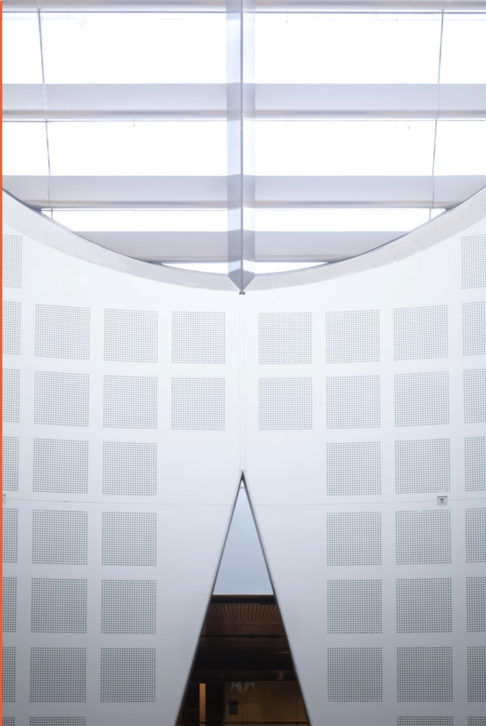
COMP2022|2922
Models of Computation

Introduction to Predicate Logic

Sasha Rubin

October 11, 2022

THE UNIVERSITY OF SYDNEY



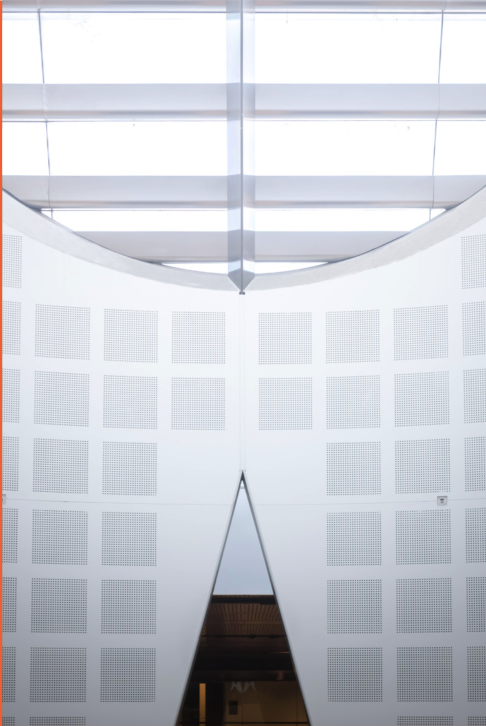
COMP2022|2922
Models of Computation

Equivalences in Predicate Logic

Sasha Rubin

October 11, 2022

THE UNIVERSITY OF SYDNEY



Formulas that “mean the same thing” are called equivalent. We now study common equivalences, also called laws.

Equivalences

F and G are logically equivalent ($F \equiv G$) if the truth value of F under α equals the truth value of G under α, for all assignments, and all domains and predicates.

All equivalences for propositional logic also hold for predicate logic.

Example (De Morgan's Law)

$\neg(\exists x P(x) \wedge Q(y))$ and $\neg\exists x P(x) \vee \neg Q(y)$ are equivalent.

Equivalences involving quantifiers

For all formulas F, G:

(Q. Negation)
$$\neg \forall x F \equiv \exists x \neg F$$
$$\neg \exists x F \equiv \forall x \neg F$$

(Q. Unification)
$$(\forall x F \wedge \forall x G) \equiv \forall x (F \wedge G)$$
$$(\exists x F \vee \exists x G) \equiv \exists x (F \vee G)$$

(Q. Transposition)
$$\forall x \forall y F \equiv \forall y \forall x F$$
$$\exists x \exists y F \equiv \exists y \exists x F$$

(Q. Extraction) if $x \notin \mathrm{Free}(G)$:
$$(\forall x F \wedge G) \equiv \forall x (F \wedge G)$$
$$(\forall x F \vee G) \equiv \forall x (F \vee G)$$
$$(\exists x F \wedge G) \equiv \exists x (F \wedge G)$$
$$(\exists x F \vee G) \equiv \exists x (F \vee G)$$

Equivalences

Here are informal reasons behind some of these equivalences:[1]

1. $\neg\forall x F \equiv \exists x \neg F$
   - the LHS says that not all x satisfy F,
   - which means the same thing as some x doesn't satisfy F,
   - which means that some x does satisfy $\neg F$,
   - which is what the RHS says.
2. $(\forall x F \wedge \forall x G) \equiv \forall x(F \wedge G)$
   - the LHS says that F holds for every x and G holds for every x,
   - which is the same as saying both F and G hold for every x,
   - which is what the RHS says.
3. $\forall x \forall y F \equiv \forall y \forall x F$
   - Both sides say that F holds for all values of the listed variables.
4. $(\forall x F \wedge G) \equiv \forall x(F \wedge G)$ if $x \notin \mathrm{Free}(G)$
   - LHS says F holds for every x, and G holds.
   - RHS says F and G hold for every x; but G doesn't depend on the value of x.

