Logic in Computer Sciences - Predicate Logic

Seon-Ho Park
`shpark@imtl.skku.ac.kr`

Sungkyunkwan University

IMTL Doctoral Seminar, 2010

Outline

Motivation and Introduction

Limitation

- In propositional logic, there are limited, several sentence components like *not*, and, *or* and *if* . . . then
- What can we do with "there exists ...", "all ...", "among ...", and "only ..." in propositional logic?
- Clear limitations of propositional logic : *the desire to express more subtle declarative sentences* led to the design of *predicate logic (first-order logic)*

Predicate Logic

- First-order logic, first-order predicate calculus, lower predicate calculus, quantification theory
- Covering predicates and quantification

Motivation and Introduction

Example : “Every ***student*** is ***younger than*** some ***instructor***”

- In Propositional Logic
  - The sentence is identified as a propositional atom p
  - It cannot reflect the finer logical structure of this sentence
- What is the statement about??
  - Being a student, being an instructor, being younger than somebody else
- Using *predicates* for expressing them together with their logical relationships and dependences
  - $S(\text{andy})$: Andy is a student
  - $I(\text{paul})$: Paul is an instructor
  - $Y(\text{andy}, \text{paul})$: Andy is younger than Paul
  - S, I, and Y are called predicates

Motivation and Introduction

Example : "***Every*** student is younger than ***some*** instructor"

- How to formalize "every" and "some" ?
- How can we reflect individuals to the predicates?
- Variables are used to specify the meanings of predicates more formally
  - $S(x)$: x is a student
  - $I(x)$: x is an instructor
  - $Y(x, y)$: x is younger than y
- Quantifiers $\forall$(read: 'for all') and $\exists$(read: 'there exists' or 'for some')
  - $\forall x(S(x) \to (\exists y(I(y) \wedge Y(x, y))))$
  - *For every x, if x is a student, then there is some y which is an instructor such that x is younger than y*

Formal Predicate Logic

The first thing to note is that there are two sorts of things involved in a predicate logic formula

- Terms : expressions in predicate logic which denote *objects*
- Formulas

Predicate vocabulary

- Predicate symbols (P): $B(\cdot)$, $M(\cdot,\cdot)$, etc. denote fixed relations "$\cdot$ is a bird", "$\cdot$ is mother of"
- Function symbols (F): represent in more direct way. ($m(y)$ means y's mother)
- Constant symbols (C): don't be used, but rather, constants are stipulated as 0-arity, so-called *nullary*, function

Terms

Definition (Terms are defined)

- Any variable is a term
- If $c \in F$ is a nullary function, then c is a term
- If $t_1, t_2, \ldots, t_n$ are terms and $f \in F$ has arity n > 0, then $f(t_1, t_2, \ldots, t_n)$ is a term
- Nothing else is a term

In Backus Naur form, *terms* may be written,

- $t ::= x \mid c \mid f(t, \ldots, t)$
  - x ranges over a set of variables, c over nullary function symbols in F, and f over those elements of F with arity n>0

Terms

Note!!

- the first building blocks of terms are *constants* (nullary functions) and *variables*
- more complex terms are built from function symbols using as many previously built terms as required by such function symbols
- the notion of terms is dependent on the set $\boldsymbol{F}$. If you change it, you change the set of terms.

Example of terms

- Suppose n, f and g are function symbols, respectively nullary, unary, binary
    - $g(f(n),n)$ and $f(g(n,f(n)))$: terms
    - $g(n)$ and $f(f(n),n)$: not (violation of the arities)

Formulas

Preliminary of formula

The choice of sets P and F is driven by what we intend to describe

- ex) DB representing relations between our kin
    - $P = \{M, F, S, D\}$ (*being male, being female, being a son of... and being a daughter of ...*)
        - F and M are unary predicates whereas D and S are binary
    - may define $F = \{mother - of, father - of\}$

Formulas

Definition (Definitions of Formulas)

We define the set of formulas over (F, P) inductively, using the already defined set of terms over F :

- If $P \in P$ is a predicate symbol of arity n ≥ 1, and if $t_1, t_2, ..., t_n$ are terms over F, then $P(t_1, t_2, \ldots, t_n)$ is a formula
- If ϕ is a formula, then so is $(\neg\phi)$
- If ϕ and ψ are formulas, then so are $(\phi \wedge \psi)$, $(\phi \vee \psi)$ and $(\phi \rightarrow \psi)$
- If ϕ is a formula and x is a variable, then $(\forall x\phi)$ and $(\exists x\phi)$ are formulas
- Nothing else is a formula

