CHAPTER 7 – PREDICATE LOGIC: SYMBOLIZATION AND SYNTAX

Section 7.1 Predicates, Singular Terms, and Quantity Expressions of English

- ***The advantages of SL***:

a) there are decidable test procedures associated with the notions of truth, falsity, validity, equivalence, and consistency, and

b) at least some of the logical results of these tests can be carried over to English.

Section 7.1 Predicates, Singular Terms, and Quantity Expressions of English

- The main problem with SL: the language of SL is not sophisticated enough to allow adequate symbolization of a great deal of natural language, e.g., there is no symbolization in SL of general claims such as "Each citizen will either vote or pay a fine. Also, some clearly valid arguments, when symbolized in SL, come out as invalid, e.g.,

None of David's friends support Republicans.	N
Sarah supports Breitlow, and B is a R.	S & B
So Sarah is no friend of David's.	~F

Section 7.1 Predicates, Singular Terms, and Quantity Expressions of English

- This is because SL, in taking sentences to be the smallest linguistic units, makes all subsentential relationships invisible.
- We will develop a new language, PL (predicate logic), which will allow us to express many subsentential relations.
- However, it will also turn out that PL and its associated procedures do not constitute a decidable system, i.e., there is no mechanical test procedure that will always yield, in a finite number of steps, a ‘yes’ or ‘no’ answer to logical questions.
- We gain expressive power but lose decidability.

Section 7.1 Predicates, Singular Terms, and Quantity Expressions of English

- ***Singular Term****: any word or phrase that designates or purports to designate some one thing.*

- a) proper names: 'George Washington,' 'Henry', etc.
- b) definite descriptions: 'the discoverer of radioactivity,' 'Mary's best friend', etc.

Section 7.1 Predicates, Singular Terms, and Quantity Expressions of English

- What a name or definite description designates clearly depends on the context of use, e.g., 'George Washington', the first US president, or Mary's dog.
- When we use a sentence of English as an example, or in an exercise set, we shall, unless otherwise noted, be assuming that a context is available for that sentence such that in that context all singular terms do designate.
- Pronouns are sometimes used in the place of proper names and definite descriptions, e.g., If Sue has read Darwin, then she's no creationist.
- Sometimes they are not, e.g., This test is so easy that if anyone fails, then it's her fault.

Section 7.1 Predicates, Singular Terms, and Quantity Expressions of English

- Sentences may contain more than one singular term, e.g., New York is between Philadelphia and Boston.
- ***Predicate****: part of English sentences that can be obtained by deleting one or more singular terms from an English sentence – it is a strip of words with one or more blanks such that when the blanks are filled by singular terms, a sentence of English results.*
- One blank: one-place predicate…n-blanks: n-place predicates, e.g., ____works for____, ____is between____and____.

- In displaying the variable of predicates, we shall use the lowercase letters 'x', 'y', 'z'…, e.g., x is between y and z.
- And we use capitals for predicates, e.g., Wxy, Bxyz

Section 7.1 Predicates, Singular Terms, and Quantity Expressions of English

- What is missing, however, is an account of “quantity” terms, such as ‘every’, ‘all’, ‘each’, ‘some’, ‘none’.
- These terms are not singular terms, i.e., there is no one thing that is designated, nor are they predicates…they serve to indicate *quantity*.

E.g.,

Everyone is easygoing.

No one likes Michael.

Michael doesn’t like anyone.

Michael doesn’t like everyone.

Someone likes Sue.

Nobody is taller than him or herself.

Section 7.2 The Formal Syntax of PL

- The vocabulary of PL consists of :

-***Sentence Letters***: *A, B, C…*

-***Predicates***: *Ax, Py…*

-***Individual Terms***: *(a) constants: a, b, c… (b) variables: w, x, y, z…*

-***Truth-Functional Connectives***: *~, &, ∨, ⊃, quantifiers: (∀x), (∃x)*

-***Punctuation Marks***: *( ), [ ]*

Section 7.2 The Formal Syntax of PL

- By including the sentence letters of SL we make every sentence of SL a sentence of PL: every sentence of SL can be symbolized in PL.

- (((( ) (((A ⊃ BBA), (A ⊃ Bab)) are expressions of PL, but (({ABA), A ⊃ ∃, A # Bab are not.

Section 7.2 The Formal Syntax of PL

- We use **P**, **Q, R** as metavariables ranging over expressions of PL, **a** as a metavariable over individual constants, **x** as a metavariable over individual variables.

