Predicate Logic

Propositional Logic is not enough to prove:

All students are rich
Chris is a student
Therefore, Chris is rich
(Have to use three different letters)

Quantificational Logic

Ir	=	Romeo is Italian.
Ix	=	x is Italian.
(x)Ix	=	For all x, x is Italian.
	=	All are Italian.
(∃x)Ix	=	For some x, x is Italian.
	=	Some are Italian.

Conventionally, we often use:
First-order logic, Predicate logic

I(r)
I(x)

∀x I(x)

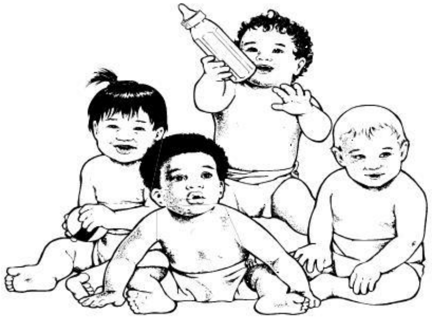
(Both are fine in exams)

Use capital letters for *general terms* (terms that *describe* or put in a *category*):

B = a cute baby
C = charming
R = rides a bicycle

Use small letters for *singular terms* (terms that pick out a *specific* person or thing):

b = the world's cutest baby
c = this child
w = William Gensler



A capital letter alone (not followed by small letters) represents a *statement*.	S = *It is snowing.*

Propositional logic

A capital letter followed by a single small letter represents a *general term*.	Ir = Romeo is *Italian*.

I(r) is OK too

A small letter from "a" to "w" is a *constant* – and stands for a specific person or thing.	Ir = *Romeo* is Italian.

I(romeo), I(r1) are OK

A small letter from "x" to "z" is a *variable* – and stands for an unspecified member of a class of things.	Ix = *x* is Italian.

I(x1) ok

"(x)" is a *universal quantifier*. It claims that the formula that follows is true for *all* values of x.

(x)Ix = For all x, x is Italian.
= All are Italian.

"(∃x)" is an *existential quantifier*. It claims that the formula that follows is true for *at least one* value of x.

(∃x)Ix = For some x, x is Italian.
= Some are Italian.

In the "domain" or "universe"

1. The result of writing a capital letter and then a small letter is a wff.
2. The result of writing a quantifier and then a wff is a wff.

LogiCola H (EM & ET) Pages 101 to 105

If the English begins with	then begin the wff with
all (every)	(x)
not all (not every)	$\sim(x)$
some	$(\exists x)$
no	$\sim(\exists x)$

LogiCola H (EM, ET)

All are Italian = $(x)Ix$
Not all are Italian = $\sim(x)Ix$

Some are Italian = $(\exists x)Ix$
No one is Italian = $\sim(\exists x)Ix$

All are rich or Italian = $(x)(Rx \vee Ix)$
Not everyone is non-Italian = $\sim(x)\sim Ix$
Some aren't rich = $(\exists x)\sim Rx$
No one is rich and non-Italian = $\sim(\exists x)(Rx \cdot \sim Ix)$

With "all … is …," use "$\supset$" for the *middle* connective.

Otherwise use "•" for the connective.

All Italians are lovers = $(x)(Ix \supset Lx)$
= For all x, *if* x is Italian *then* x is a lover.

Some Italians are lovers = $(\exists x)(Ix \cdot Lx)$
= For some x, x is Italian *and* x is a lover.

No Italians are lovers = $\sim(\exists x)(Ix \cdot Lx)$
= It is not the case that, for some x, x is Italian *and* x is a lover.

All rich Italians are lovers = $(x)((Rx \cdot Ix) \supset Lx)$
= For all x, *if* x is rich *and* Italian, *then* x is a lover.

Notice the difference

If we use (x) Ix & Lx, …

If we use Ex (Ix -> Lx)

…

Quantificational Logic

Ir	=	Romeo is Italian.		
Ix	=	x is Italian.		
(x)Ix	=	All are Italian	=	For all x, x is Italian.
(∃x)Ix	=	Some are Italian	=	For some x, x is Italian.

If the English begins with:	➔	all (every)	not all	some	no
then begin the wff with:	➔	(x)	~(x)	(∃x)	~(∃x)

With "all … is …," use "⊃" for the *middle* connective.

Otherwise use "•" for the connective.

Quantificational Inference Rules

First reverse squiggles	$\sim(x)Fx \rightarrow (\exists x)\sim Fx$ $\sim(\exists x)Fx \rightarrow (x)\sim Fx$	*	Additional S/I rules (No "Truth-table" method)
and drop existentials;	$(\exists x)Fx \rightarrow Fa$, use a *new* constant	*	Non-specific, so new constant
lastly, drop universals.	$(x)Fx \rightarrow Fa$, use any constant	Don't star	Can be any constant

	1	(x)(Fx · Gx)	
		[∴ (x)Fx	
*	2	┌ asm: ~(x)Fx	
*	3	│ ∴ (∃x)~Fx {from 2}	← reverse squiggles
	4	│ ∴ ~Fa {from 3}	← drop existentials
	5	│ ∴ (Fa · Ga) {from 1}	← drop universals
	6	└ ∴ Fa {from 5}	
	7	∴ (x)Fx {from 2; 4 contradicts 6}	

Valid

LogiCola IEV

1. Reverse squiggles.
2. Drop initial existentials, using a new letter each time.
3. Lastly, drop initial universals, using all the old letters. (Only use a new letter if you've done everything else possible, including further assumptions if needed, and still have no old letters.)

