resolution a practical process for proving theorems.