- A ***quantifier*** of PL is an expression of the form (∀*x) or (*∃*x). An expression of the first form is called a **universal quantifier**, an expression of the second is called an **existential quantifier**.*

Section 7.2 The Formal Syntax of PL

- *Definition **of an Atomic formula** of PL: every expression of PL that is either a sentence letter of PL or an n-place predicate of PL followed by n- individual terms of PL is an atomic formula.*

- *Recursive definition of a '**formula** of PL':*

1. Every atomic formula ***P*** of PL is a formula of PL.
2. If ***P*** *is a formula of PL, then so is* ***~P***
3. If ***P*** *and* ***Q*** *are formulas of PL, then so are* ***(P & Q)****,* ***(P*** $\vee$ ***Q)****,* ***(P*** $\supset$ ***Q)****, and* ***(P*** $\equiv$ ***Q)****.*
4. If ***P*** *is a formula of PL that contains at least one occurrence of* ***x*** *and no* ***x****-quantifier, then* $\forall$***xP*** *and* $\exists$***xP*** *are formulas of PL.*
5. Nothing is a formula of PL unless it can be formed by 1-4.

Section 7.2 The Formal Syntax of PL

- ***Logical operator*** *of PL: an expression of PL that is either a quantifier or a truth-functional connective.*

E.g., Rabz is an atomic formula.

~(Rabz & Hxy) is a formula

(Hab ⊃ (∀z)(Fz ⊃ Gza)) is a formula

(∀z)(Haz ⊃ (∀z)(Fz ⊃ Gza)) is not a formula (4)

Section 7.2 The Formal Syntax of PL

- Not all formulas of PL are sentences of PL (though all sentences are formulas).

1. If ***P*** *is an atomic formula of PL, then* ***P*** *contains no logical operators, and hence no main logical operator, and* ***P*** *is the only subformula of* ***P****.*
2. If ***P*** *is a formula of PL of the form* **~*Q***, *then '~' is the main logical operator of* ***P****, and* ***Q*** *is the immediate subformula.*
3. If ***P*** *is a formula of PL of the form (****Q*** *&* ***R****), (****Q*** $\vee$ ***R****), (****Q*** $\supset$ ***R****), or (****Q*** $\equiv$ ***R****), then the binary connective between* ***Q*** *and* ***R*** *is the main logical operator of* ***P****, and* ***Q*** *and* ***R*** *are the immediate subformulas of* ***P****.*
4. If ***P*** *is a formula of the form (*$\forall$***x****)****Q*** *or (*$\exists$***x****)****Q****, then the quantifier that occurs before* ***Q*** *is the main logical operator of* ***P****, and* ***Q*** *is the immediate subformula of* ***P****.*
5. If ***P*** *is a formula of PL, then every subformula of a subformula of* ***P*** *is a subformula of* ***P****, and* ***P*** *is a subformula of itself.*

Section 7.2 The Formal Syntax of PL

- ***Scope of a quantifier****: is the subformula* ***Q*** *of* ***P*** *of which the quantifier is the main logical connective. E.g., $(\forall x)Fxy$ has subformula Fxy and its scope '$(\forall x)Fxy$', likewise, $(Hx \supset (\forall y)(Fxy)$ has scope '$(\forall y)Fxy$'.*

- ***Bound variable****: an occurrence of a variable* ***x*** *in a formula* ***P*** *of PL is bound iff that occurrence is within the scope of an* ***x****-quantifier.*

- ***Free variable***: an occurrence of a variable ***x*** *in a formula* ***P*** *of PL if free iff it is not bound.*

Section 7.2 The Formal Syntax of PL

- *Definition* of ***Sentence of PL****: a formula* ***P*** *of PL is a sentence of PL iff it has no free variables.*

E.g., (Hx ⊃ (∀y)Fxy) is not a sentence of PL; neither occurrence of x is bound.

((∀z)Gz ⊃ ~Hz) is not a sentence of PL; the last occurrence of z is unbound.

- Sentences of PL are either quantified [quantifier as main logical operator], truth-functional [truth-functional connective as main logical operator], or atomic [a sentence having no logical operator].