	1	$(x)(Lx \supset Fx)$	
*	2	$(\exists x)Lx$	
		[$\therefore (x)Fx$	
*	3	asm: $\sim(x)Fx$	
*	4	$\therefore (\exists x)\sim Fx$	{from 3}
	5	$\therefore La$	{from 2}
	6	$\therefore \sim Fb$	{from 4}
*	7	$\therefore (La \supset Fa)$	{from 1}
*	8	$\therefore (Lb \supset Fb)$	{from 1}
	9	$\therefore Fa$	{from 5 and 7}
	10	$\therefore \sim Lb$	{from 6 and 8}

Invalid

a, b

La, Fa
$\sim Lb, \sim Fb$

LogiCola I (EI, EC)

Reverse squiggles, drop existentials, drop universals. If you can't get a contradiction, construct a refutation.

1	$(x)(Lx \supset Fx) = 1$	
2	$(\exists x)Lx = 1$	
	$[\therefore (x)Fx = 0$	

Invalid

a, b

La, Fa
~Lb, ~Fb

An *existential* wff is true if and only if *at least one case* is true.

A *universal* wff is true if and only if *all cases* are true.

If a wff doesn't start with a quantifier: evaluate each *part* that starts with a quantifier, and then substitute "1" or "0" for it:

$$\begin{aligned} &\sim(x)Fx \\ = &\sim 0 \\ = &\ 1 \end{aligned} \qquad \begin{aligned} &\sim(x)(Lx \supset Fx) \\ = &\sim 1 \\ = &\ 0 \end{aligned} \qquad \begin{aligned} &((\exists x)Fx \supset (x)Lx) \\ = &\ (1 \supset 0) \\ = &\ 0 \end{aligned}$$

For invalid argument, find a "counter example (refutation box), and verify

	1	(x)(Fx · Gx)		Valid
		[∴ (x)Fx		
*	2	┌ asm: ~(x)Fx		
*	3	│ ∴ (∃x)~Fx	{from 2}	
	4	│ ∴ ~Fa	{from 3}	
	5	│ ∴ (Fa · Ga)	{from 1}	
	6	└ ∴ Fa	{from 5}	
	7	∴ (x)Fx	{from 2; 4 contradicts 6}	

	1	(x)(Lx ⊃ Fx)		Invalid
*	2	(∃x)Lx		a, b
		[∴ (x)Fx		
*	3	asm: ~(x)Fx		La, Fa
*	4	∴ (∃x)~Fx	{from 3}	~Lb, ~Fb
	5	∴ La	{from 2}	
	6	∴ ~Fb	{from 4}	
*	7	∴ (La ⊃ Fa)	{from 1}	
*	8	∴ (Lb ⊃ Fb)	{from 1}	
	9	∴ Fa	{from 5 and 7}	
	10	∴ ~Lb	{from 6 and 8}	

1. Reverse squiggles.
2. Drop initial existentials, using a new letter each time.
3. Lastly, drop initial universals, using all the old letters. (Only use a new letter if you've done everything else possible, including further assumptions if needed, and still have no old letters.)
4. If you can't get a contradiction, construct a refutation.

Harder translations add statement letters, individual constants, and non-initial or multiple quantifiers:

$(S \supset Cr)$	=	If it's snowing, then Romeo is cold.
$((x)Ix \supset (x)Lx)$	=	If all are Italian, then all are lovers.

Use a separate quantifier for each "all," "some," and "no"; and place the quantifiers to mirror where they occur in English:

LogiCola H (HM, HT)

Wherever the English has	put this in the wff
all (every)	(x)
not all (not every)	$\sim(x)$
some	$(\exists x)$
no	$\sim(\exists x)$

To translate a sentence with "any":

Rephrase it to mean the same thing but not use "any," and then translate the second sentence.	or	Put a "(x)" at the *beginning* of the wff, regardless of where "any" occurs in the sentence.

Not anyone is rich = $\sim(\exists x)Rx$ = $(x)\sim Rx$

Not any Italian is a lover = $\sim(\exists x)(Ix \cdot Lx)$ = $(x)\sim(Ix \cdot Lx)$

If anyone is just, there will be peace = $((\exists x)Jx \supset P)$ = $(x)(Jx \supset P)$

Pages 116 to 118

Harder translations add statement letters, individual constants, and non-initial or multiple quantifiers:	$(S \supset Cr)$ $((x)Ix \supset (x)Lx)$

Wherever the English has:	➔	all (every)	not all	some	no
put this in the formula:	➔	(x)	$\sim(x)$	$(\exists x)$	$\sim(\exists x)$

With "all … is …," use "⊃" for the *middle* connective.	*Otherwise* use "•" for the connective.

To translate a sentence with "any":

- Rephrase it to mean the same thing but not use "any," and then translate the second sentence.
- OR: Put a "(x)" at the *beginning* of the wff, regardless of where "any" occurs in the sentence.

Proofs with harder formulas:

- use statement letters, individual constants, or non-initial or multiple quantifiers,
- often require multiple assumptions, but
- require no new inference rules.

LogiCola I (HC, MC)

Remember to drop only initial quantifiers.

"$((x)Fx \supset (x)Gx)$" is an if-then and follows the if-then rules.