Backus Naur form (BNF)

$\phi ::= P(t_1, t_2, ..., t_n) \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi) \mid (\forall x\phi) \mid (\exists x\phi)$

Formulas

Convention

- Binding priority
  - $\neg$, $\forall y$ and $\exists y$ bind most tightly
  - then $\vee$ and $\wedge$
  - then $\rightarrow$, which is right-associative
- often omit brackets around quantifiers

Formulas

Exmaple - translating sentence

“Every son of my father is my brother”

- Design choice : whether we represent ‘father’ as a predicate or as a function symbol

predicate

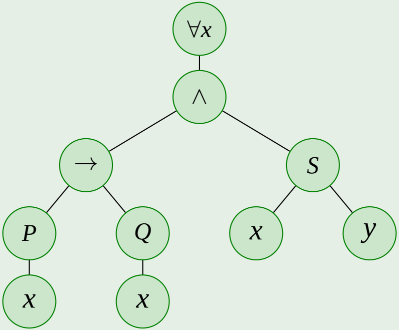
- a constant m for ‘me’ or ‘I’
- $S(x, y)$: x is a son of y
- $F(x, y)$: x is the father of y
- $B(x, y)$: x is a brother of y
- $\forall x \forall y (F(x, m) \wedge S(y, x) \rightarrow B(y, m))$

function

- keep m, S and B as a *predicate*
- f : the function which, given an argument, returns the corresponding *father*
- $\forall x (S(x, f(m)) \rightarrow B(x, m))$

Boundness of variables

Parse tree of predicate formular



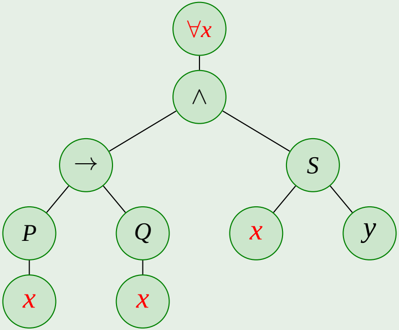
$\forall x((P(x) \rightarrow Q(x)) \wedge S(x,y))$

two additional sorts of nodes

- The quantifiers $\forall x$ and $\exists y$ form nodes and have just one subtree
- Predicate expressions, which are generally of the form $P(t_1, t_2, \ldots, t_n)$, have the symbol P as a node, but now P has n many subtrees, namely the parse trees of the terms $t_1, t_2, \ldots, t_n$

Boundness of variables

Parse tree of predicate formular



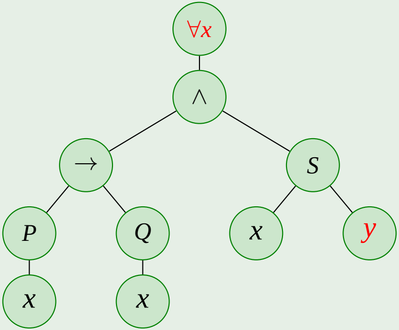
$\forall x((P(x) \rightarrow Q(x)) \wedge S(x, y))$

two additional sorts of nodes

- When walking up the tree beginning at any one of these x leaves, we run into the quantifier $\forall x$. $\Rightarrow$ occurrences of x are actually bound to $\forall x$
- In walking upwards, the only quantifier that the leaf node y runs into is $\forall x$ but that x has nothing to do with $y \Rightarrow$ So y is *free* in this formula

Boundness of variables

Parse tree of predicate formular



$\forall x((P(x) \rightarrow Q(x)) \wedge S(x,y))$

two additional sorts of nodes

- When walking up the tree beginning at any one of these x leaves, we run into the quantifier $\forall x$. $\Rightarrow$ occurrences of x are actually bound to $\forall x$
- In walking upwards, the only quantifier that the leaf node y runs into is $\forall x$ but that x has nothing to do with y $\Rightarrow$ So y is *free* in this formula

Definition of Free and Bound Variables

Definition (Free and bound variables)