[1] To prove them formally, use the inductive definition of truth-value.

Equivalences

Example

Show that $\neg(\exists x P(x,y) \vee \forall z \neg R(z)) \equiv \forall x \exists z(\neg P(x,y) \wedge R(z))$

$$
\begin{aligned}
&\neg(\exists x\, P(x,y) \vee \forall z \neg R(z)) \\
&\qquad \equiv (\neg\exists x P(x,y) \wedge \neg\forall z \neg R(z)) && \text{DeMorgan's Laws} \\
&\qquad \equiv (\forall x \neg P(x,y) \wedge \exists z \neg\neg R(z)) && \text{Quantifier Negation} \\
&\qquad \equiv (\forall x \neg P(x,y) \wedge \exists z R(z)) && \text{Double Negation} \\
&\qquad \equiv \forall x(\neg P(x,y) \wedge \exists z R(z)) && \text{Quantifier Extraction} \\
&\qquad \equiv \forall x(\exists z R(z) \wedge \neg P(x,y)) && \text{Comm } \wedge \\
&\qquad \equiv \forall x \exists z(R(z) \wedge \neg P(x,y)) && \text{Quantifier Extraction} \\
&\qquad \equiv \forall x \exists z(\neg P(x,y) \wedge R(z)) && \text{Comm. } \wedge
\end{aligned}
$$

$$\forall x \exists z (\neg P(x, y) \wedge R(z))$$

This formula has a very nice shape... all the quantifiers are out the front! this can make it easier to understand and manipulate.

Normal Forms

Definition
A formula F is in negation normal form (NNF) if negations only occur immediately infront of atomic formulas.

$\neg P(x) \to Q(y)$ is in NNF

$\neg(P(x) \to Q(y))$ is not in NNF

Normal Forms

Theorem

For every formula F there is an equivalent formula in NNF.

Algorithm ("push negations inwards by applying Q. Negation and DM")

Substitute in F every occurence of a subformula of the form $\neg\neg G$ by G, and

$$\neg\forall x F \text{ by } \exists x \neg F \qquad \neg\exists x F \text{ by } \forall x \neg F$$
$$\neg(G \land H) \text{ by } (\neg G \lor \neg H) \qquad \neg(G \lor H) \text{ by } (\neg G \land \neg H)$$

until no such subformulas occur, and return the result.

Normal Forms

Definition
A formula F is in prenex normal form (PNF) if it has the form

$$Q_1 x_1 Q_2 x_2 \cdots Q_n x_n F$$

where each $Q_i \in \{\exists, \forall\}$ is a quantifier symbol, the x_is are variables, $n \geq 0$ (so, there may be no quantifiers in the prefix), and F does not contain a quantifier.

$\forall x \exists y (P(x) \vee L(x, y))$ is in PNF.
$\forall x (P(x) \vee \exists y L(x, y))$ is not in PNF.

Normal Forms

Theorem

For every formula F there is an equivalent formula in PNF.

Algorithm ("pull quantifiers out the front by applying Q. Extraction")

1. Put F in NNF, call the result F'.
2. Substitute in F' every occurence of a subformula of the form

$$\begin{gathered}(\forall x F \wedge G) \text{ by } \forall x(F \wedge G)\\(\forall x F \vee G) \text{ by } \forall x(F \vee G)\\(\exists x F \wedge G) \text{ by } \exists x(F \wedge G)\\(\exists x F \vee G) \text{ by } \exists x(F \vee G)\end{gathered}$$

   until no such subformulas occur (use commutativity to handle $(G \wedge \forall x F)$, etc.), and return the result.

   NB. To apply these equivalences we need that $x \notin \mathrm{Free}(G)$.
   This can always be achieved by renaming the bound variable x.