- ***Substitution instance*** *of* ***P****: If* ***P*** *is a sentence of PL of the form (∀****x****)****Q*** *or (∃****x****)****Q*** *and an individual constant* ***a****, then* ***Q*** *(****a/x****) is a substitution instance of* ***P****. The constant* ***a*** *is the* ***instantiating constant****. E.g., if P is (∀x)Fxb, then P(a/x) is Fab and P(b/x) is Fbb.*

Section 7.3 Introduction to Symbolization

- Basic elements of the formal language PL:
  - Logical operators: connectives: ‘&’, ‘$\vee$’, ‘$\supset$’, ‘$\equiv$’, ‘~, quantifiers: $\forall x$, $\exists x$
  - Individual terms: individual constants: a, b, c…, and variables, x, y, z…
  - Predicates: Fx, Gxy, Hz,…

E.g., Fx is a one-place, Fxy is two-place, where x, y, z… are the variables that stand for singular terms.

We call the set of things being talked about the ***universe of discourse***.

Section 7.3 Introduction to Symbolization

E.g., UD: People in Michael's office

Lxy: x likes y

Ex: x is easygoing

Txy: x is taller than y

h: Henry

m: Michael

v: Vito

s: Sue

Section 7.3 Introduction to Symbolization

- Sentences are then generated by replacing the variables with constants, e.g., Lsh is ‘Sue like Henry’, Lhs is ‘Henry likes Sue’.
- Similarly, we can symbolize sentences,

- e.g., ‘Sue is Easygoing’ is Es,

‘If Rita likes Henry, then Rita doesn’t like Michael’ is Lrh $\supset$ ~Lrm,

‘Everyone is Easygoing’ is (Es & Eh) & (Ev & Em),

‘Michael likes someone’ is (Lmh $\vee$ Lmm) $\vee$ (Lmv $\vee$ Lms).

Section 7.3 Introduction to Symbolization

- We can symbolize ‘every’ with conjunctions and ‘some’ with disjunctions, but if the UD is very large or infinite this becomes awkward.
- We need quantifier symbols and variables that represent what the quantifier ranges over.
- The quantifier symbols are ‘$\forall$’ and ‘$\exists$’ and the variables ‘w’, ‘x’, ‘y’, ‘z’

Section 7.3 Introduction to Symbolization

- A quantifier of PL consists of a quantifier symbol followed by a variable, both in parentheses: '$(\forall x)$' and '$(\exists x)$'; e.g., 'Michael likes everyone' is $(\forall x)Lmx$ and 'Michael likes someone' is $(\exists x)Lmx$.

- Quantifiers of the form '$(\forall x)$' are called *universal quantifiers.*
- Quantifiers of the form '$(\exists x)$' are called *existential quantifiers.*

- The things being talked about, the members of the current universe of discourse, are called the ***values of the variable****.*

- Some … at least one …
- All … each, every…

Section 7.3 Introduction to Symbolization

- E.g., UD: People in Michael's office

Everyone is easygoing.	$(\forall x)Ex$
No one likes Michael.	$\sim(\exists x)Lxm$
Michael doesn't like anyone.	$\sim(\exists x)Lmx$
Michael doesn't like everyone.	$\sim(\forall x)Lmx$
Someone likes Sue.	$(\exists x)Lxs$
Nobody is taller than him or herself.	$\sim(\exists x)Txx$

Section 7.3 Introduction to Symbolization

- E.g., UD: People in Michael's office

If Rita is easygoing, everyone is. $Er \supset (\forall x)Ex$

Rita likes Sue iff everyone does. $Lrs \equiv (\forall x)Lxs$

Henry likes everyone but Sue doesn't. $(\forall x)Lhx \ \& \sim(\forall x)Lsx$

Henry likes everyone but Sue doesn't like anyone $(\forall x)Lhx \ \& \sim(\exists x)Lsx$

- Note: $(\forall x)Fx = \sim(\exists x)\sim Fx$,

  $\sim(\forall x)Fx = (\exists x)\sim Fx$,

  $\sim(\exists x)Fx = (\forall x)\sim Fx$

Section 7.3 Introduction to Symbolization

- ***Multiple Quantifers with Overlapping Scope****: when one quantifier falls within the scope of another.

E.g., Everyone likes everyone:

$(\forall x)(\forall y)Lxy$

Someone likes someone:

$(\exists x)(\exists y)Lxy$

Everyone likes someone:

$(\forall x)(\exists y)Lxy$

* Does not appear as a separate section in the text but begins on page 286

Section 7.3 Introduction to Symbolization

- When we have two universal (or two existential) quantifiers, the order of the quantifiers does not matter.
- When we have 'mixed' quantifiers, the order does matter.

E.g., $(\forall y)(\exists x)Lyx$ is not the same as $(\exists x)(\forall y)Lyx$

The former says 'For all y there is some object x, such that y likes x,' (everyone likes someone), whereas the latter says that there is some object x, such that for all y, y likes x (someone is liked by everyone).