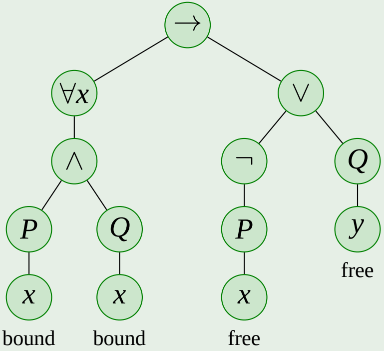
Let ϕ be a formula in predicate logic. An occurrence of x in ϕ is *free* in ϕ if it is a leaf node in the parse tree of ϕ such that there is no path upwards from that node x to a node $\forall x$ or $\exists x$. Otherwise, that occurrence of x is called *bound*. For $\forall x \phi$, or $\exists x \phi$, we say that ϕ - minus any of ϕ's subformulas $\exists x \psi$, or $\forall x \psi$ - is the scope of $\forall x$, respectively $\exists x$

Example (scope of variable)

- the scope of $\forall x$ in $\forall x(P(x) \rightarrow \exists x Q(x))$ is $P(x)$
- $(\forall x(P(x) \wedge Q(x))) \rightarrow (\neg P(x) \vee Q(y))$

Example to scope of variable

$(\forall x(P(x) \land Q(x))) \to (\neg P(x) \lor Q(y))$



Replacing variable with term

Variables are place holders so we must have some means of replacing the with more concrete information

Definition (Substitution)

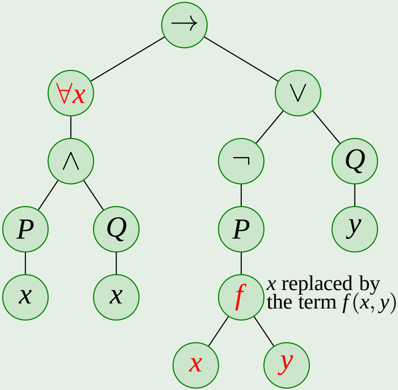
Given a variable x, a term t and a formula ϕ we define $\phi[t/x]$ to be the formula obtained by replacing each free occurrence of variable x in ϕ with t.

Definition (Free substitution)

Given a term t, a variable x and a formula ϕ, we say that t is free for x in ϕ if no free x leaf in ϕ occurs in the scope of $\forall y$ or $\exists y$ for any variable y occurring in t

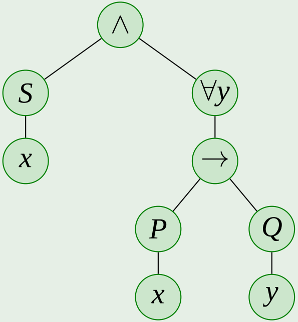
Example of Substitution

A parse tree of a formula resulting from substitution



x replaced by the term $f(x, y)$

A parse tree for which a substitution has dire consequence



the term $f(y, y)$ is not free for x in this formula

Natural deduction for predicate logic

- similar to proofs in the natural deduction calculus for propositional logic
- new proof rules for dealing with the quantifiers and with the equality symbol
  - The proof rules for equality
    - *cf*) *Equality* is a special predicate which is a binary predicate and is written =. Unlike other predicates, it is usually written in between its arguments rather than before them (e.g. x=y instead of $=(x, y)$ to say that x and y are equal
    - Equality does not mean syntactic, or intensional, equality, but equality in terms of computation results.
  - The proof rules for universal quantification
  - The proof rules for existential quantification

The proof rules for equality

Introduction rule for equality

Any term *t* has to be equal to itself

- $\dfrac{}{t=t}=\mathrm{i}$

Elimination rule for equality

- $\dfrac{t_1=t_2 \qquad \phi[t_1/x]}{\phi[t_2/x]}=\mathrm{e}$

(Using substitution principle)

example

We obtain proof

1	$(x+1)=(1+x)$	premise
2	$(x+1>1)\rightarrow(x+1>0)$	premise
3	$(1+x>1)\rightarrow(1+x>0)$	=e 1,2

establishing the validity of the sequent

$$x+1=1+x,\ (x+1>1)\rightarrow(x+1>0)\ \vdash\ (1+x)>1\rightarrow(1+x)>0$$

The proof rules for equality

Properties of Equality

- $t_1 = t_2 \vdash t_2 = t_1$
- $t_1 = t_2, t_2 = t_3 \vdash t_1 = t_3$

A proof for $t_1 = t_2 \vdash t_2 = t_1$ (where ϕ is $x = t_1$.):

1	$t_1 = t_2$	premise
2	$t_1 = t_1$	=i
3	$t_2 = t_1$	=e 1,2

A proof for $t_1 = t_2, t_2 = t_3 \vdash t_1 = t_3$ (where ϕ is $t_1 = x$.):