Logical consequence

Definition

A sentence F is a logical consequence of the set $E_1, \cdots, E_k$ of sentences if for every domain, predicates, assignments, if all of the $E_1, \cdots, E_k$ are true, also F is true. In this case we write

$$E_1, \cdots, E_k \models F$$

Example

- $\forall x\, R(x,x)$ is a logical consequence of $\forall x \forall y\, R(x,y)$.
- $P(c)$ is a logical consequence of $Q(c), \forall x\, (Q(x) \rightarrow P(x))$.

In case $\models F$ we say that F is valid.

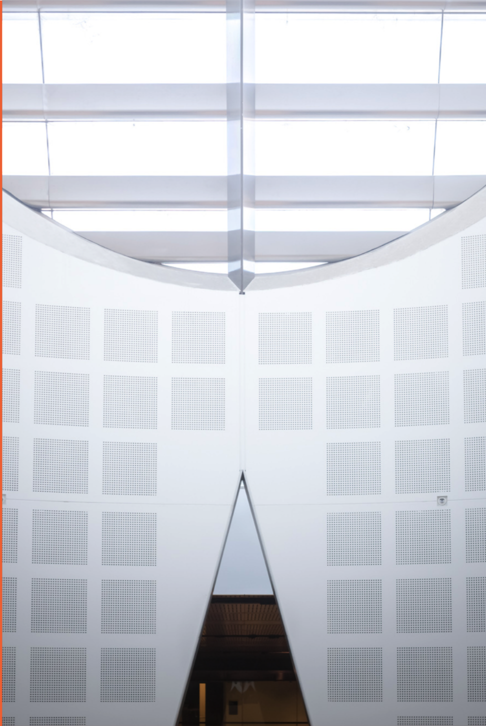
COMP2022|2922 Models of Computation

ND for predicate logic

Sasha Rubin

October 11, 2022

THE UNIVERSITY OF SYDNEY



Deductive systems are a syntactic mechanism for deriving validities as well as logical consequences from assumptions

Natural deduction

- We extend ND for propositional logic with rules to handle quantifiers.
- Each quantifier symbol $\exists, \forall$ has two types of rules:
    1. Introduction rules introduce the quantifier
    2. Elimination rules remove the quantifier

Replacing free variables by constants.

Definition

For a formula F, variable x, constant c, we can obtain a formula

$$F[c/x]$$

by simultaneously replacing all free occurrences of x in F by c.

The idea is that whatever F said about x, now $F[c/x]$ says about c.

$\forall$ elimination

$(\forall E)$	$\dfrac{\forall x F}{F[c/x]}$

$(\forall E)$ formalises the reasoning

> *If we know that F holds for every x, then it must hold, in particular, taking $x = c$*

$\forall$ introduction

$(\forall I)$	$\dfrac{F[c/x]}{\forall x F}$
	where c is a constant, not occuring in F, nor in any of the assumptions of $F[c/x]$.

$(\forall I)$ formalises the reasoning

> *Let c be any element ... (insert proof of $F[c/x]$). Since c was arbitrary, deduce F holds for all x.*

That c is arbitrary is captured by requiring that c is not in the assumptions used to prove $F[c/x]$, and so c is not constrained in any way.

$\forall x \forall y P(x,y) \vdash \forall y \forall x P(x,y)$

Plan: instantiate the variables to new constants, then introduce them in the reverse order.

Line	Assumptions	Formula	Justification	References
1	1	$\forall x \forall y P(x,y)$	Asmp. I	
2	1	$\forall y P(c,y)$	$\forall$ E	1
3	1	$P(c,d)$	$\forall$ E	2
4	1	$\forall x P(x,d)$	$\forall$ I *	3
5	1	$\forall y \forall x P(x,y)$	$\forall$ I **	4

* the constant c does not occur in F (i.e., $P(x,d)$), nor in the formula of its assumption (in line 1).

** the constant d does not occur in F (i.e., $\forall x P(x,y)$), nor in the formula of its assumption (in line 1).

$$\forall x(P(x) \land Q(x)) \vdash \forall x P(x) \land \forall x Q(x)$$

Plan: instantiate the variable to a new constant, split, then introduce the variables back.

Line	Assumptions	Formula	Justification	References
1	1	$\forall x(P(x) \land Q(x))$	Asmp. I	
2	1	$P(c) \land Q(c)$	$\forall$ E	1
3	1	$P(c)$	$\land$ E	2
4	1	$Q(c)$	$\land$ E	2
5	1	$\forall x P(x)$	$\forall$ I *	3
6	1	$\forall x Q(x)$	$\forall$ I *	5
7	1	$\forall x P(x) \land \forall x Q(x)$	$\land$ I	4,6

* c does not appear in $P(x)$ nor in the assumption 1

What is wrong with the following "proof" of $P(c) \vdash \forall x P(x)$?