- If we expand the UD, we must introduce a predicate to restrict it to, say, persons. Generally we use 'Px' to mean x is a person.

E.g., UD: living things

Everyone likes everyone:

$(\forall x)(\forall y)[(Px \& Py) \supset Lxy]$

Section 7.3 Introduction to Symbolization

- *Note: choice of UD affects symbolization:*

All marbles are blue.

None of the marbles are blue.

Some of the marbles are blue.

UD: Marbles	UD: Marbles and marble players
$(\forall y)By$	1. $(\forall y)(My \supset By)$
$\sim(\exists y)By$	2. $(\forall y)(My \supset \sim By)$
$(\exists y)By$	3. $(\exists y)(My \& By)$

Section 7.3 Introduction to Symbolization

- Generally if we restrict our UD to **P**, then

A: $(\forall x)Qx$		A: $(\forall x)(Px \supset Qx)$
E: $(\forall x){\sim}Qx$	represents	E: $(\forall x)(Px \supset {\sim}Qx)$
I: $(\exists x)Q$		I: $(\exists x)(Px \;\&\; Qx)$
O: $(\exists x){\sim}Q$		O: $(\exists x)(Px \;\&\; {\sim}Qx)$

Section 7.3 Introduction to Symbolization

- Before concluding, we note a limited parallel between PL and ***Aristotelian logic*** which recognizes found kinds of sentences
- A: All Q are R — $(\forall x)(Qx \supset Rx)$

  E: No Q are R — $(\forall x)(Qx \supset \sim Rx)$

  I: Some Q are R — $(\exists x)(Qx \,\&\, Rx)$

  O: Some Q are not R — $(\exists x)(Qx \,\&\, \sim Rx)$

E.g., UD: Jawbreakers

Yz: z is yellow

Sz: z is sweet

All yellow jawbreakers are sweet.	$(\forall w)(Yw \supset Sw)$
No yellow jawbreakers are sweet.	$(\forall w)(Yw \supset \sim Sw)$
Some yellow jawbreakers are sweet.	$(\exists x)(Yw \,\&\, Sw)$
Some yellow jawbreakers are not sweet.	$(\exists x)(Yw \,\&\, \sim Sw)$

Section 7.3 Introduction to Symbolization

- Aristotle's square of opposition:

$(\forall x)(P \supset Q) \equiv \sim(\exists x)(P \& \sim Q)$

$(\forall x)(P \supset \sim Q) \equiv \sim(\exists x)(P \& Q)$

$(\exists x)(P \& Q) \equiv \sim(\forall x)(P \supset \sim Q)$

$(\exists x)(P \& \sim Q) \equiv \sim(\forall x)(P \supset Q)$

Section 7.3 Introduction to Symbolization

- *Note: the English sentence may not contain the terms 'all', 'no', 'some'.*

E.g., Mammals are warm blooded. $(\forall x)(Mx \supset Wx)$

A reptile is not warmblooded. $(\forall x)(Rx \supset \sim Wx)$

There are carnivorous mammals. $(\exists x)(Cx \,\&\, Mx)$

- *Note: context matters, e.g., "all lawyers are not rich" does not, in all likelihood, mean* $(\forall x)(Lx \supset \sim Rx)$.

Rather, it means $\sim(\forall x)(Lx \supset Rx)$ *or* $(\exists x)(Lx \,\&\, \sim Rx)$.

Section 7.3 Introduction to Symbolization

- More complex forms.

Michael likes everyone that both Sue and Rita like:

$(\forall x)((Lrx \,\&\, Lsx) \supset Lmx)$

Rita doesn't like Michael, but she likes everyone Michael likes:

$\sim Lrm \,\&\, (\forall y)(Lmy \supset Lry)$

Every integer is odd or even:

$(\forall x)(Ix \supset (Ox \vee Ex))$ or $(\forall x)(Ox \vee Ex)$

Every integer is odd or every integer is even:

$(\forall x)Ox \vee (\forall x)Ex$ or $(\forall x)(Ix \supset Ex) \vee (\forall x)(Ix \supset Ox)$

Section 7.3 Introduction to Symbolization

- *Note: 'any' can be tricky*

Anyone who likes Sue likes Rita:
$(\forall x)(Lxs \supset Lxr)$
If anyone likes Sue Michael does:
$(\exists x)Lxs \supset Lms$

- *Rule: where a quantity term is used in the antecedent of an English conditional and there is, in the consequent, pronominal cross-reference to that quantity term, a universal quantifier is called for.*

E.g., Anyone who fails the exam flunks the course:
$(\forall x)(Fx \supset Cx)$

Section 7.3 Introduction to Symbolization

- E.g., If any student passes, Donna will pass: $(\exists x)Px \supset Pd$.