1	$t_2 = t_3$	premise
2	$t_1 = t_2$	premise
3	$t_1 = t_3$	=e 1,2

The discussion of the rules =i and =e has shown that they force equality to be *reflexive*, *symmetric* and *transitive*

The proof rules for universal quantification

The rule for eliminating $\forall$

$$\frac{\forall x \phi}{\phi[t/x]}\ \forall x e$$

- Condition: $\forall x \phi$ is true
- Side Cond.: t be free for x in ϕ
- Conclusion: $\phi[t/x]$ is true as well

The rule for introducing $\forall$

$$\frac{\boxed{\begin{array}{l} x_0 \\ \quad \vdots \\ \quad \phi[x_0/x] \end{array}}}{\forall x \phi}\ \forall x\ \mathrm{i}$$

- Using proof box similar
- The box is to stipulate the scope of the ‘dummy variable’ x_0

The proof rules for universal quantification

example of $\forall x$e and $\forall x$i

1		$\forall x(P(x) \to Q(x))$	premise
2		$\forall x P(x)$	premise
3	x_0	$P(x) \to Q(x)$	$\forall x$ e 1
4		$P(x_0)$	$\forall x$ e 2
5		$Q(x_0)$	$\to$ e 3, 4
6		$\forall x Q(x)$	$\forall x$ i 3-5

example of $\forall x$e and $\forall x$i

1	$P(t)$	premise
2	$\forall x(P(x) \to \neg Q(x))$	premise
3	$P(t) \to \neg Q(t))$	$\forall x$ e 2
4	$\neg Q(t)$	$\to$ 3, 1

The proof rules for existential quantification

There is the analogy between $\forall$ *and* $\wedge$*, and also* $\exists$ *and* $\vee$

Exists-Introduction

- Let's consider or-introduction and and-elimination
  - $\dfrac{\phi_1 \wedge \phi_2}{\phi_k} \wedge\mathrm{e}_k \quad \dfrac{\phi_k}{\phi_1 \vee \phi_2} \vee\mathrm{i}_k$
- given the form of forall-elimination, we can infer that exists-introduction must be simply

$$\frac{\phi[t/x]}{\exists x \phi} \; \exists x \text{ i}$$

Exists-Elimination

$$\frac{\exists x \phi \quad \boxed{\begin{array}{ll} x_0 & \phi[x_0/x] \\ & \vdots \\ & \chi \end{array}}}{\chi} \; \exists \text{ e}$$

- The proof box is is controlling two things : the scope of x_0 and also the scope of the assumption $\phi[x_0/x]$

The proof rules for existential quantification

On the Exists-Elimination

- Like $\vee$e, it involves a *case analysis*. The reasoning goes: We know $\exists x \phi$ is true, so ϕ is true for *at least one 'value' of* x. So we do a case analysis over all those possible values, writing x_0 as a generic value representing them all.
- If assuming $\phi[x_0/x]$ allows us to prove some χ which doesn't mention x_0, then this χ must be true whichever x_0 makes $\phi[x_0/x]$ true
- Just as $\vee$e says that to use $\phi_1 \vee \phi_2$, you have to be prepared for either of the ϕ_i, so $\exists$e says that to use $\exists x \phi$ you have to be prepared for any possible $\phi[x_0/x]$. Another way of thinking about $\exists$e goes like this: If you know $\exists x \phi$ and you can derive some χ from $\phi[x_0/x]$, i.e. by giving a name to the thing you know exists, then you can derive χ even without giving that thing a name (provided that χ does not refer to the name x_0).

Examples for quantifiers

Example (Proving the validity of the sequent $\forall x\phi \vdash \exists x\phi$)

1	$\forall x\phi$	premise
2	$\phi[x/x]$	$\forall x$e 1
3	$\exists x\phi$	$\exists x$i 2

Example (Proving the validity of the sequent $\forall x(P(x) \rightarrow Q(x))\ \exists xP(x) \vdash \exists xQ(x)$)

1		$\forall(P(x) \rightarrow Q(x))$	premise
2		$\exists P(x)$	premise
3	x_0	$P(x_0)$	assumption
4		$P(x_0) \rightarrow Q(x_0)$	$\forall x$e 1
5		$Q(x_0)$	$\rightarrow$e 4,3
6		$\exists xQ(x)$	$\exists x$i 5
7		$\exists xQ(x)$	$\exists x$e 2,3-6