Line	Assumptions	Formula	Justification	References
1	1	$P(c)$	Asmp. I	
2	1	$\forall x P(x)$	$\forall$ I	1

What is wrong with the following "proof" of $P(c) \vdash \forall x P(x)$?

Line	Assumptions	Formula	Justification	References
1	1	$P(c)$	Asmp. I	
2	1	$\forall x P(x)$	$\forall$ I	1

$(\forall I)$	$\dfrac{F[c/x]}{\forall x F}$ where c is a constant, not occuring in F, nor in any of the assumptions of $F[c/x]$.

- $F = P(x)$
- $F[c/x] = P(c)$
- The assumption of $F[c/x]$ is $P(c)$.

$\exists$ Introduction

$(\exists I)$	$\dfrac{F[c/x]}{\exists x F}$

$(\exists I)$ formalises the reasoning

> *If we know that F holds for a specific constant c, then we know it holds for some x.*

$\forall x P(x) \vdash \exists x P(x)$

Plan: instantiate x arbitrarily, then introduce x existentially.

Line	Assumptions	Formulas	Just.	Ref.
1	1	$\forall x P(x)$	Asmp. I	
2	1	$P(c)$	$\forall$ E	1
3	1	$\exists x P(x)$	$\exists$ I	2

$\neg\exists x \neg P(x) \vdash \forall x P(x)$

Plan: take a fresh constant c, assume $\neg P(c)$, get a contradiction, deduce $P(c)$, and then that $\forall x P(x)$ since c was arbitrary.

Line	Asmp.	Form.	Just.	Ref.
1	1	$\neg\exists x \neg P(x)$	Asmp. I	
2	2	$\neg P[c/x]$	Asmp. I	
3	2	$\exists x \neg P(x)$	$\exists I$	2
4	1,2	$\perp$	$\perp I$	1,3
5	1	$P[c/x]$	$\neg E$	2,4
6	1	$\forall x P(x)$	$\forall I$*	5

* c does not appear in $P(x)$ nor in the assumption 1

$\exists$ Elimination

$(\exists E)$	$$\frac{\exists x F \qquad F[c/x] \vdash G}{G}$$ where c is a constant symbol, not occuring in F, nor in G, nor in any assumption used in the proof of G except for $F[c/x]$

$(\exists E)$ formalises the reasoning

> *From $\exists x F$ we know there is an x that satisfies F, so we take one and call it c. If c is new, and has not been used so far, and we manage to derive G, then we can deduce G from that weaker assumption that there is some x that satisfies F (even if we don't know which one).*

$\exists$ Elimination

$(\exists E)$	$$\frac{\exists x F \qquad F[c/x] \vdash G}{G}$$ where c is a constant symbol, not occuring in F, nor in G, nor in any assumption used in the proof of G except for $F[c/x]$

How to use $(\exists E)$?

1. Assume $F[c/x]$ ensuring that c does not occur in F.
2. Derive G making sure that c is not in the assumption set of G except for $F[c/x]$.
3. Cancel the assumption $F[c/x]$, and conclude G.

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

Line	Assumptions	Formulas	Just.	Ref.
1	1	$\forall x(Q(x) \to P(y))$	Asmp. I	
2	2	$\exists x Q(x)$	Asmp. I	
3	1	$Q(c) \to P(y)$	$\forall$ E	1
4	4	$Q(c)$	Asmp. I	
5	1,4	$P(y)$	$\to$ E	3,4
6	1,2	$P(y)$	$\exists$ E	2,4,5

What is wrong with the following "proof" of $\exists x P(x) \vdash \forall x P(x)$?

Line	Asmp.	Form.	Just.	Ref.
1	1	$\exists x P(x)$	Asmp. I	
2	1	$P(c)$	$\exists$ E	1
3	1	$\forall x P(x)$	$\forall$ I	2

Here is the faulty argument in natural language:

1. *We are given that P is satisfied by some x.*
2. *Let c be such an x.*
3. *Since c was chosen arbitrarily (?!), conclude that every x satisfies P.*

Wrapping up

ND for predicate logic.

- Allows a machine to check if a given proof is correct.
- It is sound and complete.

$$E_1, \cdots, E_k \models F \text{ if and only if } E_1, \cdots, E_k \vdash F$$

- However, unlike Propositional Logic, the problem of checking if F is a logical consequence of $E_1, \cdots, E_k$ is undecidable.
- This means that finding proofs of Predicate Logic cannot be fully automated.