Each student is such that if that student passes, Donna will pass:

$(\forall x)(Px \supset Pd)$.

- But both ‘$(\forall x)(Px \supset Pd)$’ and ‘$(\exists x)Px \supset Pd$’ commit us to Donna’s passing if at least one student passes. These are, then, equivalent.

- *Other pairs of equivalent sentence forms are listed on pages 301 302 of the text.

Section 7.3 Introduction to Symbolization

- *Note: there is a difference between 'any' when combined with negation.*

1. Michael likes everyone. $(\forall x)Lmx$
2. Michael likes anyone. $(\forall x)Lmx$
3. Michael doesn't like everyone. $\sim(\forall x)Lmx$
4. Michael doesn't like anyone. $\sim(\exists x)Lmx$

- *Note: 'some' is not always symbolized by existential quantifier.*

E.g., If someone likes Sue, then he or she likes Rita:

$(\forall x)(Lxs \supset Lxr)$

If someone likes Sue, then someone likes Rita:

$(\exists x)Lxs \supset (\exists y)Lyr$

Section 7.3 Introduction to Symbolization

- *Note: in the system of logic we are studying, if 'All rabbits are mammals' is true, it does not follow that 'Some rabbits are mammals' (I-sentence) is true. This is because PL allows predicates that are not satisfied by any member of the UD.*

- ***Problem***: not every sentence of English or of PL can reasonably be construed as being of the A, E, I, O form.

Section 7.3 Introduction to Symbolization

- Recall our first example:

None of David's friends supports Republicans.

<u>Sarah supports Breitlow and Breitlow is a Republican.</u>

So Sarah is no friend of David's.

UD: people

Fxy: x is a friend of y Sxy: x supports y

Rx: x is Republican d: David

b: Breitlow s: Sarah

$(\forall x)[Fxd \supset {\sim}(\exists y)(Ry \,\&\, Sxy)]$

<u>Ssb & Rb</u>

~Fsd

- Note: we are skipping Section 7.4

Section 7.5 The Language of PLE (Predicate Logic Extended)

Identity

- When ‘some’ means at least two.

e.g., ‘There are some apples in the basket’:

UD: everything

Nxy: x is in y

Ixy: x is identical with y

Ax: x is an apple

b: the basket

There is at least one apple in the basket:

$(\exists x)(Ax \& Nxb)$

There are at least two apples in the basket:

$(\exists x)(\exists y)([(Ax \& Ay) \& (Nxb \& Nyb)] \& {\sim}Ixy)$

Section 7.5 The Language of PLE (Predicate Logic Extended)

- Using different variables does not commit us to there being more than one thing of the specified sort.

- PLI is predicate logic (PL) together with the symbol '=' for the two-place predicate xIy.

Section 7.5 The Language of PLE (Predicate Logic Extended)

There is exactly one apple in the basket:

$(\exists x)[(Ax \& Nxb) \& (\forall y)((Ay \& Nyb) \supset x = y)]$

Henry has not read Alice in Wonderland, but everyone else has:

$\sim Ah \& (\forall y)(y = \sim h \supset Ay)$

Only Henry and Bob have not read Alice in Wonderland:

$\sim(Ah \vee Ab) \& (\forall x)[\sim Ax \supset (x = h \vee x = b)]$

Section 7.5 The Language of PLE (Predicate Logic Extended)

E.g., The Roman general who defeated Pompey invaded both Gaul and Germany. Therefore, Pompey was defeated by someone who invaded both Gaul and Germany.

UD: persons and countries

Ixy: x invaded y

Dxy: x defeated y

v: the Roman general who defeated Pompey

p: Pompey

g: Gaul

e: Germany

Ivg & Ive

(∃x)(Dxp & (Ixg & Ixe)

- This argument, so symbolized, is invalid.

Section 7.5 The Language of PLE (Predicate Logic Extended)

- ***Definite descriptions*** are descriptions that are satisfied by exactly one thing, so we can capture the 'finer' structure of definite descriptions by using PLI.
- We will add 'Rx' for 'x is a Roman general', so we get:

$(\exists x)[((Rx \& Dxp) \& (\forall y)((Ry \& Dyp) \supset y = x)) \& (Ixg \& Ixe)]$.