Examples for quantifiers

Example (Illegal proof for $\forall x(P(x) \to Q(x))\ \exists x P(x) \vdash \exists x Q(x)$)

1		$\forall(P(x) \to Q(x))$	premise
2		$\exists P(x)$	premise
3	x_0	$P(x_0)$	assumption
4		$P(x_0) \to Q(x_0)$	$\forall x$e 1
5		$Q(x_0)$	$\to$e 4,3
6		$Q(x_0)$	$\exists x$e 2,3-5
7		$\exists x Q(x)$	$\exists x$i 6

- Line 6 allows the fresh parameter x_0 to escape the scope of the box which declares it

Examples for quantifiers

Example argument: *If all quakers are reformists and if there is a protestant who is also a quaker, then there must be a protestant who is also a reformist.*

Proof for $\forall x(Q(x) \rightarrow R(x)), \exists(P(x) \wedge Q(x)) \vdash \exists(P(x) \wedge R(x))$.

1		$\forall(Q(x) \rightarrow R(x))$	premise
2		$\exists(P(x) \wedge Q(x))$	premise
3	x_0	$P(x_0) \wedge Q(x_0)$	assumption
4		$Q(x_0) \rightarrow R(x_0)$	$\forall x$e 1
5		$Q(x_0)$	$\wedge e_2$ 3
6		$R(x_0)$	$\rightarrow$e 4,5
7		$P(x_0)$	$\wedge e_1$ 3
8		$P(x_0) \wedge R(x_0)$	$\wedge$ i 7,6
9		$\exists x(P(x) \wedge R(x))$	$\exists x$i 8
10		$\exists x(P(x) \wedge R(x))$	$\exists x$e 2,3-9

□

Examples for quantifiers

Note 1 The rules ∀i and ∃e both have the side condition that the dummy variable cannot occur outside the box in the rule.

Note 2 These rules may still be nested, by choosing another fresh name (e.g. y_0) for the dummy variable

Example ($\exists P(x), \forall x \forall y (P(x) \to Q(y)) \vdash \forall y Q(y)$)

1		$\exists P(x)$	premise
2		$\forall x \forall y (P(x) \to Q(y))$	premise
3	y_0		
4	x_0	$P(x_0)$	assumption
5		$\forall y (P(x_0) \to Q(y))$	$\forall x$e 2
6		$P(x_0) \to Q(y)$	$\forall y$e 5
7		$Q(y_0)$	$\to$e 6,4
8		$Q(y_0)$	$\exists x$e 1, 4-7
9		$\forall y Q(y)$	$\forall y$i 3-8

Theorem for Quantifier Equivalences

Example

‘Not all birds can fly’ is specified as

- $\neg\forall x(B(x) \to F(x))$

or as

- $\exists x(B(x) \wedge \neg F(x))$

The former formal specification is closer to the structure of the English specification, but the latter is logically equivalent to the former.

Theorem for Quantifier Equivalences

Theorem (Quantifier Equivalences)

1. **(a)** $\neg\forall x\phi \dashv\vdash \exists x\neg\phi$
   (b) $\neg\exists x\phi \dashv\vdash \forall x\neg\phi$

2. *Assuming that x is not free in ψ:*

(a) $\forall x\phi\wedge\psi \dashv\vdash \forall x(\phi\wedge\psi)$
(b) $\forall x\phi\vee\psi \dashv\vdash \forall x(\phi\vee\psi)$
(c) $\exists x\phi\wedge\psi \dashv\vdash \exists x(\phi\wedge\psi)$
(d) $\exists x\phi\vee\psi \dashv\vdash \exists x(\phi\vee\psi)$
(e) $\forall x(\psi \to \phi) \dashv\vdash \psi \to \forall x\phi$
(f) $\exists x(\phi \to \psi) \dashv\vdash \forall x\phi \to \psi$
(g) $\forall x(\phi \to \psi) \dashv\vdash \exists x\phi \to \psi$
(h) $\exists x(\psi \to \phi) \dashv\vdash \psi \to \exists x\phi$

3. **(a)** $\forall x\phi \wedge \forall x\psi \dashv\vdash \forall x(\phi \wedge \psi)$
   (b) $\exists x\phi \vee \exists x\psi \dashv\vdash \exists x(phi \vee \psi)$

4. **(a)** $\forall x\forall y\phi \dashv\vdash \forall y\forall x\phi$
   (b) $\exists x\exists y\phi \dashv\vdash \exists y\exists x\phi$