- We can then show in PLI that the conclusion, $(\exists x)(Dxp \& (Ixg \& Ixe))$, does follow.
- We should transform definite descriptions into unique existence claims, e.g., 'John's only daughter is a biochemist': $(\exists x)((Dxj \& (\forall y)(Dyj \supset y = x)) \& Bx)$.

Section 7.5 The Language of PLE (Predicate Logic Extended)

Properties of Relations

- Properties of relations: identity has three special properties:

transitivity: $(\forall x)(\forall y)(\forall z)((x = y \mathbin{\&} y = z) \supset x = z)$

Other transitive relations include taller, larger, ancestor, heavier, etc.

symmetric: $(\forall x)(\forall y)(x = y \supset y = x)$

Other symmetric relations include sibling, classmate, relative, etc.

reflexive: $(\forall x)x = x$

Other reflexive relations include same age as y, same color as y, etc.

Section 7.5 The Language of PLE (Predicate Logic Extended)

Functions

- A function is an operation that takes one or more element of a set as arguments and returns a single value.
- Example: Addition f(x,y) takes in x, y and return x+y, subtraction g(x, y) takes in x, y and returns x-y, successor f'(x) takes in x and returns n+1
- Not all functions are from arithmetic, we have encounter truth-function – functions that map values from the set {T, F} to truth values
- Example: Negation takes in T and returns F or takes in F and returns T, conjunction takes in T,T and returns T, and for all other arguments returns F

Section 7.5 The Language of PLE (Predicate Logic Extended)

- Functions are also found outside both mathematics and logic

- Example: Spouse takes in x and returns the spouse of x, twin takes in x and returns the twin of x.

- In *PLE we use lowercase italicized Roman letters a-z, with or without a positive integer subscript to symbolize functions. We call these symbols* ***functors****.*

Section 7.5 The Language of PLE (Predicate Logic Extended)

- Example: UD: Positive Integers

f(x): the successor of x	h(x,y): the sum of x and y
Ex: x is even	Ox: x is odd
a: 2	b: 3

3 is odd	Ob
the successor of 2 is odd	Of(a)

f(a) = b	the successor of 2 is 3
∃xOf(x) & ∃xEf(x)	some positive integers are odd and some are even
∀x∀y ((Ex & Oy) ⊃ Oh(x,y))	the sum of an even and odd number is odd

Section 7.5 The Language of PLE (Predicate Logic Extended)

Example: UD: set of twins

f(x): the twin of x B(x): x is bald

h: Henry j: Jose s: Simone

Simone is Henry’s twin $s = f(h)$

Jose is Simone’s twin $j = f(s)$

A twin is bald if and only if his or her twin is bald $\forall x(Bx \equiv Bf(x))$

Some bald twins have twins that are not $\exists x(Bx \ \& \ {\sim}Bf(x))$

Section 7.5 The Language of PLE (Predicate Logic Extended)

- We require that the functions we symbolize with functors have the following characteristics:
- An n-place function must yield one and only one value for each n-tuple of arguments
- The value of a function for an n-tuple of members of a UD must be a member of that UD

Section 7.5 The Language of PLE (Predicate Logic Extended)

- We have seen, functors can be used to generate a new kind of individual term (in addition to individual constants and variables); these have the form $f(t_1, t_2,\ldots t_n)$ where f is an n-place functor and t_1, $t_2,\ldots t_n$ are individual terms

- ***The Syntax of PLE***, in addition to the vocabulary of PL, PLE includes:
  - '=' the two place ***identity predicate***
  - ***Functors*** of PLE

Section 7.5 The Language of PLE (Predicate Logic Extended)

Individual terms

	Open Individual Term	**Closed Individual Term**
Simple	Individual variables: x, y, z	Individual constants: a, b, c…
Complex	Individual term formed from a functor and at least one Individual variable: f(x), f(a,x), g (f(a), y)	Individual term formed from a functor and no individual variable: f(a), g(a,b), g(f(b))

Section 7.5 The Language of PLE (Predicate Logic Extended)

- In *PLE a* **substitution instance** *is defined as follows: If* ***P*** *is a sentence of PLE of the form* $\forall$***xQ*** *or* $\exists$***xQ*** *and* **t** *is a closed individual term, then* ***Q(t/x)*** *is a substitution instance of* ***P****. The individual term* **t** *is the* ***instantiating individual term****.*